NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

GUILHERME CRUZ FERNANDES

BSc in Computer Science and Engineering

BEAM DIFFUSION MODELS FOR DECODING MULTI-SHOT VISUAL SEQUENCES

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon

December, 2025

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

BEAM DIFFUSION MODELS FOR DECODING MULTI-SHOT VISUAL SEQUENCES

GUILHERME CRUZ FERNANDES

BSc in Computer Science and Engineering

Adviser: João Miguel da Costa Magalhães
Full Professor, NOVA School of Science and Technology

Co-adviser: Idan Szpektor
Google Research

Examination Committee

Chair: João Alexandre Carvalho Pinheiro Leite
Full Professor, NOVA School of Science and Technology

Rapporteur: Ana Catarina Fidalgo Barata
Auxiliar Professor, Instituto Superior Técnico

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
December, 2025

Beam Diffusion Models for Decoding Multi-Shot Visual Sequences

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.29) [1].

Acknowledgements

I would like to express my heartfelt gratitude to my advisor, Professor João Magalhães, for his invaluable guidance, encouragement, and for opening the door to inspiring projects and experiences throughout my research journey. I am equally thankful to my co-advisor, Idan Szpektor and Regev Cohen from Google Research, for the valuable discussions, feedback, and ideas that helped shape this work. I am also deeply grateful to Vasco Ramos, whose advice, support, and patience as a PhD mentor greatly contributed to both my learning and the progress of this thesis.

I am sincerely thankful to my colleagues for the stimulating discussions and the many hours shared in reading groups, which not only enriched this work but also made the research process collaborative and inspiring.

Beyond academia, I owe immense gratitude to my friends who shared in the laughter, conversations, and much-needed distractions along the way, as well as to all those who reminded me of the importance of balance and connection outside research.

Above all, I thank my family for their unconditional love and support, without which this journey would not have been possible.

”

“Research is what I’m doing when I don’t know what I’m doing.”

— Wernher von Braun, en

Abstract

Images have become a powerful medium for communication, offering an intuitive way to convey concepts, instructions, and narratives. In domains such as education, design, and storytelling, sequences of images are often essential for expressing temporal or logical progressions. For instance, illustrating step-by-step guides or unfolding a visual story requires not only generating high-quality images but also maintaining coherence across the entire sequence. While Latent Diffusion Models (LDMs) have achieved impressive results in generating individual images, extending them to generate consistent and semantically aligned image sequences introduces significant challenges. A key limitation lies in their inability to retain visual and contextual continuity across multiple steps. Generated images may individually align with textual descriptions but fail to form a cohesive visual narrative, particularly when the semantics of one image depend on prior context.

To overcome these challenges, we introduce a novel sequence decoding method inspired by beam search, designed to guide LDMs in generating coherent image sequences. At each generation step, the model samples multiple candidate images and evaluates them based on both their alignment with the current text and their consistency with prior visual context. A cross-attention mechanism further strengthens this process by integrating both local prompt information and cumulative visual history, allowing the model to reason over the evolving narrative. This iterative, contrastive selection process enables the model to explore a rich set of latent space candidates while pruning those that break narrative flow or introduce inconsistencies. The result is a visually and semantically consistent image sequence that better reflects the intended progression of instructions or events.

We conduct both automatic and human evaluations to assess sequence coherence, alignment with text, and visual clarity. Results demonstrate that our approach significantly improves sequence consistency over baseline LDMs, enabling more effective communication of complex, multi-step concepts and improving the interpretability of generated content.

Keywords: Latent Diffusion Models, Visual Consistency, Semantic Coherence, Beam Search

Resumo

As imagens tornaram-se um meio poderoso de comunicação, oferecendo uma forma intuitiva de transmitir conceitos, instruções e narrativas. Em áreas como a educação, o design e a narração visual, as sequências de imagens são frequentemente essenciais para representar progressões temporais ou lógicas. Ilustrar guias passo a passo ou desenvolver uma história visual exige não só imagens de alta qualidade, mas também coerência ao longo de toda a sequência. Embora os Modelos de Difusão Latente (LDMs) tenham alcançado resultados impressionantes na geração de imagens isoladas, a sua aplicação à geração de sequências consistentes e semanticamente alinhadas apresenta desafios significativos. As imagens podem, individualmente, corresponder às descrições textuais, mas não formam necessariamente uma narrativa visual coesa, sobretudo quando a semântica de uma imagem depende do contexto anterior.

Para superar estas limitações, propomos um novo método de descodificação sequencial inspirado na técnica de beam search, concebido para orientar os LDMs na geração de sequências de imagens coerentes. Em cada passo, o modelo avalia múltiplas imagens candidatas com base no seu alinhamento com o texto atual e na sua consistência com o contexto visual anterior. Um mecanismo de atenção cruzada reforça este processo, integrando a informação do prompt e o histórico visual acumulado. Este processo iterativo de seleção contrastiva permite explorar um espaço latente diversificado, descartando as opções que comprometem a continuidade da narrativa ou introduzem incoerências. O resultado é uma sequência visual e semanticamente coerente, que reflecte com maior fidelidade a progressão pretendida de instruções ou eventos.

Realizámos avaliações automáticas e humanas para medir a coerência das sequências, o alinhamento com o texto e a clareza visual. Os resultados demonstram que a nossa abordagem melhora significativamente a consistência narrativa face aos modelos base, facilitando a comunicação de conceitos complexos.

Palavras-chave: Modelos de Difusão Latente, Consistência Visual, Coerência Semântica, Beam Search

Contents

1 Introduction

1.1 Instructional Visual Sequences

Visual representations have long been essential in helping humans understand and perform complex tasks. From observing others during early learning to studying diagrams in manuals, visuals provide crucial anchors for interpreting physical procedures [2, 3]. This is particularly relevant in manual tasks, such as cooking or assembling objects, where text alone often fails to convey spatial or procedural nuances. A visual depiction of the instructions that the user need to execute is far richer and helpful, as is confirmed by the growing number of content created in the field, e.g. Instructables, WikiHow, iFixIt.

While traditional instructional materials include static images, these often lack sufficient context or continuity. Users may find it difficult to infer the progression between steps, especially when intermediate states or object transitions are not depicted. This lack of visual consistency or information, can lead to confusion or misinterpretation of the task. To mitigate this problem, one may turn to recent advances in generative models to generate visual illustrations from textual prompts [4–6]. However, generating a sequence of images that maintains semantic and visual coherence presents new challenges. As task complexity increases, so does the need for consistency in object appearance, spatial layout, and alignment with instructions across multiple steps.

1.2 Challenges in Sequential Visual Generation

Instructional tasks typically unfold through a step-by-step structure, but these steps are seldom independent. Earlier actions influence later ones, both in terms of how objects should appear and how they are spatially arranged. For example, in a cooking recipe, omitting a marination step affects not only the flavor but also the visual appearance and handling of the food in subsequent stages. However, most text-to-image models treat each prompt in isolation, without accounting for previous context. This architectural limitation prevents them from learning dependencies across time, resulting in generated sequences that often suffer from inconsistencies: objects may change appearance between

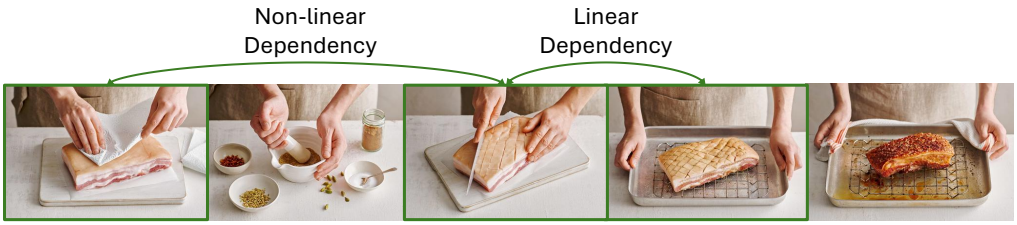


Figure 1.1: Illustration of linear and nonlinear dependencies in image sequence generation.

steps, tools might suddenly disappear, or the environment may shift unexpectedly. These discontinuities can disrupt the user's mental model of the task, making the instructions harder to follow and reducing the overall reliability of the visual guide.: objects may change appearance between steps, tools might suddenly disappear, or the environment may shift unexpectedly. These discontinuities can disrupt the user's mental model of the task, making the instructions harder to follow and reducing the overall reliability of the visual guide.

For sequential image generation to be effective, models must reason over the full set of instructions, maintain visual continuity, and account for both linear and non-linear dependencies between steps (see Figure 1.1). This calls for context-aware visual synthesis methods that satisfy the following properties:

- Alignment with textual instructions;
- Temporal consistency across sequential steps;
- Handling of dependencies, both linear and non-linear, throughout the task.

To address these requirements, we propose **BeamDiffusion**, a decoding-time algorithm that addresses the shortcomings of step-isolated generation by incorporating both prior text and previously generated images to guide generation. By applying beam search with contrastive scoring, BeamDiffusion selects coherent candidates that promote both semantic alignment and visual consistency across the sequence. While our approach addresses these core properties, we limit our scope to linear and moderately non-linear dependencies. More complex scenarios involving multi-branching instructions or long-range cross-step interactions remain outside the focus of this work.

1.3 Image Sequences as Instructional Narratives

A coherent image sequence acts as a visual narrative: each frame provides a standalone view of a task step while also linking to adjacent steps through implicit visual continuity. Unlike videos, which offer smooth temporal transitions, image sequences must imply change and progression through carefully chosen intermediate states.

As illustrated in Figure 1.2, the objective of instructional image sequence generation is to produce coherent visual representations of each task step while maintaining continuity

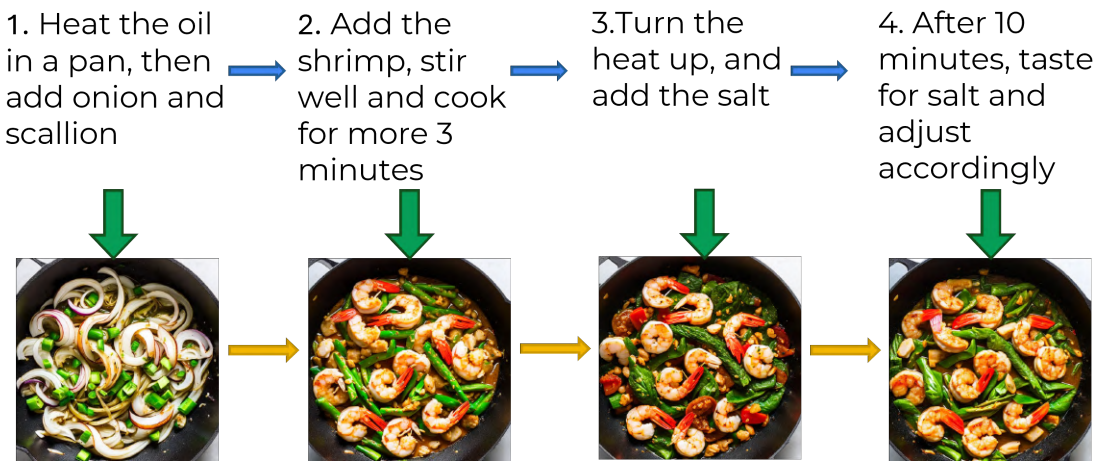


Figure 1.2: Illustration of the goal for image sequence generation: given a sequence of textual steps, the model generates a coherent series of images by conditioning each generated image on the current step's text, the previous steps' text, and the previously generated images to ensure visual and semantic continuity.

across the entire sequence. Each image should reflect not only the current instruction but also the outcomes of prior steps, ensuring semantic alignment and visual stability. This format presents unique affordances and constraints. On one hand, image sequences are easier to render, distribute, and interpret in static contexts. On the other hand, they require precise modeling of transitions, object states, and visual consistency since temporal cues are absent. Thus, instructional image generation is not merely about creating plausible illustrations, it requires encoding a form of visual logic that aligns with the progression and dependencies outlined in the instructions.

1.4 Objective

The primary objective of this thesis is **to research and propose a novel generative algorithm, capable of producing coherent and contextually aligned sequences of images that visually represent multi-step instructional tasks**. Unlike conventional single-image generation methods, our approach focuses on the unique challenges of sequential generation, such as *maintaining semantic and visual continuity across a series of discrete images*. This involves ensuring that each generated image accurately reflects not only the current instructional step but also the cumulative outcomes and object states, an aspect that conventional models typically ignore, leading to incoherent sequences in real-world applications.

To achieve this, our model must effectively *capture the dependencies and nonlinear relationships that exist between sequential steps*, including object transformations, spatial arrangements, and task-specific constraints. By integrating information from prior textual instructions and previously generated images, the model aims to produce visual narratives that enhance user comprehension and reduce ambiguities inherent in text-only or isolated image-based guidance.

Ultimately, the goal is to create *a generalizable generative algorithm for instructional content generation that is robust across diverse domains*, including cooking, crafting, assembly and visual storytelling, where clear visual sequencing is critical. Through this work, we seek to advance the state of the art in instructional image synthesis by addressing the challenges of continuity, coherence, and scalability in sequential generation tasks.

1.5 Contributions

This thesis makes the following key contributions:

- **Fundamental problem:** We analyze the fundamental research challenges inherent to sequential image generation for instructional tasks, emphasizing the need for capturing dependencies and preserving visual continuity.
- **Novel algorithm:** We propose a novel generative algorithm that conditions each image generation on both the current and prior textual instructions, as well as previously generated images, enabling context-aware and temporally coherent synthesis.
- **Experimental Validation:** We demonstrate the effectiveness of our approach through extensive experiments on benchmark datasets, showcasing improved consistency and semantic alignment compared to single-image generation baselines.
- **Generalization:** We provide qualitative and quantitative evaluations illustrating the potential of our method to serve as an aid in real-world instructional scenarios, such as cooking or assembly guidance.

1.6 Document Structure

This thesis is organized into seven chapters, each building upon foundational concepts to present the proposed BeamDiffusion model and its evaluation.

- **Chapter 2 – Background:** This chapter reviews foundational concepts, including the Transformer architecture, encoder-decoder models and text decoding strategies. It concludes with an overview of multimodal representation learning.
- **Chapter 3 – Related Work:** A comprehensive survey of relevant literature, this chapter focuses on diffusion models and their extensions. It covers the fundamentals of the diffusion process, training methods, U-Net architectures, and latent diffusion. Additionally, it explores interpretability efforts, alternatives to diffusion-based approaches, and recent developments in visual sequence generation.
- **Chapter 4 – BeamDiffusion Model:** This chapter introduces the core contribution of the thesis. It formally defines the problem and outlines the BeamDiffusion model, including its mechanisms for latent space exploration and beam pruning. Techniques

such as contrastive scoring and decoding risk minimization are also discussed, along with implementation details.

- **Chapter 5 – Experimental Setup and Methods:** This chapter describes the datasets used, baseline models for comparison, and evaluation methodologies. Both human and automated evaluation protocols are explained, including the Gemini evaluation and automatic metrics for measuring performance and agreement.
- **Chapter 6 – Results and Discussion:** A detailed analysis of experimental results is presented here. It includes quantitative metrics, qualitative insights, and an exploration of how latent selection affects performance. The chapter also examines the impact of non-linear sequence denoising and optimal configurations for beam search.
- **Chapter 7 – Conclusions:** The final chapter summarizes the achievements and contributions of the thesis, discusses research challenges and limitations, outlines broader impacts, and presents directions for future work.

This structure is intended to guide the reader progressively from foundational knowledge to the novel contributions of this work and their empirical validation.

2

Background

The field of neural networks and generative models has witnessed rapid advancements, driven by continuous innovation in model architectures, training methodologies, and the integration of multimodal data. These developments have enabled machines to generate increasingly realistic and coherent outputs across a variety of modalities, such as text, images, and audio. This chapter provides an overview of foundational concepts that underpin modern generative systems, including embeddings as a method of representing input data in continuous vector spaces, encoder-decoder structures for sequence modeling, transformer architectures that have become central to state-of-the-art models, and various strategies for text decoding. Understanding these components is crucial for analyzing and contextualizing the design and capabilities of contemporary generative models.

2.1 The Transformer Architecture

The transformer architecture [7], illustrated in Figure 2.1 has significantly reshaped neural network design, introducing novel mechanisms that enhance both efficiency and effectiveness in processing sequential data. Unlike traditional approaches such as recurrent neural networks (RNNs) [8] and long short-term memory (LSTM) networks [9], which process sequences iteratively, transformers leverage parallelization and a self-attention mechanism to capture relationships across the entire input sequence simultaneously. This innovation enables transformers to process large datasets more efficiently and extract global dependencies in data.

2.1.1 Positional Embeddings

Embeddings [10] play a foundational role in modern neural networks by transforming discrete data into continuous vector spaces. These compact representations capture semantic relationships, allowing models to process and understand complex input data effectively. Transformers process input sequences in parallel rather than sequentially, which means they lack an inherent understanding of order. To address this, positional embeddings are added to the input embeddings to encode sequence order information.

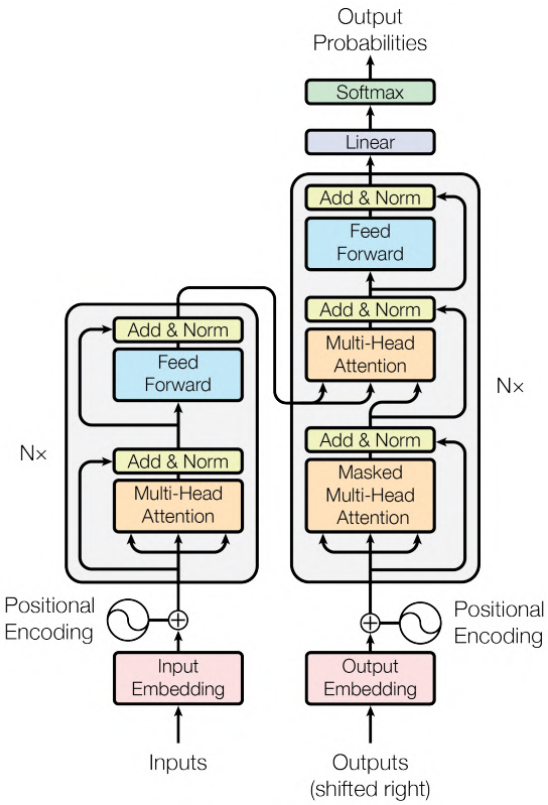


Figure 2.1: Transformer architecture [7]

These embeddings can be learned parameters or derived from fixed functions, such as sine and cosine functions of varying frequencies, as described in the original transformer paper [7]. The positional encoding is defined as:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{2.1}$$

where pos is the position, i is the dimension, and d_{model} is the embedding size. By incorporating positional embeddings, transformers gain the ability to recognize and leverage the order of tokens within sequences.

2.1.2 Core Principles of Self-Attention

At the heart of the transformer lies the self-attention mechanism, illustrated in Figure 2.2, which dynamically weighs the importance of different elements in an input sequence. Unlike RNNs, which process sequences in order, self-attention allows for global context comprehension by attending to all tokens simultaneously. Self-attention operates by creating three learned projections for each token, queries (Q), keys (K), and values (V). Attention scores are computed by measuring the similarity between queries and keys. These scores are normalized using a softmax function to generate attention weights, which are then applied to the value vectors, producing context-aware token representations. The

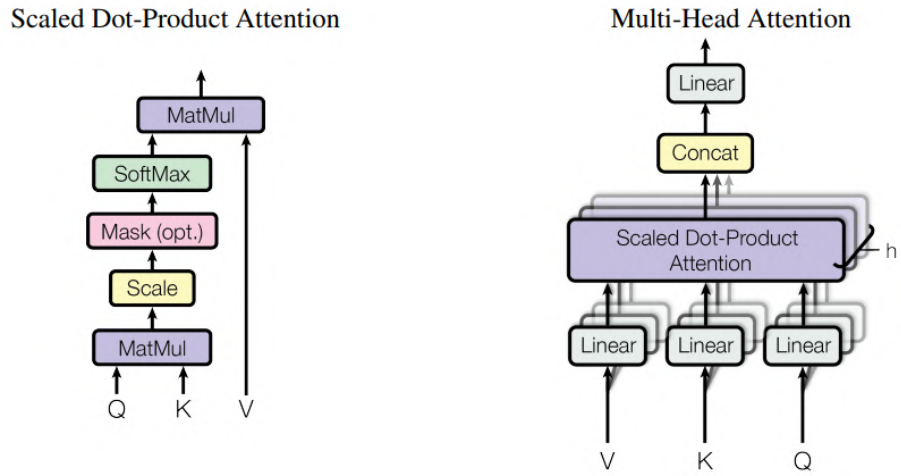


Figure 2.2: (Left) Scaled Dot-Product Attention. (Right) Multi-Head Attention [7]

self-attention mechanism can be expressed as,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V \tag{2.2}$$

where d_k represents the dimensionality of the key vectors.

2.1.3 Cross-Attention for Conditional Dependencies

Cross-attention extends the self-attention mechanism by allowing one sequence to attend to another, rather than itself. This mechanism is particularly valuable in multi-modal models and encoder-decoder architectures, where information from different sources needs to be integrated effectively [11, 12]. Unlike self-attention, which computes attention weights within a single input sequence, cross-attention takes queries from one sequence and keys and values from another. In applications such as machine translation, cross-attention allows the decoder to dynamically attend to relevant encoder states while generating output tokens. Similarly, in vision-language models, cross-attention facilitates interactions between image embeddings and text representations, enhancing the fusion of multi-modal information. By incorporating cross-attention, transformers gain the ability to process conditional dependencies, making them highly effective for tasks that require information integration from multiple sources.

2.1.4 Multi-Head Attention for Diverse Representations

The multi-head attention mechanism builds upon self-attention by introducing multiple independent attention heads, each operating with distinct learned parameters. While self-attention computes a single set of attention weights for a given token, multi-head attention splits the input into multiple subspaces, where each attention head calculates its own set of attention scores. This allows the model to capture a broader range of relationships and dependencies within the sequence. For instance, one attention head might focus

on local, short-range dependencies, while another might capture long-range interactions or more abstract patterns. After computing the attention scores and generating context-aware representations for each head, the results are concatenated and passed through a learned linear transformation to produce the final output. This process allows multi-head attention to integrate information from diverse perspectives, enhancing the model's ability to capture a broader range of relationships within the sequence. While self-attention enables the model to focus on relevant tokens, multi-head attention improves its capacity to simultaneously attend to different syntactic features, contextual nuances, and other task-specific patterns. As a result, multi-head attention generates richer and more varied representations, making it a powerful tool for tasks like language modeling, machine translation, and other sequence-based applications.

2.1.5 Feedforward Networks and Residual Connections

In addition to the self-attention mechanism, transformers incorporate position-wise feed-forward neural networks (FFNs) [13] in each layer. These FFNs are simple yet powerful fully connected networks applied independently to each token representation. Despite their simplicity, they play a critical role in transforming and refining the representations output by the attention mechanism. An FFN typically consists of two linear transformations separated by a non-linear activation function [14] (such as ReLU). The intermediate layer expands the dimensionality of the token embeddings, allowing the network to capture complex relationships and features before reducing them back to the original size. This transformation enhances the model's capacity to learn and represent intricate patterns within the data. Residual connections are employed to bypass the FFN and other main computations in the layer by directly adding the input of a layer to its output. This addition helps mitigate the vanishing gradient problem, ensuring gradients flow smoothly during backpropagation. Additionally, layer normalization is applied to stabilize training, standardizing the outputs of each layer and improving convergence. Together, these mechanisms create a stable and robust framework that allows transformers to learn effectively, even in deep architectures.

2.1.6 Evolution of Embedding Techniques

The transformer architecture has revolutionized deep learning by introducing self-attention, multi-head attention, and feedforward networks, enabling models to efficiently capture complex dependencies in sequential data. One of the most significant outcomes of this evolution is the advancement of embedding techniques, which have greatly enhanced the ability of models to understand and represent language. Early methods like Word2Vec [15] pioneered word embeddings through the skip-gram and continuous bag-of-words (CBOW) architectures. The skip-gram model predicts surrounding words based on a target word, while CBOW does the inverse, predicting the target word from its context. These methods mapped words into a high-dimensional space, where semantic similarities were reflected

Figure 2.3: Encoder-decoder architecture [18]

in vector proximities. However, these early embeddings were static, meaning each word had a single fixed representation, regardless of its context.

The introduction of transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers)[16] and GPT (Generative Pre-trained Transformer)[17], addressed this limitation by introducing dynamic embeddings. BERT generates context-aware representations by processing sequences bidirectionally, capturing both preceding and succeeding words to enhance semantic understanding. In contrast, GPT focuses on autoregressive text generation, predicting the next token in a sequence to create embeddings optimized for fluent text generation.

These advancements have significantly improved the adaptability and expressiveness of embeddings, allowing models to better capture linguistic nuances. As transformer-based architectures continue to evolve, embedding techniques will likely become even more sophisticated, further enhancing the capabilities of natural language processing and other sequential data tasks.

2.2 Encoder-Decoder Architectures

Encoder-Decoder architectures [18], as shown in Figure 2.3, are a foundational framework for tasks that involve transforming one sequence into another, such as machine translation, summarization, or multimodal generation. By separating the processing of input and output into distinct components, these architectures allow for flexible and efficient handling of diverse tasks.

2.2.1 Role of the Encoder

The encoder's primary role is to convert the input sequence into a compact, latent representation that encapsulates its semantic and contextual information. This representation acts as a summary of the input, enabling the decoder to generate relevant and accurate outputs based on it. In transformer-based architectures, the encoder is composed of multiple identical layers, each containing self-attention and feedforward networks. The self-attention mechanism allows the encoder to consider all tokens in the sequence simultaneously, capturing both local and global dependencies. This ability to attend to the entire input sequence is particularly valuable in tasks like translation, where understanding context across long sentences is crucial. As the data passes through successive encoder layers, its representation becomes progressively refined, enabling the model to abstract higher-level features. For instance, in language tasks, the initial layers might focus on word-level dependencies, while deeper layers capture sentence-level semantics.

2.2.2 Role of the Decoder

The decoder is responsible for transforming the latent representation generated by the encoder into the output sequence. It does so through a combination of self-attention and cross-attention mechanisms. The self-attention mechanism in the decoder allows it to analyze its own previously generated outputs, ensuring consistency and fluency in the generated sequence. For example, in a translation task, the decoder ensures that the output adheres to grammatical rules and maintains logical coherence with the preceding words. Cross-attention, on the other hand, enables the decoder to align with the encoder's latent representation, ensuring that the generated sequence accurately reflects the input sequence. This dual attention mechanism facilitates a dynamic interplay between the encoder and decoder, ensuring that the output is both relevant and contextually appropriate. Like the encoder, the decoder also incorporates feedforward networks, residual connections, and normalization layers to maintain stability and enhance its representational power.

2.3 Text Decoding Techniques

In natural language generation, text decoding refers to the process of transforming a model's output probabilities into coherent and meaningful sequences of text [19]. This step is especially critical in transformer-based architectures, where the quality of the generated output heavily depends on the decoding strategy employed. The choice of decoding method directly influences the fluency, diversity, and relevance of the output, and it plays a central role in mitigating common issues such as repetition, incoherence, or generic responses.

Recent research has focused on evaluating and improving decoding techniques, particularly in the context of large language models (LLMs). For example, A Thorough Examination of Decoding Methods in the Era of LLMs [20] provides an in-depth analysis of various strategies and their performance trade-offs, shedding light on how different approaches affect the behavior of modern generative models.

2.3.1 Deterministic Approaches

In this section, we focus on two of the most widely used deterministic decoding methods, greedy decoding and beam search. These techniques are foundational for understanding text generation, as they strike different balances between computational efficiency and output quality. Greedy decoding is one of the simplest methods, where at each step, the model selects the most probable token as the next element in the sequence. This method is computationally efficient, as it makes a single deterministic choice per step. However, greedy decoding often results in suboptimal outputs, as it fails to consider alternative sequences. This limitation arises because it makes decisions without considering the broader context, potentially missing out on more fluent or coherent results that might require exploring less probable paths [21]. The advantage of greedy decoding lies in

its simplicity and speed. Since it makes a single deterministic choice at each step, it is computationally efficient and easy to implement. This can be particularly useful in real-time applications or when the generation task requires high consistency. However, its main disadvantage is that it lacks flexibility. Without exploring alternatives, the generated text tends to be more repetitive and sometimes less coherent, especially for complex or long sequences.

Beam search enhances greedy decoding by maintaining a set of candidate sequences, which are explored simultaneously at each step. The algorithm tracks the top w candidates, determined by the beam width (w), and continues exploring the most promising options. This method improves the quality of the generated text, particularly for more complex sequences, as it allows the model to consider multiple possibilities before settling on a final output. However, beam search does not solve the problem of limited diversity. While it explores more options than greedy decoding, it tends to focus on the most probable sequences and may overlook less likely but potentially more creative or interesting solutions [22]. One of the key strengths of beam search is its ability to generate high-quality outputs by balancing exploration and exploitation. It is particularly useful for tasks where longer or more structured outputs are required. Nonetheless, beam search is computationally more expensive than greedy decoding, as it maintains and evaluates multiple candidate sequences at each time step. Additionally, the beam search method may still suffer from a lack of diversity, especially when a large number of candidates are focused on the most probable paths.

2.3.2 Probabilistic Sampling

Probabilistic sampling introduces randomness into the text generation process, allowing models to produce more varied and creative outputs than deterministic approaches like greedy decoding or beam search. Rather than always choosing the most likely next token, these methods sample from a distribution over possible tokens, which helps generate responses that are less repetitive and more engaging—especially in open-ended tasks such as storytelling, dialogue, and creative writing.

Two common techniques in this category are top-k sampling and top-p (nucleus) sampling. In top-k sampling, the model restricts its choices to the k most probable tokens at each step and selects one of them at random according to their normalized probabilities. This limits the chance of producing irrelevant or low-probability words while still introducing variability. The value of k significantly influences the output: a smaller k leads to more focused and predictable text, while a larger k allows for greater diversity but can also result in less coherent results. This balance between control and creativity makes top-k sampling useful in domains where both structure and originality are valued.

Top-p sampling takes a more flexible approach by dynamically choosing the smallest set of tokens whose combined probabilities exceed a predefined threshold p. Unlike

top-k, which always considers a fixed number of tokens, top-p adapts to the shape of the probability distribution at each step. When one token is much more likely than the others, the method becomes more deterministic, producing safer and more consistent text. However, when the distribution is flatter, it includes a wider range of tokens, allowing for more diverse and creative outputs. The threshold p controls how much variation is introduced, with lower values yielding more conservative responses and higher values encouraging greater diversity. This adaptability makes top-p sampling particularly well-suited for applications like conversational AI and poetry generation, where maintaining coherence while encouraging originality is essential.

2.3.3 Temperature Sampling

Temperature sampling is another probabilistic technique that controls the randomness in token selection. By adjusting the temperature T, the model's output distribution is scaled, making the most probable tokens more or less likely to be chosen. A lower temperature (e.g., $T = 0.5$) sharpens the distribution, making the model favor more likely tokens, while a higher temperature (e.g., $T = 1.5$) flattens the distribution, encouraging a more random selection of tokens. This technique offers finer control over the diversity of generated text, allowing the user to either generate more focused, deterministic outputs or more varied, exploratory text [23].

2.4 Multimodal Representation Learning

Recent advancements in artificial intelligence have increasingly focused on models that integrate multiple modalities, such as text, images, audio, and video. Rather than processing each modality in isolation, multimodal embeddings enable the creation of shared semantic spaces where different data types can interact and influence one another. These unified representations have enabled a range of powerful cross-modal applications, including retrieval, classification, and generative modeling. This section explores three key approaches to multimodal representation learning: CLIP, BLIP, and DINO.

2.4.1 CLIP: Contrastive Language-Image Pre-training

CLIP (Contrastive Language-Image Pre-training) [24] is a foundational model in multimodal learning that aligns visual and textual embeddings within a shared semantic space. By training on large datasets of paired image-text data, CLIP learns to associate images with their corresponding textual descriptions using a contrastive loss function. The model architecture consists of two separate encoders, one for images and one for text, that project their respective inputs into the same embedding space. This enables the model to perform a wide variety of cross-modal tasks, such as image-to-text and text-to-image retrieval, as well as zero-shot classification. As illustrated in Figure 2.4, the alignment between modalities allows for effective semantic comparison across data types. Additionally, CLIP's

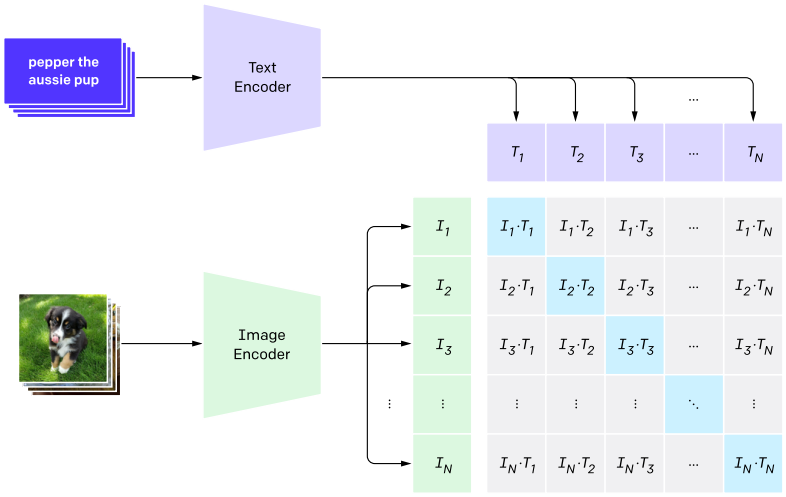


Figure 2.4: CLIP architecture [24]

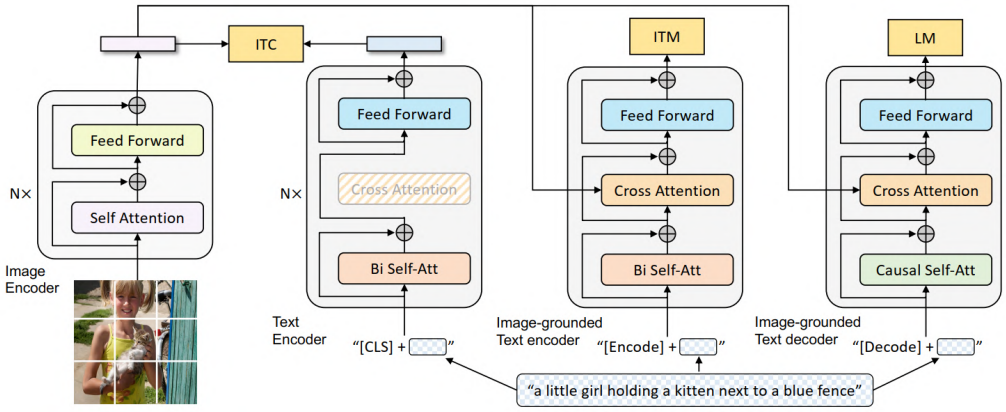


Figure 2.5: BLIP architecture [25]

embeddings are frequently used in generative models, where text embeddings guide image generation processes, such as in diffusion-based architectures, enabling coherent and controllable visual outputs conditioned on natural language prompts.

2.4.2 BLIP: Bootstrapped Language-Image Pre-training

BLIP (Bootstrapped Language-Image Pre-training) [25] extends the capabilities of contrastive models like CLIP by combining vision-language understanding and generation in a single unified architecture. BLIP employs a mixture of encoder-decoder components that support multiple tasks, including contrastive learning, image-text matching, and image-conditioned caption generation. As shown in Figure 2.5, the architecture consists of three main components: a unimodal text encoder, an image-grounded encoder with cross-attention layers, and an image-grounded decoder for generation. The unimodal encoder is trained using Image-Text Contrastive (ITC) loss, which aligns embeddings from the two modalities. The image-grounded encoder is trained with Image-Text Matching (ITM) loss

to determine whether text-image pairs are semantically aligned. For generative tasks, the decoder uses Language Modeling (LM) loss to generate coherent textual outputs based on visual input. BLIP shares parameters between the encoder and decoder, excluding self-attention layers, to enable efficient multi-task learning across retrieval and generation.

Visual Feature Encoders: The Role of DINO

Although not a multimodal model, DINO (Self-Distillation with No Labels) [26] plays an important supporting role in multimodal systems as a powerful visual feature extractor. DINO employs a self-supervised learning strategy in which a student network is trained to match the output of a slowly updated teacher network across multiple augmented views of the same image. This process leads to the learning of high-level, invariant visual features without requiring labeled data.

Despite being trained solely on images, the robustness of DINO's embeddings makes them well-suited for integration into multimodal pipelines [27]. They are often used as visual backbones in systems that require strong image representations, such as image-text retrieval, captioning, or generation. Additionally, DINO's features are particularly effective for tasks involving image-to-image comparison, clustering, and semantic matching, due to their ability to capture meaningful visual similarity.

3

Related Work

This chapter reviews foundational and recent research relevant to this work. It begins with an overview of diffusion models, explaining their underlying mechanisms and key components such as the U-Net architecture and the diffusion process in latent spaces. Following this, recent advances in interpreting and diagnosing diffusion models are discussed, shedding light on their internal workings and improving model transparency. The chapter also covers alternative generative modeling approaches, highlighting their differences compared to diffusion models. Lastly, it examines progress in generating visual sequences, including image sequences and video synthesis, and concludes with a critical summary of the field's current challenges and opportunities.

3.1 Diffusion Models

The concept of diffusion-based generative modeling originated with the work of Sohl-Dickstein et al.[28], who introduced a framework inspired by nonequilibrium thermodynamics. Their method gradually transforms a data distribution into Gaussian noise through a forward diffusion process and learns a reverse process to reconstruct the original data. This foundational idea later evolved into Denoising Diffusion Probabilistic Models (DDPMs)[29], which demonstrated strong generative capabilities in high-resolution image synthesis.

Early applications of diffusion models emerged in the field of medical imaging, where the models' stability and ability to produce high-fidelity reconstructions made them suitable for tasks such as segmentation, reconstruction, and synthesis. The diffusion framework provides a principled and flexible alternative to adversarial approaches like GANs by learning an explicit data likelihood and avoiding mode collapse.

3.1.1 The Diffusion Process

Diffusion models are a class of generative models that synthesize data by learning to reverse a gradual noising process. The fundamental idea is to take a clean data sample and incrementally add Gaussian noise to it over a sequence of time steps, until the sample

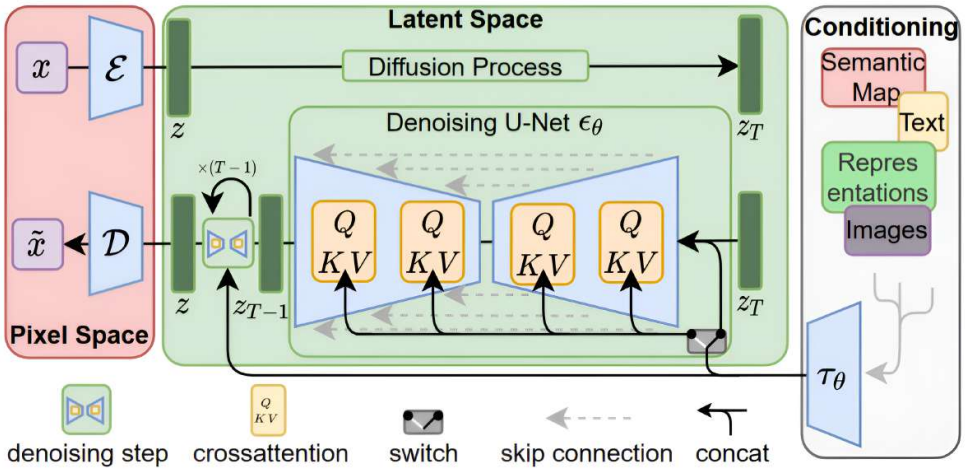


Figure 3.1: Architecture of the Latent Diffusion Model (LDM) [4].

becomes indistinguishable from pure noise. A neural network is then trained to reverse this process, that is, to transform noise back into data by learning how the original signal was corrupted.

In the forward (or diffusion) process, we define a sequence of noisy samples $x_1, x_2, \ldots, x_T$ where each x_t is a slightly noisier version of the previous one. This is formulated as:

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1 - \alpha_t} \epsilon_t, \tag{3.1}$$

where α_t is a predefined noise schedule that controls how much noise is added at each time step, and $\epsilon_t \sim \mathcal{N}(0, I)$ is standard Gaussian noise. By repeatedly applying this process, we can express any noisy sample x_t directly in terms of the original data x_0 and a single noise variable ϵ, as follows:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \tag{3.2}$$

where $\bar{\alpha}_t$ is the cumulative product of the noise schedule up to time t. This form reveals that x_t is a mixture of the original sample x_0 and Gaussian noise ϵ, and that the diffusion process can be seen as gradually obscuring the signal with noise.

The reverse process aims to recover x_0 from x_t by reversing the steps of this noising procedure. Since the exact reverse transition distribution is intractable, we learn an approximate reverse process using a neural network, often a U-Net [30], which predicts the parameters of the distribution $p_\theta(x_{t-1} \mid x_t)$. A common formulation of this is:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{3.3}$$

where the network outputs the mean and variance of a Gaussian distribution that describes how to denoise x_t into x_{t-1}.

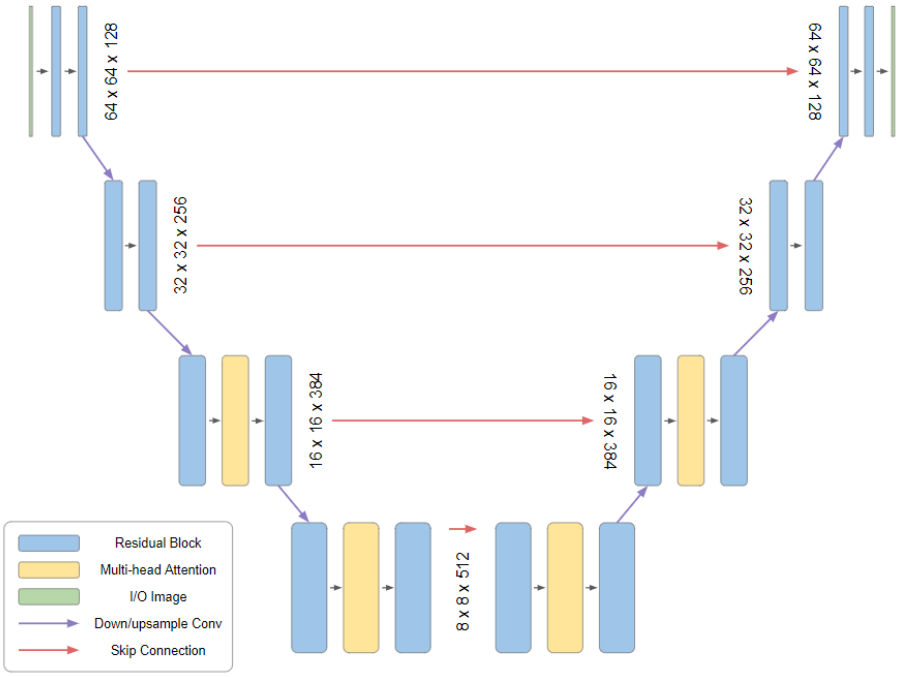


Figure 3.2: U-Net architecture used in the reverse path of denoising diffusion models [31]

3.1.2 Training Diffusion Models

To train the model effectively, we reframe the learning task: instead of predicting the clean sample x_0 directly, we train the model to predict the noise ϵ that was added at a particular time step. This works because the forward process defines a known relationship between x_0, ϵ, and the noisy sample $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and t is a randomly sampled time step.

The model $\epsilon_\theta(x_t, t)$ is trained to approximate the noise that was added at step t. The training objective minimizes the mean squared error between the predicted and true noise:

$$L_{\text{DM}} = \mathbb{E}_{x_0,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]. \tag{3.4}$$

Importantly, the model is trained only to predict the noise corresponding to a specific noise level (timestep t), not to remove all noise in one step. At inference time, the model is used iteratively: starting from pure noise x_T, it gradually denoises the sample step-by-step through the learned reverse process, ultimately producing a realistic sample x_0.

3.1.3 U-Net Architecture for Denoising

A common neural architecture used to parameterize the reverse process in diffusion models is the U-Net, originally proposed for biomedical image segmentation [32]. U-Net follows an encoder-decoder structure, making it well-suited for image-to-image tasks.

The encoder (contracting path) consists of convolutional and downsampling layers that extract multiscale features, while the decoder (expanding path) applies upsampling and

convolution to reconstruct the spatial resolution. Skip connections between corresponding encoder and decoder layers help preserve fine-grained details lost during downsampling.

In generative diffusion models, U-Net is often extended with attention mechanisms, especially cross-attention, to incorporate external conditioning signals such as text prompts. These layers allow the model to align image features with textual semantics, enabling precise control over the image generation process.

3.1.4 Latent Diffusion Models

Latent Diffusion Models (LDMs) [4], illustrated in Figure 3.1, improve computational efficiency by performing the diffusion process in a lower dimensional latent space rather than directly in the pixel space. A *latent* is a compact, abstract representation of an image that captures its essential features, such as structure, texture, and semantic content, while discarding pixel-level details. This representation is obtained by passing the input image I through a pre-trained variational autoencoder (VAE) [33], which encodes the image into a latent vector $z = \mathcal{E}(I)$.

The forward diffusion process is then applied in this latent space, similar to the original pixel-space formulation, but at a much lower dimensionality. At each timestep t, Gaussian noise is gradually added to the latent representation. The noised latent z_t is given by:

$$z_t = \sqrt{\bar{\alpha}_t} z + \sqrt{1 - \bar{\alpha}_t} \epsilon, \tag{3.5}$$

where $\epsilon \sim \mathcal{N}(0, I)$ is standard Gaussian noise. This process results in a smooth degradation of the latent, which is later reversed during generation to reconstruct the image from noise.

The training loss in latent space is defined as:

$$L_{\text{LDM}} = \mathbb{E}_{I, \epsilon \sim \mathcal{N}(0,1), t} \left[\| \epsilon - \epsilon_\theta(z_t, t) \|^2 \right]. \tag{3.6}$$

In cases where the model is conditioned on external inputs, such as a text prompt s, the loss function is adjusted to include the encoded representation of the text. In this scenario, the denoising process is conditioned using a cross-attention mechanism [7] (as detailed in Sec. 2.1.3) within the U-Net architecture[4]. This mechanism enables the model to focus on relevant information from the text prompt during denoising, facilitating the generation of high-quality images based on the provided text descriptions.

The loss function for a conditioned model becomes:

$$L_{\text{LDM}} = \mathbb{E}_{I, s, \epsilon \sim \mathcal{N}(0,1), t} \left[\| \epsilon - \epsilon_\theta(z_t, t, \tau_\theta(s)) \|^2 \right], \tag{3.7}$$

where $\tau_\theta(s)$ represents the embedding of the text prompt s produced by the text encoder. The cross-attention mechanism allows the model to effectively condition the denoising process on the external input, such as a text description, guiding the generation of samples that align with the provided input.

Once trained, the model generates images by starting from random Gaussian noise in the latent space, iteratively denoising it over several time steps. After the denoising

process, the final latent code z_0 is decoded back into the image space using the VAE decoder $\mathcal{D}$, resulting in the generated image:

$$I' = \mathcal{D}(z_0). \tag{3.8}$$

3.2 Advances in Interpreting and Diagnosing Diffusion Models

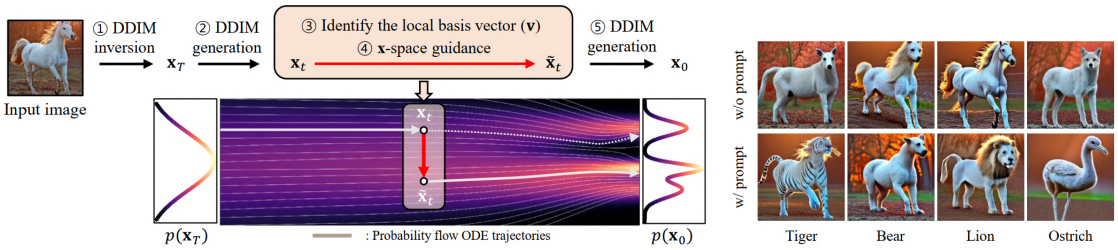
Understanding the inner workings of diffusion models is critical for improving their performance, troubleshooting potential issues, and advancing their capabilities. Recent research has introduced sophisticated interpretability techniques that allow for a much deeper understanding of these models, moving beyond simple visualizations to provide a more rigorous and comprehensive analysis.

A major advancement in interpreting diffusion-based generative models is Differentiable Attention Attribution Maps (DAAM) [34], which provide spatial attributions between individual words in a prompt and specific regions in the generated image. This is achieved by leveraging the cross-attention mechanism embedded within the U-Net component of latent diffusion models. As shown in the Latent Diffusion Model architecture (Figure 3.1), generation takes place in a compressed latent space, where an input noise tensor is iteratively refined into a coherent image through denoising steps. This process is governed by a U-Net (Figure 3.2), which operates at each timestep to predict the denoised latent. To condition the generation on textual input, the U-Net incorporates cross-attention mechanisms that inject text embeddings at multiple spatial resolutions (i.e., at low, mid, and high levels of the U-Net), with each level containing multiple attention heads. DAAM works by capturing and aggregating cross-attention scores across all these layers and heads, accumulating a global attention profile for each word. This means it doesn't rely on a single attention snapshot but instead builds a comprehensive attribution map that reflects how text tokens influence the denoising process over time and across scales. The result is a set of interpretable token-to-pixel heatmaps, where each map reflects the spatial contribution of a specific word. Figure 3.3 shows an example of DAAM applied to the prompt "monkey with hat walking". Each word is associated with its own attention heatmap, clearly revealing how the model localizes concepts such as "monkey" and "hat" within different parts of the generated image. This level of interpretability allows researchers and practitioners to diagnose failures in prompt-image alignment, understand semantic grounding, and gain insight into how diffusion models internalize and spatialize language.

By integrating attention across the hierarchical architecture of the U-Net, DAAM enables a granular and semantically grounded view of multimodal alignment—making it an indispensable tool for analyzing diffusion-based text-to-image systems.

Complementing DAAM, the DiffusionPID technique [35] takes a more nuanced approach by utilizing partial information decomposition (PID). This method dissects the

Figure 3.3: The original synthesized image alongside three DAAM maps highlighting the attention for "monkey", "hat", and "walking" based on the prompt “monkey with hat walking” [34].



(a) Method overview. (b) Editing results.

Figure 3.4: Image editing with the discovered latent basis. (a) Schematic depiction of image editing procedure. 1- An input image is subjected to DDIM inversion, resulting in an initial noisy sample x_T . 2- The sample x_T is progressively denoised until reaching the point t through DDIM generation. 3- Subsequently, the local latent basis $\{v_1, \ldots, v_n\}$ is identified by using the pullback metric. 4- This enables the manipulation of the sample x_t along one of the basis vectors using x-space guidance. 5- The DDIM generation concludes with the progression from the modified latent variable $\tilde{x}_t$. (b) Examples of edited images using a selected basis vector. The latent basis vector could be conditioned on prompts and it facilitates text-aligned manipulations [36].

contributions of individual components within the prompt and provides a detailed analysis of how each element of the text impacts the generated image. DiffusionPID goes beyond DAAM by not only identifying the attention patterns but also by quantifying the information flow within the model. It breaks down the shared, unique, and synergistic contributions of different tokens, allowing researchers to pinpoint which parts of a prompt are most influential in shaping the visual output. This level of decomposition helps in diagnosing potential biases, understanding interdependencies between words, and improving the model’s overall alignment with the user’s intent. By highlighting which words or phrases are responsible for specific features in the image, DiffusionPID provides an essential tool for controlling text-to-image generation, especially in cases where precise output is required, such as in scientific visualization or artistic creation.

Another profound approach for improving model transparency is provided by Riemannian Geometry in the study titled "Understanding the Latent Space of Diffusion Models through the Lens of Riemannian Geometry" [36]. In this paper, the authors argue that the latent space of diffusion models should not be viewed as a simple Euclidean vector space but rather as a non-Euclidean manifold that inherently carries curvature. This

geometric viewpoint opens up new avenues for understanding the complex structure of latent representations within the model. The curvature of the latent space significantly influences how the model navigates the generative process. For instance, the shape and topology of the latent manifold affect how different image features evolve over timesteps and how textual conditioning guides the model's latent transformations. By treating the latent space as a Riemannian manifold, the authors propose novel techniques for manipulating image features with greater precision. This is particularly useful for tasks that require nuanced control over image properties, such as image editing or refining generative outputs according to specific constraints.

In practice, this approach enables the identification of local latent bases vectors that represent key features of the image. The manipulation of these bases allows for text-aligned editing, where image properties can be modified in response to specific prompt conditions, offering better control over the model's output. For example, editing a generated image by modifying a single latent vector aligned with a specific concept, such as "brightness" or "object position" can lead to targeted changes in the image without distorting other aspects of the generated content. The schematic in Figure 3.4a demonstrates this editing process, where the manipulation of latent vectors results in the modification of visual elements based on the user's intent. Furthermore, Figure 3.4b presents an example of image editing results achieved through this method, showcasing how manipulation of latent space vectors leads to specific, text-aligned modifications in the generated image.

By integrating insights from DAAM, DiffusionPID, Riemannian geometry, and guided synthesis techniques, researchers are building a more comprehensive understanding of how diffusion models process and control visual generation. These methods collectively offer powerful tools for debugging, aligning, and refining text-to-image models, ensuring more predictable and semantically faithful outputs. They are particularly valuable in multimodal applications, such as cross-modal retrieval, image captioning, and interactive design, where precise mappings between textual and visual domains are essential.

Expanding this toolbox, Guided Image Synthesis via Initial Image Editing in Diffusion Model [37] presents a hybrid approach that combines low-level user edits with high-level textual prompts to steer the generative process. Unlike techniques that rely solely on text conditioning, this method introduces spatially grounded edits to an initial image, which are then refined through diffusion to align with the semantic intent of the prompt. This dual-conditioning strategy provides a more intuitive and interpretable interface for users, anchoring generation in both visual and linguistic cues.

As shown in Figure 3.5, the model architecture is designed to preserve user-edited regions while incorporating textual guidance throughout the denoising process. By explicitly conditioning the generative trajectory on both modalities, the framework offers improved controllability and a transparent mapping between user intent and output synthesis.

Together, these interpretability-driven techniques bridge the gap between opaque

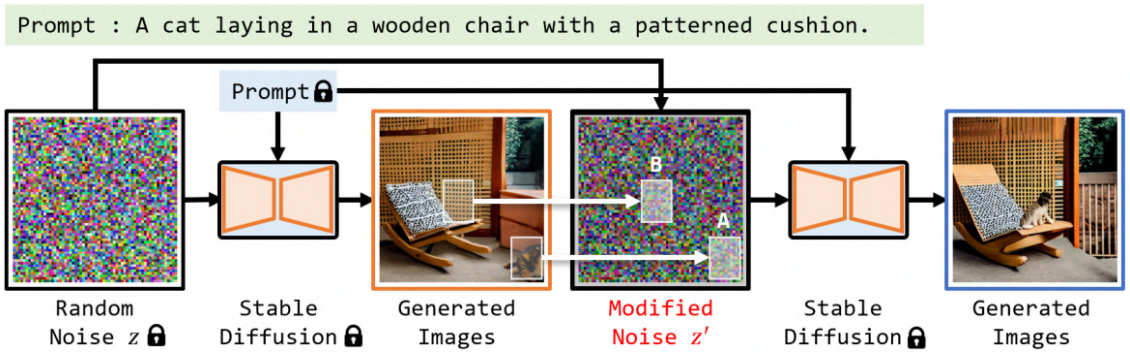


Figure 3.5: Architecture of guided synthesis via initial image editing. The diffusion process incorporates both user-edited image input and textual prompts to control the generative trajectory [37].

model decisions and user understanding, enabling more trustworthy, precise, and user-controllable generative AI systems.

3.3 Alternatives to Diffusion Models

In addition to diffusion models, other generative approaches such as Generative Adversarial Networks (GANs)[38], Variational Autoencoders (VAEs)[39], Autoregressive models [40], and Diffusion Transformers [41] offer compelling alternatives for data generation, each with unique strengths and limitations.

GANs consist of two competing networks: a generator that creates synthetic data and a discriminator that evaluates its authenticity. GANs are known for generating high-quality images, but they suffer from training instability, often facing challenges like mode collapse, where the generator produces limited variations of output.

VAEs work by probabilistically encoding data into a latent space and then decoding it back to the original data. VAEs offer more stable training compared to GANs but tend to produce blurrier images due to the trade-off between optimizing for the latent space structure and reconstruction quality. This trade-off is influenced by the KL divergence term, which encourages a distribution that closely matches a prior, potentially sacrificing image sharpness for latent space regularity.

Autoregressive models generate data sequentially, often one pixel or token at a time, modeling the conditional probability distribution of each element given the previous ones. While these models can generate highly detailed images, they are computationally expensive during sampling because of their sequential nature, which requires more time to generate a complete image compared to other models.

A promising emerging approach is the diffusion transformer, which combines the principles of diffusion models with transformer-based architectures. Diffusion transformers integrate the iterative denoising process of diffusion models with the efficiency and scalability of transformers. This hybrid approach leverages the transformer's self-attention

mechanism and parallelization capabilities, improving the generation process and reducing the computational inefficiencies traditionally associated with diffusion models. Although diffusion transformers show great potential in enhancing output quality and sampling speed, they are still computationally intensive, particularly for complex tasks requiring numerous iterations.

Each of these models has distinct advantages and limitations depending on the task at hand. GANs excel at generating high-quality, diverse outputs but are difficult to train. VAEs provide stable training but often produce lower-quality, blurrier images. Autoregressive models offer highly detailed results but are slow due to their sequential nature. Diffusion models strike a balance between flexibility and high-quality generation but are computationally expensive and slower. Finally, the diffusion transformer combines the strengths of both diffusion models and transformers, providing an exciting direction for future generative model development, though it also faces its own set of computational challenges.

3.4 Generation of Visual Sequences

While diffusion models excel in generating high-quality static images, extending them to generate sequences of images, such as animations, videos, or multi-frame visual narratives, introduces new complexities [42–45]. The challenge lies in maintaining temporal consistency and visual coherence across frames, ensuring that each frame aligns with the narrative and evolves smoothly. Unlike single image generation, where the focus is on the fidelity of a single output, sequential image generation requires the model to ensure that characters, objects, and environments maintain continuity over time.

In this section, we explore various techniques and models that address these challenges. From frameworks that condition each frame on its predecessors to methods that enforce temporal alignment and consistency across scenes, we look at how these innovations improve the generation of dynamic and coherent visual sequences. This includes advancements like autoregressive latent diffusion models (AR-LDMs), memory-based architectures, and novel approaches for multi-scene video generation, all of which work to enhance the temporal and contextual coherence necessary for producing high-quality visual narratives.

3.4.1 Generating Sequences of Images

An emerging direction in vision-language modeling is the integration of image and text modalities into a unified generative framework. GILL [46] exemplifies this trend by extending a large pretrained language model with special `[IMG]` tokens, enabling joint generation of interleaved text and images within a single sequence. As shown in Figure 3.6, GILL handles multimodal generation by deciding at each `[IMG]` position whether to retrieve a relevant image or generate a new one.

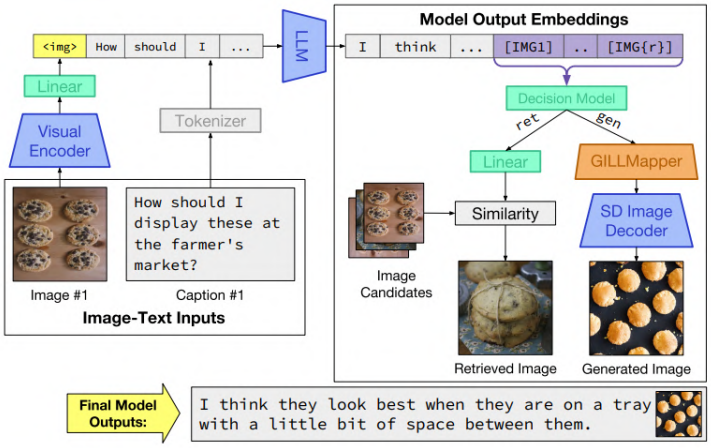


Figure 3.6: Inference time procedure for GILL. The model takes in image and text inputs, and produces text interleaved with image embeddings. After deciding whether to retrieve or generate for a particular set of tokens, it returns the appropriate image outputs. [46]

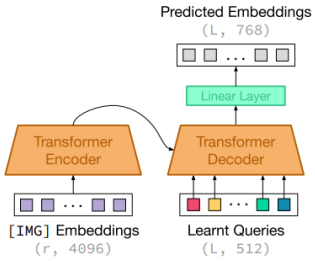
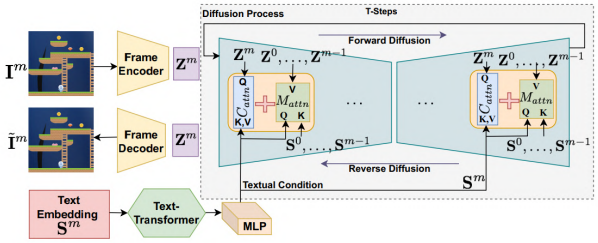
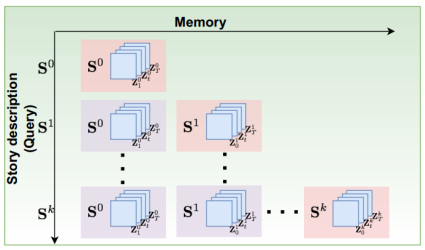


Figure 3.7: GILLMapper model architecture. It is conditioned on the hidden [IMG] representations and a sequence of learnt query embedding vectors. [46]



(a) Story-LDM architecture for conditional story generation.

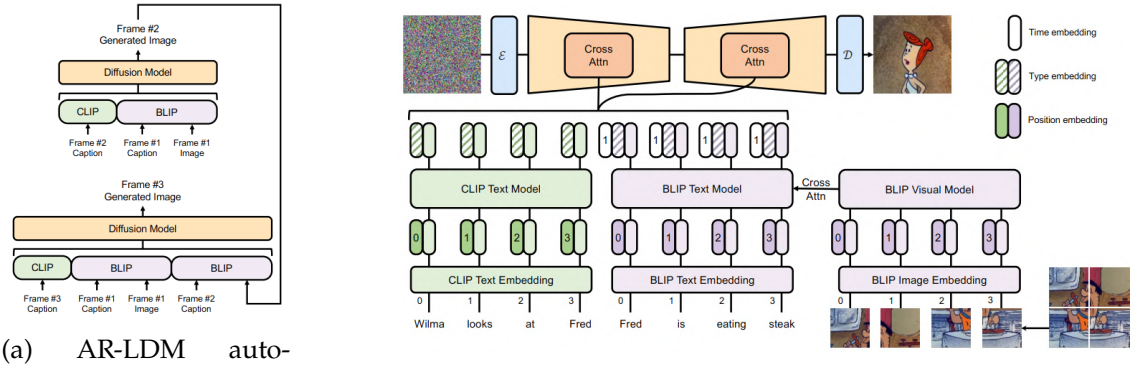


(b) Memory state with query and values for our memory-attention module.

Figure 3.8: Illustration of the Make-A-Story model [47].

Rather than fine-tuning the full model, GILL freezes both the core language and vision components and only trains a small number of lightweight parameters: linear projection layers that map image features into the language embedding space and trainable [IMG] token embeddings. To support image generation, GILL introduces a compact transformer module called GILLMapper (Figure 3.7), which transforms [IMG] token embeddings into latents compatible with a pretrained text-to-image generator such as Stable Diffusion. This enables high-quality image synthesis directly from language model outputs.

While GILL focuses on token-level integration of images and text, other models address the more specific challenge of generating coherent sequences of images conditioned on narrative input. To this end, the *Make-A-Story* framework [47] enables sequential image generation while ensuring a coherent narrative flow. The framework integrates both textual and visual information throughout the generation process. It begins by encoding a global story representation using a transformer-based text encoder, capturing key themes and structure from the input text. This high-level representation guides the generation of each image, ensuring that the images stay consistent with the overarching narrative.



(a) AR-LDM auto-regression process.

(b) AR-LDM model architecture.

Figure 3.9: Illustration of the AR-LDM model [49].

Rather than generating each image independently, *Make-A-Story* ensures that each new image is informed by its predecessors, preserving continuity in characters, objects, and environments. This is achieved by maintaining a shared latent space, which allows earlier visual elements to influence the generation of new ones. This approach prevents abrupt transitions and maintains smooth progression across frames. By conditioning each image on both previous visual latents and evolving textual descriptions, *Make-A-Story* effectively aligns the generated visuals with the story, creating a structured and coherent sequence.

However, while *Make-A-Story* preserves narrative consistency, it does so by conditioning each new frame on all past latent representations (Figure 3.8b). This may introduce unnecessary complexity, as not all past latent representations are equally important for the generation of each new frame. Bordalo et al. [48] showed that only earlier latent representations contain crucial generative information, as later latents become too conditioned, reducing their ability to flexibly generate new content. By focusing on the most informative early latent representations and discarding later ones, Bordalo et al. show that the model can streamline the process, improving both efficiency and coherence across the sequence.

In addition to structured narrative models like Make-A-Story, another promising approach is the use of *autoregressive latent diffusion models* (AR-LDM) [49], which maintain consistency across a sequence by conditioning each generated frame on its predecessors. This approach ensures smooth transitions and temporal coherence by modeling the conditional dependencies between frames. As illustrated in Figure 3.9b, AR-LDM consists of a conditioning network that integrates CLIP and BLIP to maintain both contextual and visual consistency. CLIP encodes textual descriptions, ensuring each generated frame aligns with the provided captions, while BLIP captures cross-modal dependencies by jointly encoding previous image-caption pairs. This allows the model to reference past scenes, maintaining consistent character appearances, object placements, and visual elements across the sequence. By leveraging these multimodal embeddings, AR-LDM effectively grounds new frames in the context of previous ones, improving coherence in sequential image generation.

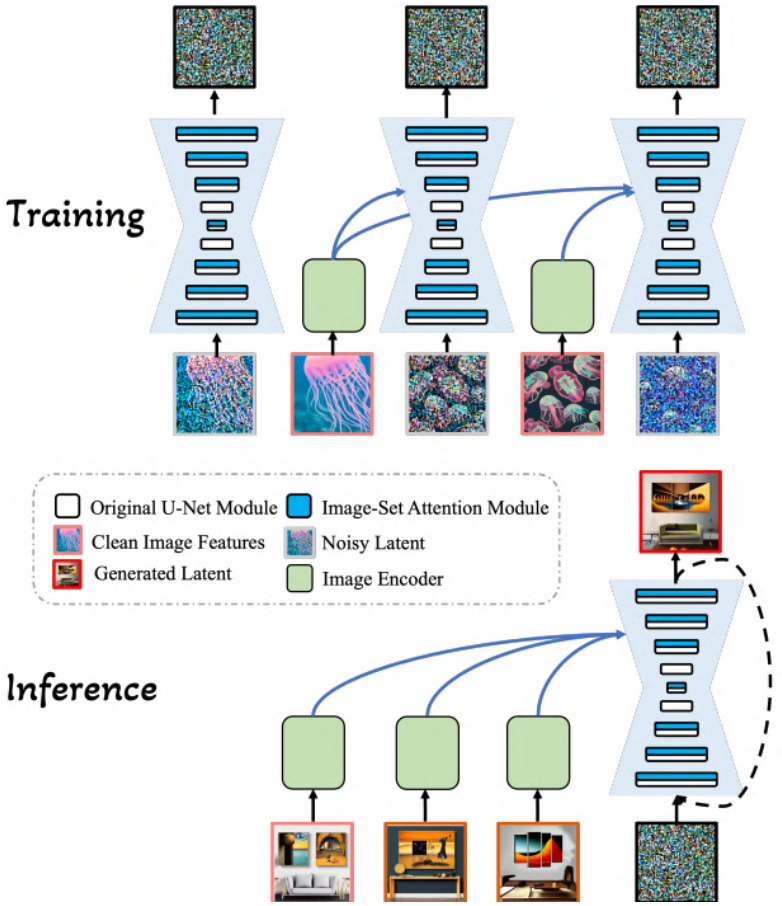


Figure 3.10: Overview of the Many-to-Many (M2M) auto-regressive diffusion pipeline [50]. During training, the model is provided with sets of noised latent images and their corresponding clean image features, learning to predict the noise added to each noised image conditioned on preceding clean features. At inference time, the model generates a coherent set of images in an autoregressive manner by iteratively incorporating previously generated images back into the input.

Despite its strengths, AR-LDM also has limitations. While its autoregressive approach (Figure 3.9a) helps maintain temporal consistency, it can lead to error accumulation—small inconsistencies in early frames may propagate over time, resulting in visual artifacts or a gradual loss of coherence, particularly in extended sequences.

To address this issue, recent research has explored alternative designs that reduce reliance on strict autoregression. One such advancement is the *Many-to-Many Image Generation* framework (M2M) [50], which introduces an autoregressive latent diffusion strategy focused on modeling inter-image relationships directly in the visual latent space. Unlike approaches that depend heavily on memory modules or textual prompts, M2M leverages a novel Image-Set Attention mechanism to capture contextual dependencies across sets of related images (see Figure 3.10). This mechanism enables the model to generate each image based on a flexible and dynamic understanding of the set as a whole, rather than relying on a fixed sequence of prior frames.

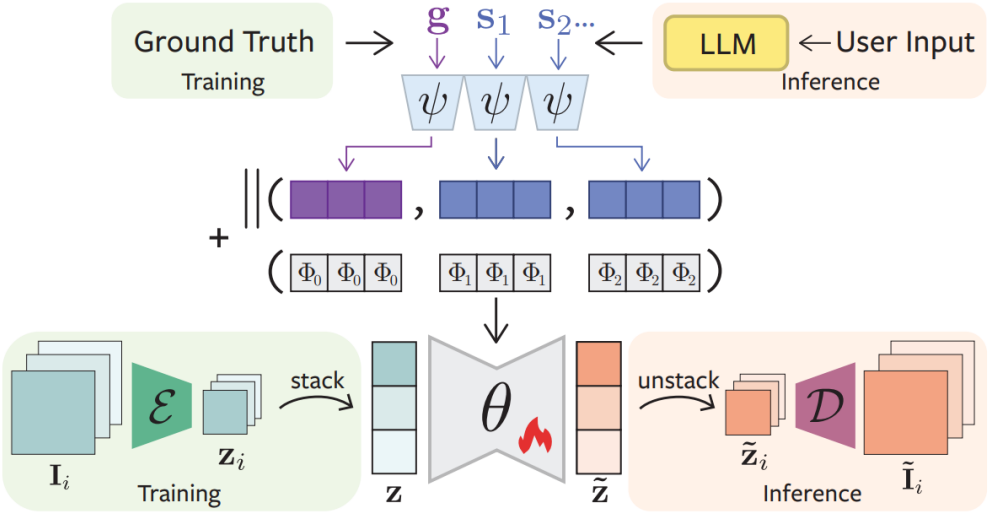


Figure 3.11: Architecture of the Generating Illustrated Instructions model [51].

As a result, M2M facilitates coherent visual generation in a range of tasks, including multi-view synthesis, consistent scene illustration, and sequential storytelling, without the need for explicit narrative structure. By avoiding over conditioning on the full generation history, it effectively mitigates the compounding error problem observed in strictly autoregressive models like AR-LDM, while maintaining strong semantic and visual consistency throughout the sequence. This makes M2M a robust and scalable alternative for structured image set generation in both text-free and text-conditioned settings.

StackedDiffusion [51], a latent diffusion-based model designed for simultaneous generation of multi-step instructional images. Rather than autoregressively producing one image at a time, StackedDiffusion encodes the goal and each step description separately using a frozen text encoder, appending positional embeddings to preserve semantic order. These text embeddings condition a single, spatially stacked latent tensor, which is decoded using a standard diffusion U-Net, as we can see in Figure 3.11. This tiling strategy enables the model to leverage intra-image coherence priors for inter-image consistency, effectively treating the sequence of step images as a large composite image during both training and inference.

However, all the above models share a common limitation, they require significant computational resources during training due to the complexity of diffusion objectives, multimodal conditioning, and the scale of the underlying architectures.

3.4.2 Multi-Shot Video Generation

For improving multi-scene video generation, techniques such as TALC (Time-Aligned Captions for Multi-Scene Text-to-Video Generation) [52] enforce temporal alignment between scenes, offering a solution to the challenges faced by auto-regressive models. TALC ensures that each scene in the generated video smoothly transitions to the next,

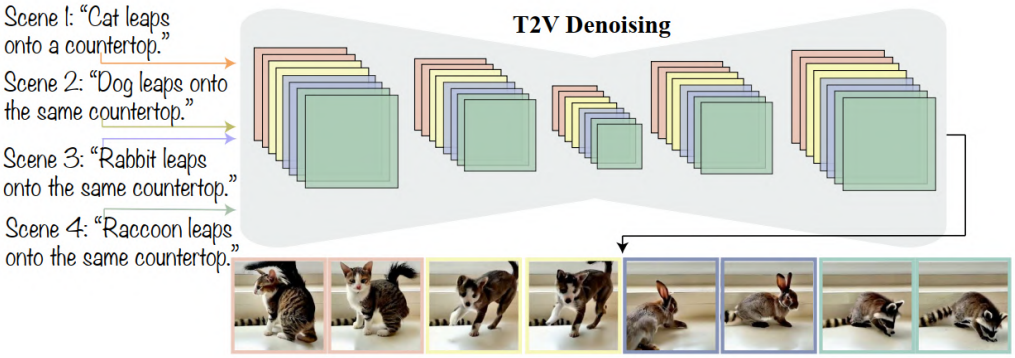


Figure 3.12: TALC architecture. [52]

creating a more coherent and visually consistent multi-scene video. The architecture of TALC, Figure 3.12, incorporates an innovative temporal alignment mechanism that aligns each textual caption to the corresponding scene in the video, while simultaneously maintaining consistency between scenes.

TALC modifies the standard text-to-video architecture by incorporating a two-step process: the first step involves encoding the textual description of each scene using pre-trained vision-language models, such as CLIP, to obtain a robust multimodal representation. The second step applies a temporal alignment mechanism to synchronize the visual features from each scene with the corresponding textual descriptions, ensuring a seamless flow from one scene to the next. This approach improves the ability of the model to generate videos with consistent visual entities, backgrounds, and character appearances across multiple scenes. Similarly, Align Your Latents [53] and NUWA-XL [54] utilize sparse keyframe generation and interpolation to enhance frame continuity and fluidity, ensuring smooth transitions across generated images.

A recent advancement in video generation is Contrastive Sequential Diffusion Learning (CoSeD) [55], which introduces an architecture designed to improve temporal consistency and semantic alignment across multiple scenes. Unlike standard diffusion models that generate each frame independently, CoSeD explicitly models inter-frame relationships to ensure coherence throughout a video.

As shown in Figure 3.13, CoSeD combines two main components: sequential conditioning and contrastive selection. During the denoising process, latent features from previously generated frames are used to guide the generation of new frames, replacing the usual random noise initialization. This encourages temporal continuity between frames.

For each step, multiple candidate frames are generated using different conditioned latent seeds. A contrastive module then selects the most coherent candidate by comparing it with the past context in a joint vision-language embedding space. This ensures the selected frame aligns semantically with the current step and remains visually consistent with earlier frames. Through this architecture, CoSeD enables more stable and context-aware multi-scene video generation.

Other models incorporate self-attention mechanisms to strengthen frame-to-frame

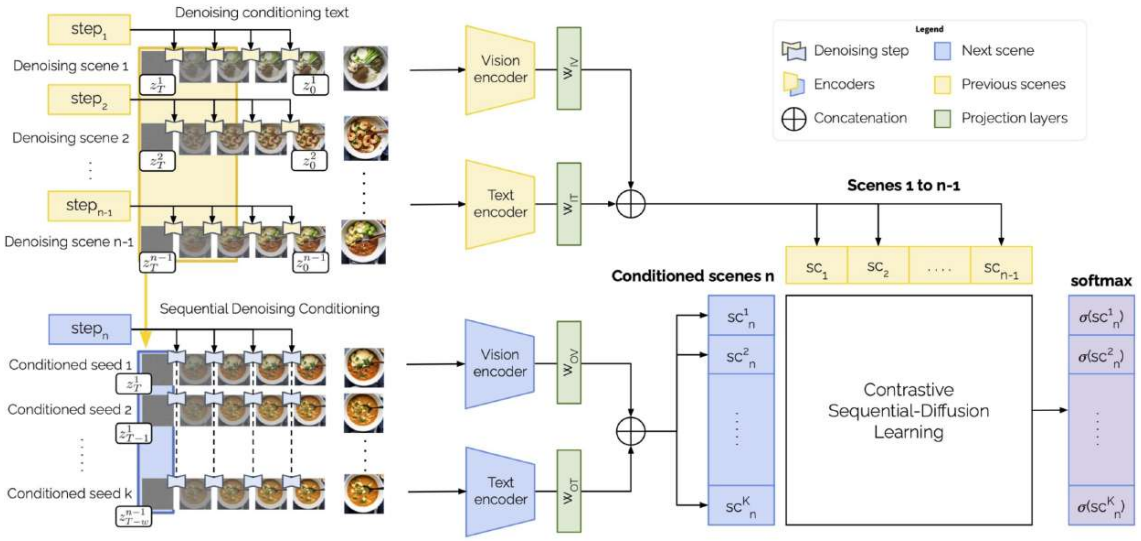


Figure 3.13: Architecture of Contrastive Sequential Diffusion Learning (CoSeD) [55], integrating sequential conditioning and contrastive learning to enhance multi-scene video generation.

consistency. StoryDiffusion [57] utilizes self-attention to facilitate seamless transitions, making it particularly effective for context-driven storytelling applications. A slightly different approach is taken by VideoDirectorGPT [56] pushes the boundaries of structured video generation by integrating multimodal inputs in a sophisticated pipeline-based architecture. This approach enables the generation of coherent and contextually rich videos with improved consistency across scenes. As illustrated in Figure 3.14, the architecture of VideoDirectorGPT begins with video planning by using a large language model (LLM) to create a detailed video plan. The video plan consists of four main components: (1) multi-scene descriptions, (2) entity names and their bounding boxes, (3) background descriptions, and (4) consistency groupings that indicate which entities should remain visually consistent across scenes. This plan is generated in two steps. In the first step, the LLM generates scene descriptions, identifies entities, and groups them based on consistency across scenes. In the second step, the model assigns bounding boxes to each entity within each scene, ensuring that their spatial layout is well-defined. The model generates these layouts for multiple frames in each scene, interpolating between them to create smoother transitions for the final video sequence. Once the video plan is created, VideoDirectorGPT proceeds to the video generation stage using a technique called Layout2Vid, which guides the video generation process based on the detailed layout provided by the video plan. Layout2Vid builds upon a pre-existing T2V (Text-to-Video) model like ModelScopeT2V [58], but introduces explicit layout control by incorporating bounding boxes and entity information into the diffusion process. The model's core mechanism uses guided 2D attention that takes both text descriptions of scenes and the

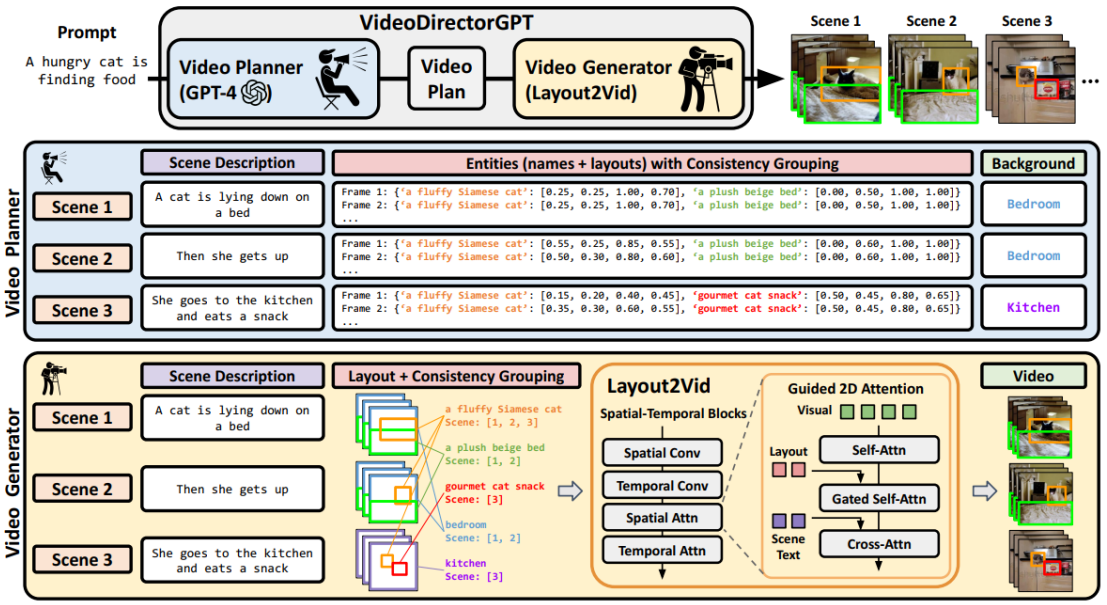


Figure 3.14: Overview of the VideoDirectorGPT architecture. In the first stage, an LLM is used as a video planner to create a detailed video plan, providing an overarching plot for multi-scene videos. In the second stage, Layout2Vid, a grounded video generation module, renders the video based on the video plan, ensuring spatial and temporal consistency across scenes [56].

grounding of entities based on visual and textual embeddings. By using both CLIP image and text embeddings for entity grounding, VideoDirectorGPT ensures that each entity remains visually consistent across frames, which is crucial for maintaining continuity in longer sequences. This design allows VideoDirectorGPT to generate high-quality videos that are not only contextually coherent but also temporally consistent, with smooth transitions between scenes and entities that remain visually consistent throughout. By using multimodal inputs and detailed scene-level control, VideoDirectorGPT extends the capabilities of diffusion models to structured video generation, making it an ideal tool for complex video creation tasks where narrative coherence and visual consistency are essential.

These advancements collectively push the boundaries of diffusion models for sequential image generation. By addressing the challenges of continuity and coherence, these methods enable new applications in animation, video generation, and multi-frame storytelling. As research progresses, integrating these techniques may lead to even more sophisticated systems capable of generating fluid, contextually rich visual narratives.

3.5 Critical Summary

Despite significant advancements in diffusion models, a fundamental gap remains in understanding their latent space. While latent representations are central to guiding image generation, their structure, evolution during training, and influence on coherence

across sequential outputs are not well characterized. This lack of interpretability poses critical challenges in applications that demand visual consistency and control, such as multi step or temporally aware image generation. Compounding the issue is the training complexity of diffusion models themselves. These models require extensive computational resources, long training times, and careful noise scheduling to achieve high-fidelity results. Yet, despite this investment, the internal representations they learn, particularly in the latent space, remain largely opaque. This opacity limits our ability to diagnose failure modes, improve sample efficiency, or generalize across tasks. It also inhibits the design of more structured or modular training objectives that could promote latent disentanglement or semantic alignment.

Prior research has explored techniques such as latent reuse, shared conditioning vectors, and consistency losses to address temporal or cross-frame stability. However, these approaches often operate without a principled understanding of which latent features encode durable semantics versus those that amplify variation or noise. As a result, strategies for leveraging latent information are largely heuristic and can fail under distributional shifts or when generalizing to more structured or compositional generation tasks. This challenge becomes especially acute when generating structured sequences, such as storyboards, animation frames, or visual narratives, where preserving both local consistency (e.g., between adjacent frames) and global coherence (e.g., character identity, spatial layout, object permanence) is essential. Current models may preserve low-level texture or style while exhibiting semantic drift over time, undermining the reliability of generation in downstream applications.

Overcoming these limitations will require moving beyond black-box training regimes toward a more principled understanding of latent dynamics and structure. This includes developing interpretability tools, diagnostics for latent consistency, and perhaps new training paradigms that explicitly encourage coherence-preserving features in the latent space. Only through such efforts can diffusion models achieve robust, controllable, and semantically faithful generation in multi-step or structured image synthesis tasks.

4

BeamDiffusion Model

4.1 Problem Statement

Generating a full sequence of visually coherent images from textual scene descriptions is a challenging research problem. Given a sequence of steps $S = \{s_1, s_2, \ldots, s_L\}$, where each step s_j is a textual description, our goal is to generate a corresponding sequence of images $\{I_1, I_2, \ldots, I_L\}$ that maintain visual and semantic consistency across the entire sequence.

Latent Diffusion Models (LDMs) [4], described in Section 3.1.4, generate high-quality images in a compressed latent space by encoding an input image I into a latent z via an encoder $\mathcal{E}$, and reconstructing through a decoder $\mathcal{D}$. Starting from a random latent $z_T \sim \mathcal{N}(0, \mathbf{I})$, a function ϵ_θ, implemented as a U-Net [30], learns how to iteratively estimate the Gaussian noise present in z_t, conditioning on textual embeddings $\tau_\theta(s)$ via cross-attention [7].

The training objective minimizes the mean difference between the noise ϵ added during the diffusion process, and the estimated noise ϵ_θ:

$$L_{\text{LDM}} = \mathbb{E}_{\mathcal{E}(I),s,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(s))\|_2^2\right], \tag{4.1}$$

where each denoising process t is treated independently. This independence complicates maintaining consistency across sequences of related prompts or images, often causing visual discrepancies in character identity, lighting, or scene layout.

Such inconsistencies pose significant challenges in applications requiring temporal or narrative coherence, especially in non-linear narratives [59] where context must persist despite jumps in time or perspective. Hence, while LDMs excel at individual image generation, extending them to coherent sequential generation necessitates methods that incorporate inter-step dependencies or global context to preserve visual and semantic continuity.

Prior work [47, 48, 55, 60, 61] has attempted to address these challenges by reusing latent denoising iterations from earlier steps in the sequence. As discussed in Section 3.1.4, these latents are intermediate representations in a compressed space where the diffusion process operates. However, a key open question remains: *how can we determine the optimal*

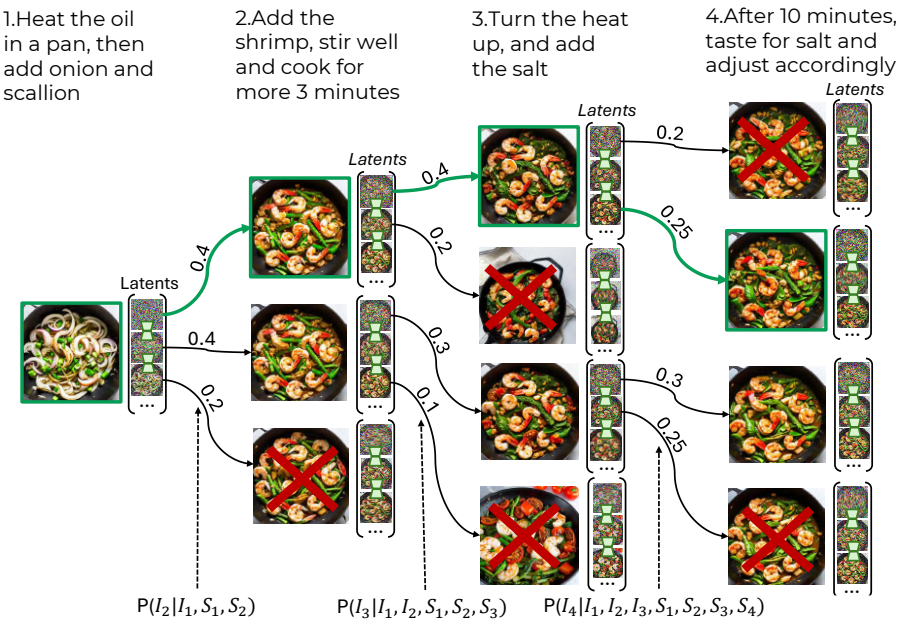


Figure 4.1: BeamDiffusion performs tree-structured search in latent space, maintaining multiple denoising trajectories per step to identify the most coherent sequence. At each step S_n, candidate latents are explored and pruned based on their alignment with the current prompt and prior image context.

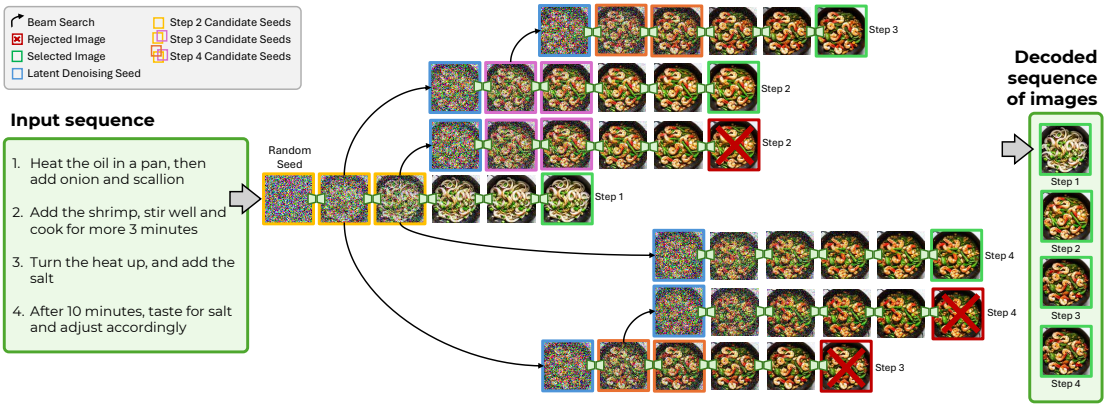
set of latent denoising paths that maximize coherence across the entire image sequence? The answer lies in efficiently exploring a large combinatorial search space of possible latent trajectories.

Our proposed method, BeamDiffusion, tackles this problem by incorporating a beam search strategy that traverses the latent denoising space to find an optimal sequence of latent representations. This approach enables iterative refinement and selection of the most coherent paths, thereby enhancing consistency and quality across the generated image sequence. The next sections will describe the details of this search strategy and its integration within the diffusion process.

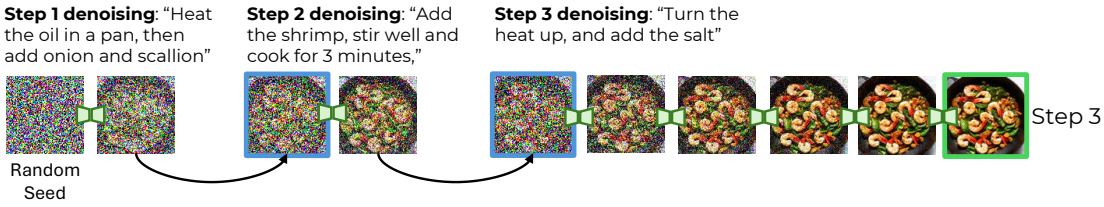
4.2 Beam Diffusion Model

To address the limitations of standard Latent Diffusion Models (LDMs) in generating coherent image sequences, we introduce the BeamDiffusion model. The proposed approach enhances both consistency and quality by adopting a strategy inspired by beam search. Unlike traditional beam search, described in Section 2.3.1, which operates in discrete token spaces, our method explores the continuous latent denoising space of the diffusion model.

As illustrated in Figure 4.1, BeamDiffusion constructs a tree of candidate images, where each node represents a fully generated image resulting from a complete latent denoising trajectory. Each branch reflects a different path through the latent space, capturing distinct generation outcomes. At every step in the input sequence, the model expands the most promising branches by generating new image candidates conditioned on the next textual prompt. These candidates are then evaluated based on two key criteria: semantic alignment



(a) The beam denoising search tree.



(b) Example of one complete decoded denoising process using contextual captions.

Figure 4.2: The beam diffusion model works in the denoising latent space by denoising the initial seed to latents and them decoding different beam paths. This shows how our method evolves in the latent space and how it can explore the latent space.

with the current prompt and visual consistency with previously selected images in the sequence. By scoring and pruning the candidate set at each step, BeamDiffusion retains only the most coherent image denoising trajectories. This approach allows the model to preserve important visual details, such as character appearance, spatial layout, and background elements, while dynamically adapting to changes in the scene description.

Overall, this search-based formulation addresses the typical inconsistencies observed in sequential image generation. It offers a principled and flexible framework for generating temporally and semantically coherent image sequences. For clarity, throughout this thesis, the term step refers to an element in the input sequence (e.g., a scene description), while iteration refers to a single denoising step within the diffusion process.

4.2.1 Latent Space Exploration with Diffusion Beams

To generate coherent and diverse image sequences, we explore the latent space of diffusion models using a beam search-inspired strategy. For each sequence step j, we generate a set of candidate images $\mathcal{I}_j = \{I_j^{(1)}, I_j^{(2)}, \dots\}$, where each $I_j^{(k)}$ is conditioned on a denoising latent from a previous generation.

Algorithm 1 BeamDiffusion Algorithm

1: **Input parameters:**
$\{s_1, \ldots, s_L\}$: sequence of steps; r: number of random seeds; m: steps back; w: beam width.
2: Initialize beam: $\hat{B}_1 \leftarrow \{\text{SD}(s_1, \text{rnd})\}^r$ ▷ Generate r different images for first step.
3: **for** $j = 2$ **to** L **do**
4: **for all** beam $b \in \hat{B}_{j-1}$ **do**
5: $\mathcal{Z}_j \leftarrow \{\hat{B}_{j-m}, \ldots, \hat{B}_{j-1}\}$ ▷ Retrieve latent denoising vectors from previous steps.
6: **for all** latent $z \in \mathcal{Z}_j$ **do**
7: $I_j \leftarrow \text{SD}(s_j, z)$ ▷ Generate candidate image conditioned on latent z.
8: $\hat{B}_{j-1} \leftarrow \hat{B}_{j-1} \cup \{I_j\}$ ▷ Extend beam with new candidate image.
9: **end for**
10: **end for**
11: Compute scores: $\text{score}(I_j) \leftarrow \varphi(I_j \in \hat{B}_j)$ ▷ Score images based on coherence and prompt alignment.
12: Prune beams: $B_j \leftarrow \text{argmax}_{|B_j|=w} \text{score}(\hat{B}_j)$ ▷ Keep top w beams.
13: **end for**
14: **return** best beam $B^* \leftarrow \text{argmax}_{B_L} \text{score}(B_L)$

4.2.1.1 Sampling the Denoising Latent Space

In our approach, the denoising process for each sequence step is conditioned by latent representations from prior steps, which helps the model maintain coherence across the generated sequence.

On the first step s_1, we generate multiple candidate images by running the diffusion process with r random noise seeds (Alg. 1, line 2). This enables a broader exploration of the latent space and reduces Bayes risk [62], as discussed in Section 4.4. For every generation at sequence step s_i, we store the first N latent vectors $Z_i = \{z_{i,t}\}_{t \in \subseteq \{1,\ldots,N\}}$ produced during the denoising iterations.

At every subsequent step $j > i$, we generate diverse samples that (a) explore different denoising trajectories in the latent space, and (b) are strongly correlated with previous samples. To achieve this, we leverage the stored latent vectors from the previous m steps, aggregate their latent representations as:

$$\mathcal{Z}_j = \bigcup_{i \in \{j-m,\ldots,j-1\}} \{Z_i\}. \tag{4.2}$$

To generate new candidate images $\mathcal{I}_j = \{I_j^{(1)}, \ldots, I_j^{(k)}, \ldots, I_j^{(N \cdot m)}\}$ at step j, we condition the diffusion model on the latent trajectories in $\mathcal{Z}_j$. Specifically, for each latent trajectory $z \in \mathcal{Z}_j$, the model performs a new diffusion process to produce a corresponding candidate image $I_j^{(k)}$ (Alg. 1, lines 5 and 7). This establishes a dependency between generations, where latent vectors from earlier steps guide the synthesis of visually coherent transitions over time, as illustrated in Figure 4.2. Rather than conditioning each image solely on its corresponding prompt, our model leverages a temporally interleaved mixture of prompts

from multiple preceding steps. Specifically, the denoising process for a given image incorporates guidance from earlier prompts alongside the current one, allowing the model to anchor its generations in a broader temporal context. For example, as shown in Figure 4.2b, the image for step 3 is synthesized using a sequence of prompts from step 1, step 2, and finally step 3. This compositional prompt strategy encourages richer temporal coherence and stylistic consistency, helping retain key attributes such as character appearance, object layout, or lighting across the sequence. As a result, the candidate set $\mathcal{I}_|$ spans a diverse yet coherent range of plausible continuations, each reflecting different contextual interpretations of the current scene prompt.

4.2.1.2 Diffusion Beams

We introduce *diffusion beams* as a structured search mechanism to guide sequential image generation over time. This approach is inspired by the beam search strategy commonly used in sequence generation tasks, where multiple candidate sequences (beams) are maintained and iteratively extended to explore a wider space of possibilities [63]. In the context of diffusion models for image generation, we adapt this idea to operate over sequences of images rather than tokens.

Let $\hat{B}_{j-1} = \{\hat{B}_{j-1}^{(1)}, \hat{B}_{j-1}^{(2)}, \ldots, \hat{B}_{j-1}^{(B)}\}$ denote the set of current beams at sequence step $j-1$, where each beam $\hat{B}_{j-1}^{(b)}$ is a sequence of images generated so far. At the next sequence step j, the model generates a set of candidate images $\mathcal{I}_j = \{I_j^{(1)}, I_j^{(2)}, \ldots, I_j^{(N \cdot m)}\}$, which represent possible continuations of the sequence for beam b.

To form the new set of beams $\hat{B}_j$, we extend each current beam $\hat{B}_{j-1}^{(b)}$ by appending each of the $N \cdot m$ candidate images. This results in $B \cdot (N \cdot m)$ possible new beams:

$$\hat{B}_j = \bigcup_{(b,k)} \left\{ \hat{B}_{j-1}^{(b)} \oplus I_j^{(b \cdot k)} \mid I_j^{(b \cdot k)} \in \mathcal{I}_j \right\}, \tag{4.3}$$

where $\oplus$ denotes the concatenation operation, i.e., adding the new image $I_j^{(b \cdot k)}$ to the end of beam $\hat{B}_{j-1}^{(b)}$. Each resulting beam in $\hat{B}_j$ represents a possible image sequence up to step j.

This beam extension process can be viewed as constructing a tree over time,(Alg. 1, line 8), where each beam at step $j-1$ branches into $N \cdot m$ possible continuations at step j. The key idea behind diffusion beams is to avoid premature commitment to a single generation path. By explicitly maintaining multiple possible sequences, the model is able to explore diverse continuations while also reinforcing coherent and high-quality trajectories. This is especially important in sequential generation tasks, where early decisions can strongly constrain the final output. The diffusion beams approach helps mitigate this issue by preserving flexibility over time and allowing the model to refine or redirect its generative path as new information becomes available.

4.2.2 Pruning Diffusion Beam Paths

Tracking multiple *diffusion beams* enables BeamDiffusion to maintain visual coherence while exploring diverse continuations across a sequence. However, as the sequence length increases, the number of candidate beams grows combinatorially, making it computationally infeasible to retain and evaluate all possible trajectories. To manage this complexity, we apply a pruning strategy based on a learned contrastive scoring function, which allows the system to retain only the most promising beams while still preserving diversity in generation, i.e., the red crosses in Figure 4.1. This strategy effectively balances exploration and exploitation, encouraging diversity by considering multiple candidate directions, while focusing computational resources on the most semantically and visually coherent trajectories.

At each sequence step j, we construct a set of candidate beams $\hat{B}_j$. Each candidate beam $\hat{B}_j^{(b)}$ is formed by extending a beam from the previous step $\hat{B}_{j-1}^{(b)}$ with a newly generated image $I_j^{(k)}$. A complete candidate beam at step j is a sequence of images $\hat{B}_j^{(b)} = [I_1^{(b)}, I_2^{(b)}, \ldots, I_j^{(b)}]$, where each $I_i^{(b)}$ corresponds to the image generated at step i in beam b.

To evaluate the overall quality of a beam, we employ a contrastive classifier [55], denoted as

$$\varphi(I_j \mid \{s_1, \ldots, s_j\}, \{I_1, \ldots, I_{j-1}\}) \tag{4.4}$$

where for each image $I_i^{(b)}$ in the beam, the classifier computes a compatibility score based on the image's prompt s_i and the past text steps and the corresponding generated images $B_{i-1}^{(b)} = [I_1^{(b)}, I_2^{(b)}, \ldots, I_j^{(b)}]$. The total beam score aggregates these contextual evaluations using log-softmax normalization:

$$\text{score}(\hat{B}_j^{(b)}) = \sum_{i=1}^{j} \log \left(\frac{\exp\left(\varphi(I_i^{(b)}, s_i, B_{i-1}^{(b)})\right)}{\sum_{I_i' \in \mathcal{I}_i} \exp\left(\varphi(I_i', s_i, B_{i-1}^{(b)})\right)} \right), \tag{4.5}$$

where $\mathcal{I}_i$ denotes the set of candidate images at step i. This contrastive formulation prioritizes candidates that are both semantically relevant and visually consistent with prior context, helping the model prefer coherent trajectories over locally optimal but globally inconsistent ones.

Once scores are computed, we prune the candidate beams by selecting the top w scoring sequences:

$$B_j = \underset{\substack{B_j \subseteq \hat{B}_j \\ |B_j| = w}}{\text{argmax}} \left(\sum_{i=1}^{w} \text{score}(\hat{B}_j^{(i)}) \right), \tag{4.6}$$

where B_j denotes the final set of retained beams at step j, and w is a fixed beam width.

To balance exploration with computational feasibility, pruning is disabled during the first few steps of the sequence. This allows the model to retain a wide range of initial hypotheses and establish diverse narrative anchors. After these early steps, pruning is gradually introduced to constrain the beam set, focusing computation on the most promising paths. This scheduling strategy ensures a healthy balance between early-stage

diversity and late-stage consistency, leading to high-quality image sequences that remain coherent over long horizons.

4.3 Model Training

BeamDiffusion is a train-free algorithm, except for its hyperparameters beam width (w in Eq. 4.6) and steps back (m in Eq. 4.2) and for its pruning classifier. The contrastive classifier $\varphi(I_j \mid \{s_1, \ldots, s_j\}, \{I_1, \ldots, I_{j-1}\})$ is responsible for guiding BeamDiffusion in scoring each beam of images and steps, section 4.2.2. It is used to evaluate how well a candidate image I_j fits into the evolving visual-semantic context. This classifier plays a critical role in scoring and ranking the multiple candidate continuations generated at each step, enabling the model to retain beams that remain contextually relevant and discard those that drift semantically or visually.

At each generation step j, BeamDiffusion maintains a set of beams, each representing a partial sequence of images up to step $j-1$, denoted as $B_{j-1}^{(b)} = \{(s_1, I_1), \ldots, (s_{j-1}, I_{j-1})\}$. For each beam b, a new set of candidate images $\mathcal{I}_j = \{I_j^{(1)}, \ldots, I_j^{(N \cdot m)}\}$ is generated conditioned on the next prompt s_j. The classifier φ evaluates each candidate $I_j^{(k)}$ with respect to both the current prompt s_j and the prior context $B_{j-1}^{(b)}$, assigning a compatibility score that reflects the overall sequence-level coherence.

Architecture. The classifier builds upon the CLIP architecture to extract joint representations of text and images, but goes beyond direct use of raw CLIP embeddings. Specifically, two separate learned linear projections are applied: one for the context embeddings (representing the previous image-text pairs in the beam), and another for the current candidate image and prompt. These projections map CLIP outputs into a task-specific embedding space optimized for sequence-level alignment.

Given a beam $B_{j-1}^{(b)} = \{(s_1, I_1), \ldots, (s_{j-1}, I_{j-1})\}$, the classifier encodes each context pair $(s_k, I_k) \in B_{j-1}^{(b)}$ for $k = 1, \ldots, j-1$ as:

$$\mathbf{c}_k = \text{Proj}_{\text{ctx}}([\text{CLIP}(s_k); \text{CLIP}(I_k)]), \tag{4.7}$$

and similarly encodes the current candidate (s_j, I_j) as:

$$\mathbf{c}_j = \text{Proj}_{\text{cand}}([\text{CLIP}(s_j); \text{CLIP}(I_j)]). \tag{4.8}$$

Here, $[\cdot;\cdot]$ denotes vector concatenation, and Proj_{ctx}, $\text{Proj}_{\text{cand}}$ are learned linear layers that transform the embeddings into a shared space suitable for scoring.

Scoring Function. The compatibility score is computed by aggregating the similarity between the current candidate and all previous steps in the beam:

$$\varphi(I_j, s_j, B_{j-1}^{(b)}) = \sum_{k=1}^{j-1} \langle \mathbf{c}_j, \mathbf{c}_k \rangle, \tag{4.9}$$

where $\langle \cdot, \cdot \rangle$ denotes dot-product similarity. A higher score indicates stronger alignment of the candidate with the preceding visual-semantic context.

Training Objective. We train the classifier using supervised binary labels $l_{d,j} \in \{0, 1\}$ over candidate sets, employing the Recipes dataset which provides rich sequential image-text pairs of cooking steps. This dataset enables the model to learn meaningful context transitions in a structured procedural domain. The cross-entropy loss is computed across sequences $d = 1, \ldots, D$ and steps $j = 1, \ldots, L$:

$$\operatorname{argmin} \sum_{d=1}^{D} \sum_{j=1}^{L} l_{d,j} \cdot \log \sigma(\varphi(I_j, s_j, B_{j-1}^{(b)})), \tag{4.10}$$

where σ denotes the softmax normalization over competing candidates. During training, ground-truth histories are used for conditioning, ensuring stability and accuracy.

Intuition and Role in BeamDiffusion. The contrastive classifier enables BeamDiffusion to make informed decisions when selecting which beams to retain at each step. Rather than relying solely on local image quality or unconditional generative likelihoods, the classifier assesses how well a candidate image fits within the broader story or visual progression defined by previous steps. It captures both high-level semantic alignment (e.g., narrative coherence) and fine-grained visual consistency (e.g., character identity, scene layout).

By scoring each beam's continuation in a context-aware manner, the classifier allows BeamDiffusion to perform effective pruning of incoherent or implausible trajectories, while still maintaining diverse hypotheses across beams. As a result, the final sequences generated are more contextually faithful and semantically consistent across time.

4.4 Minimizing Decoding Risk

The generation process using diffusion models begins with a seed that determines the initial conditions of the model. The choice of this seed significantly affects the outcome, as it influences the path the model takes during generation. Using multiple random seeds in the first step ensures a diverse exploration of possible outputs, avoiding premature narrowing of options into a single path based on that seed. This limits the diversity of the results and increases the chance of missing better options.

Greedy Decoding and Nucleus Sampling. In both greedy decoding and nucleus sampling, we use the same approach in the first step. Multiple random seeds are employed to generate a set of candidate images. Each candidate is then evaluated using the CLIP score mechanism, which compares the generated images with the step text. The image that best aligns with the text, according to the CLIP score, is selected to continue the process. This method ensures that both greedy decoding and nucleus sampling benefit from multiple diverse starting points, increasing the quality and variety of the generated outputs.

BeamDiffusion. In BeamDiffusion, relying on a single random seed in the first step effectively makes the process a greedy selection, as it locks the generation into a specific path. This limits diversity and reduces the chances of discovering higher-quality outputs. By using multiple random seeds, each seed generates a different decoding trajectory, helping to broaden the search space and increasing the likelihood of finding better solutions. Using multiple random seeds in the initial step enhances the exploration of the latent space, ensuring better and more diverse results across different decoding strategies.

Random Seeds for mid-sequence Generation. The sequence of scenes $S = \{s_1, s_2, \ldots, s_L\}$ may have one scene in the middle of the sequence that should be independent. To address this, for steps $s_{j>1}$, we use a combination of random seeds and past latent vectors to maintain a balance between diversity and coherence, which is particularly important for such scenarios.

4.5 Contextual Scene Descriptions

In sequential visual generation tasks, the textual descriptions associated with individual steps, referred to as *step descriptions* s_j, often fall short of providing the specificity required for accurate and detailed image synthesis. These descriptions may be under-specified, ambiguous, or overly compressed, relying on information that is only understandable in the broader context of the surrounding sequence.

For example, consider a step such as *"Put the ice cream in the freezer for 1 hour"* This instruction implicitly assumes that the visual system already knows what the ice cream and the freezer look like, information that might have been introduced in earlier steps. Similarly, a compound instruction like *"Slice the tomatoes, grate the cheese, and stir the sauce"* compresses multiple temporally and visually distinct actions into a single sentence, making it difficult to render as a coherent and focused image prompt.

To address these challenges, a more sophisticated textual grounding mechanism is needed, one that augments step descriptions with missing visual details and disambiguate references by leveraging prior context. To this end, we employ the Gemini 1.5 Flash model [64], a large multimodal foundation model capable of rich language understanding and generation. Gemini is used to refine each step description into a more detailed and visually grounded caption, which we refer to as a *contextualized scene description*.

Formally, for each step j in a sequence, we construct the contextualized caption c_j by conditioning on the current step description s_j and all preceding steps $\{s_1, s_2, \ldots, s_{j-1}\}$:

$$c_j = \phi(s_j \mid \{s_1, s_2, \ldots, s_{j-1}\}), \tag{4.11}$$

where $\phi(\cdot)$ denotes the Gemini-based contextual captioning function.

This refinement process serves two primary purposes:

- **Context Integration:** By incorporating earlier steps into the generation of c_j, the model can resolve ambiguous references (e.g., "the bowl", "the mixture") and carry

over relevant visual attributes (e.g., the size, color, or identity of objects introduced earlier).

- **Visual Enrichment:** The generated caption is not simply a rephrasing of s_j, but a visually informed reinterpretation. It includes specific scene level details, such as object properties, spatial configurations, and action decompositions—that improve the ability of an image generation model to produce consistent and plausible visual outputs.

In practice, the use of Gemini as a prompt expansion module significantly improves the semantic richness and continuity of generated image sequences. This is particularly beneficial in domains where visual coherence and step-to-step consistency are critical, such as instructional or procedural images sequences.

By converting sparse or compound instructions into well-grounded and visually rich scene descriptions, this approach enhances the alignment between text and image at every stage in the sequence. The contextual captions c_j are then used as direct inputs to the image generation pipeline, serving as high-fidelity visual prompts that reflect both the local instruction and its global context.

4.6 Implementation Details

Base Model. Our method is implemented on top of Stable Diffusion v2.1 [4], a latent diffusion model for high-quality text-to-image synthesis. We build upon the open-source implementation provided by Stability AI, extending it to support beam-based generation, latent trajectory tracking, and contrastive scoring mechanisms as described in the main text.

Latent Trajectory Reuse. In our approach, maintaining coherence across sequential generations requires access to the intermediate latent representations produced during the generative process. Fortunately, the model pipeline inherently provides the full sequence of latent representations at each generation step. This enables us to easily extract and reuse specific latents as seeds for new generations.

Contrastive Classifier Training. The contrastive classifier described in Section 4.3 is trained on the VIST dataset [65] and the Recipes dataset [48], which are described in detail in Section 5.1. Training is conducted using a batch size of 500, a learning rate of 0.01, and spans 50 epochs. We employ a staged fine-tuning strategy: the CLIP encoder remains frozen for the first 10 epochs and is then unfrozen for the remainder of training to enable task-specific adaptation.

Hardware. All training and evaluation experiments were conducted on NVIDIA A100 GPUs with 40GB of memory.

4.7 Summary

BeamDiffusion presents a novel approach to the challenging task of generating visually and semantically coherent image sequences from textual narratives. Unlike conventional Latent Diffusion Models (LDMs) that treat each generation step independently, BeamDiffusion explicitly models inter-step dependencies by navigating the latent denoising space using a beam search-inspired strategy.

The core innovation lies in the introduction of *diffusion beams*, a structured mechanism for maintaining multiple latent trajectories over time. By leveraging latent representations from prior steps, BeamDiffusion conditions the generation of future images on contextual information, enabling greater temporal and narrative consistency. The model explores a combinatorial space of latent paths, guided by a contrastive scoring function that ranks candidates based on their alignment with both the current textual prompt and accumulated visual context.

This framework incorporates several key contributions:

- A continuous-space beam search formulation that operates in the latent diffusion space, allowing systematic exploration of plausible generative paths.
- A latent reuse mechanism that conditions generation on stored denoising trajectories from earlier steps, promoting coherence in character identity, object layout, and scene composition.
- A contrastive scoring model that evaluates image candidates based on their semantic fit and visual consistency, enabling effective pruning of suboptimal paths.
- A flexible generation pipeline that balances early-stage diversity with late-stage consistency through scheduled pruning, resulting in high-quality, coherent image sequences.

By integrating structured search with latent reuse and learned scoring, BeamDiffusion bridges the gap between isolated image generation and sequence-level coherence. It sets a new foundation for narrative-consistent image synthesis, particularly in domains requiring temporal grounding, character continuity, and visual storytelling.

5

Experimental Setup and Methods

In this chapter, we describe the experimental setup to evaluate BeamDiffusion's performance in generating coherent image sequences. We outline the datasets used, identify baseline methods, and explain the human, automatic, and LLM-as-a-judge (Gemini [64]) evaluations focused on semantic and visual consistency across generated sequences.

5.1 Datasets

We constructed our dataset using publicly available resources from the Recipes and DIY domains [48], as well as the Visual Storytelling (VIST) dataset [65]. The Recipes dataset serves as our in-domain benchmark, while DIY tasks and VIST provide out-of-domain examples, allowing us to assess the generalizability of our model across different types of procedural and visual content.

The Recipes and DIY datasets consist of manual tasks that include a title, description, list of ingredients or tools, and a sequence of step-by-step instructions. These steps may be illustrated with images, particularly in the Recipes dataset, which contains approximately 1,400 tasks, with an average of 5.06 steps per task, and a total of 6,860 steps. In contrast, the DIY dataset covers a broader range of project complexity, from simple crafts to more intricate builds, all described through detailed instructional steps.

To further test our model's ability to generalize beyond structured, instruction-based data, we incorporated the Visual Storytelling (VIST) dataset. We specifically used the Descriptions of Images-in-Isolation (DII) annotations, which provide focused, standalone textual descriptions of individual images. These DII annotations are particularly useful for generating accurate and informative visual prompts, as they capture key visual elements without relying on surrounding narrative context. To further enhance text-image alignment, we leveraged CLIP to automatically select the annotation that best represents each image. VIST comprises over 50,000 image sequences, with an average of 5 steps per task, offering a diverse and challenging testbed for both visual and textual understanding.

5.2 Baselines

To evaluate the performance of our image sequence generation method, we compare **BeamDiffusion** against Stable Diffusion 2.1 [4] and two competitive structured generation methods: **GILL** [46] and **StackedDiffusion** [66].

- **GILL:** Enables joint text and image generation by inserting image tokens into language sequences. It treats image generation as part of language modeling and leverages a pretrained image decoder to synthesize visuals from token embeddings. Details are provided in Section 3.4.1.

- **StackedDiffusion:** Generates entire image sequences in a single pass by treating them as spatially tiled representations, conditioned on step-wise text inputs. This promotes temporal coherence without autoregressive decoding. Further described in Section 3.4.1.

We also include two other decoding strategies adapted for sequential image generation: **Greedy** and **Nucleus Sampling** [67], both of which are novel in this context. All methods are evaluated with 50 denoising steps.

- **Greedy**: At each step, a new image is generated using the first five latents from the previous image, and the highest-scoring candidate (based on CLIP similarity) is selected. This strategy mirrors the greedy text decoding method described in Section 2.3.1, where only the most likely continuation is selected at each step. While similar to $\text{CoSeD}_{\text{len}=1}$ [55], this variant uses only CLIP scores without a learned scoring module.

- **Nucleus Sampling**: A probabilistic decoding method that samples from the top-p cumulative CLIP-scored candidates instead of deterministically selecting the top image. This encourages diversity while preserving quality, as detailed in Section 2.3.2.

5.3 Human Evaluation

To facilitate the process of human annotation, we developed a custom website tailored specifically for human annotation, as existing tools lacked the flexibility required to evaluate multiple configurations simultaneously. Selection criteria focused on how well the image sequences captured the meaning of the input text (semantic alignment) and maintained coherent visual elements such as backgrounds, objects, and ingredients throughout the sequence (visual consistency).

This human evaluation designed was deployed to determine BeamDiffusion's hyperparameters (Figure 5.1) and to compare different methods (Figure 5.2).

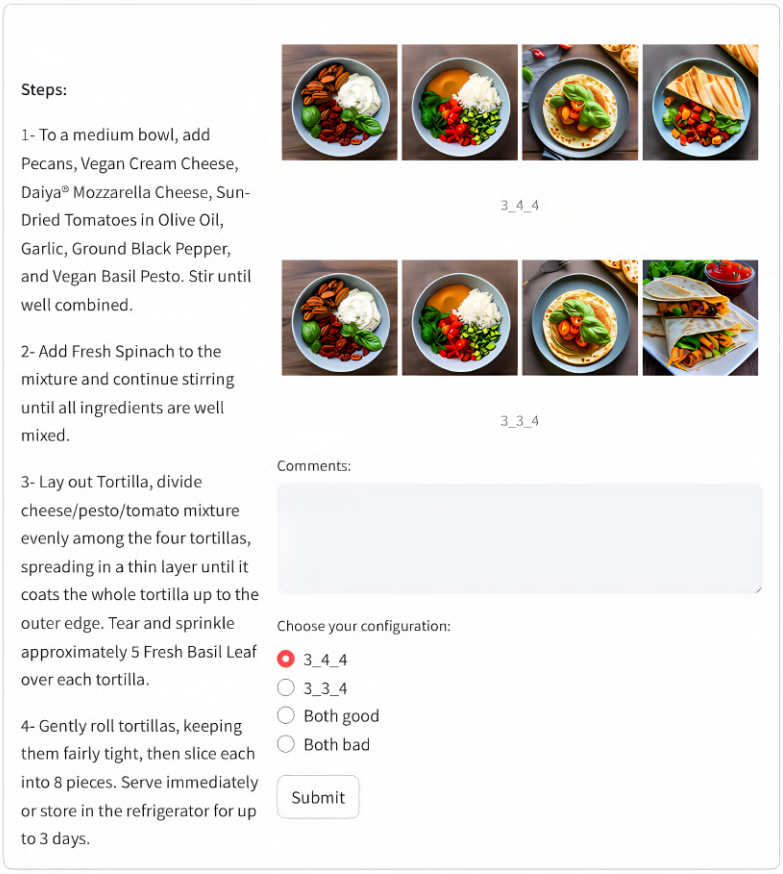


Figure 5.1: Example of the human annotation elimination round for selecting the best beam search configuration.

5.4 Gemini Evaluation

To further validate our findings and evaluate a wider set of tasks, we employed Gemini to assess both beam search configurations and baseline methods. Gemini's evaluation protocol closely mirrored the human annotation process, with one important modification: whereas human annotators could select options such as *"both good"* or *"both bad"* when comparing two sequences, Gemini tended to default to these non-decisive choices when available. To encourage more definitive and accurate evaluations, we removed these options from Gemini's prompt.

Additionally, we refined the comparison methodology for Gemini. Initially, presenting all five sequences simultaneously led to biased results, with the model disproportionately favoring the last two sequences. To address this, we adopted a pairwise evaluation strategy, comparing two sequences at a time. This approach produced more balanced and reliable assessments across comparisons. Figure 5.3 shows the prompt used with Gemini to guide the evaluation of image sequences based on both visual and semantic consistency.

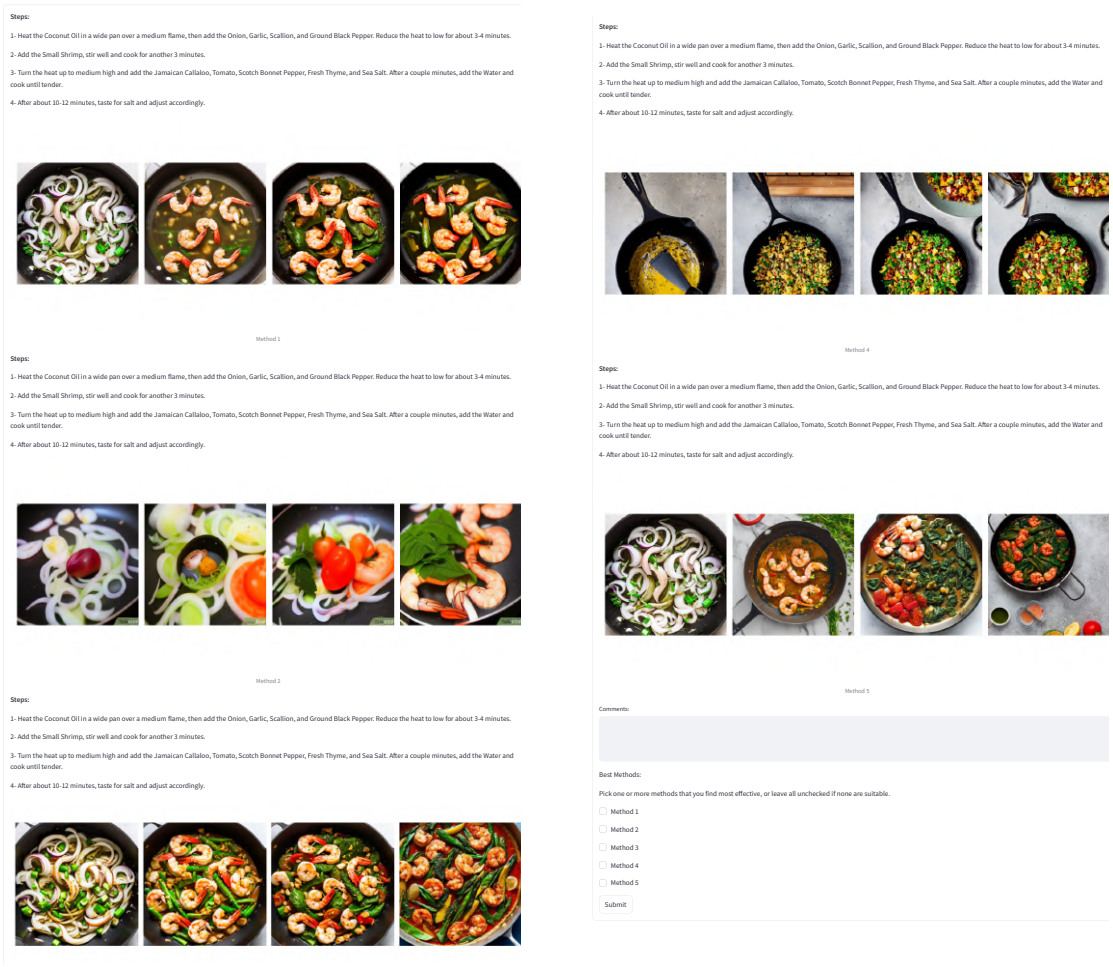


Figure 5.2: Example of a human annotation interface used to select the best method based on generated image sequences.

Gemini Prompt:
Evaluate two image sequences using the following criteria:

- **Visual Consistency**: Objects, ingredients, backgrounds, colors, and styles should remain consistent across frames. Frames should transition smoothly, resembling a video-like progression with no sudden changes or missing elements.
- **Semantic Consistency**: Images must logically follow the described steps, clearly and accurately.

Steps of the process: {steps}
After evaluating both sequences, select the one with the strongest overall consistency (visual and semantic). Pay close attention to even minimal differences, as the sequences are very similar. If both sequences are equally good or bad, choose the one that is slightly better. Choose exactly one of the following options:
- "First sequence"
- "Second sequence"

Figure 5.3: Gemini prompt used to select the best beam search configuration and best method.

5.5 Human-Gemini Agreement

To assess the reliability of using Gemini for automatic evaluation, we first measured the alignment between Gemini's evaluations and human judgments. We collected human ratings for a representative subset of tasks, specifically, 16 tasks from the Recipes dataset, 10 from the DIY dataset, and 10 from the VIST dataset. Each task was independently rated by multiple human annotators. To quantify inter-rater agreement, we computed Fleiss' kappa [68], a statistical measure used to assess the reliability of agreement among multiple raters. The resulting kappa score was 0.70, indicating substantial agreement between human annotators and a high level of consistency in Gemini's assessments relative to human judgment. Given this strong alignment, we proceeded to use Gemini to evaluate the remaining tasks in our dataset. This approach allowed us to scale the evaluation process efficiently while maintaining a high degree of confidence in the quality and reliability of the assessments.

5.6 Automatic Metrics

Following prior work [48, 55, 60], we evaluate methods using two main criteria: (1) *Visual Consistency*, measured via average image similarity using DINO [26] (DINO-I) and CLIP [24] (CLIP-I); and (2) *Semantic Alignment*, measured via CLIP (CLIP-T), which compares each generated image to the averaged step text up to and including the current step. To ensure comparability, CLIP-I, CLIP-T, and DINO-I scores are normalized by their respective maxima. We then compute two composite metrics: CLIP*, defined as the product of CLIP-I and CLIP-T; and DINO*, defined as the product of DINO-I and CLIP-T.

These composite metrics balance visual and semantic quality, with a bias toward semantic alignment, which is crucial for reflecting the sequential text input. However, raw CLIP similarity scores can be misleading. In image sequence generation, CLIP-I values between visually similar frames often exceed 0.85, even when the content is nearly duplicated. Wang et al. [69] note that such high CLIP-I scores can indicate semantic duplication, which undermines meaningful visual progression. Conversely, poor alignment between images and their textual prompts is reflected in low CLIP-T scores. Lin et al. [70] address this by clipping CLIP-T values below 0.1 to filter out weak matches. To address these shortcomings, we apply a clipping strategy:

- Normalized CLIP-I scores above 0.9 and normalized DINO-I scores above 0.85 are set to zero, discouraging near-duplicates.
- Normalized CLIP-T scores below 0.1 are zeroed out, penalizing poor semantic alignment.

This strategy encourages the generation of sequences that are both semantically faithful and visually diverse. As illustrated in Figure 5.4, high CLIP-I does not necessarily imply

correctness, GILL produces a visually similar but semantically incorrect frame, revealing the need for more nuanced evaluation.

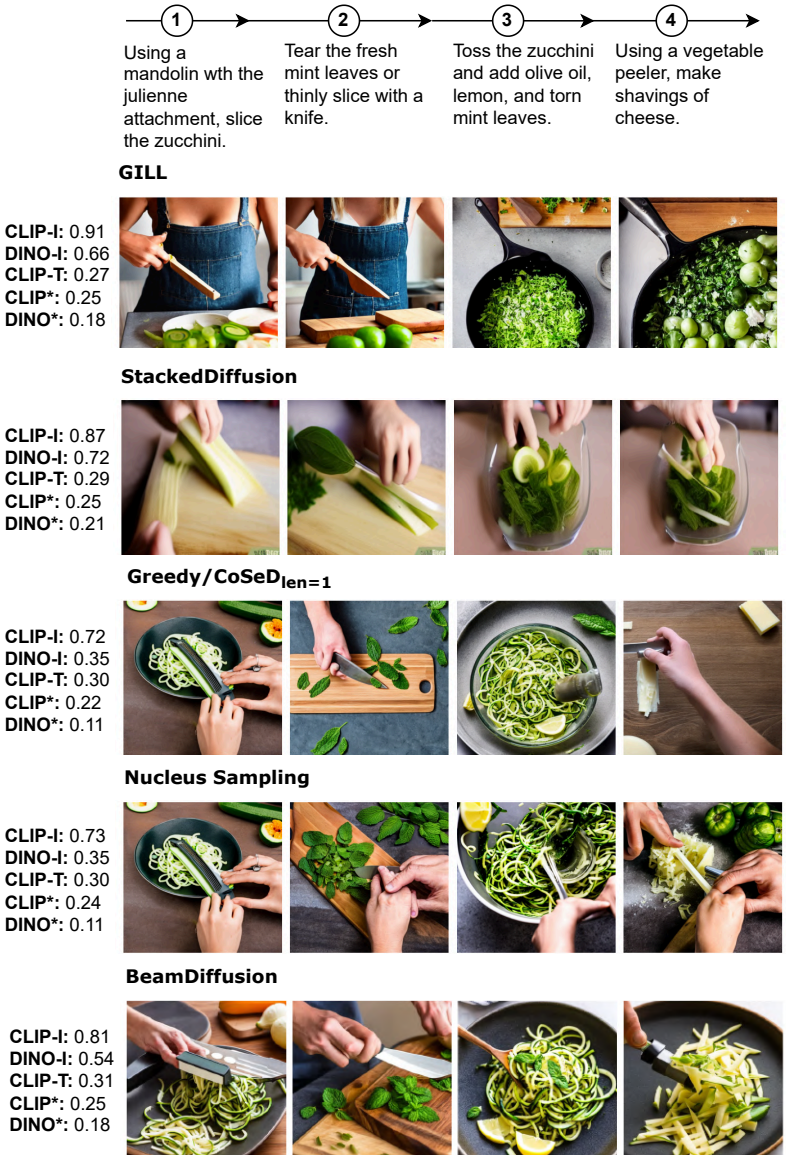


Figure 5.4: Illustration of the limitation of CLIP-I: despite a high CLIP-I score, the result is not semantically correct or visually faithful. The example compares several automatic metrics (CLIP-T, CLIP-I, DINO-I, CLIP*, DINO*), highlighting discrepancies and failure cases in sequence evaluation.

While frame-level consistency and image-text alignment are important, these previous metrics primarily capture local relationships within sequences. To better assess the models' overall instruction-following ability and sequence coherence, we also evaluate them using more global metrics inspired by StackedDiffusion [66].

Goal Faithfulness (GF). This metric evaluates how accurately the *final generated image* represents the overall goal of the instruction (e.g., "How to Bake Turkey Wings"). For each

generated image, we compute the CLIP similarity between the image and its corresponding goal text, and compare it against similarities with three randomly selected, unrelated goal texts. The model is considered correct if the true goal has the highest similarity. The final score is the accuracy over all tasks, with higher scores indicating better goal alignment.

Step Faithfulness (SF). Step Faithfulness measures how well each generated image corresponds to the specific step it is intended to depict. Like Goal Faithfulness, this is evaluated based on CLIP similarity. For each image, we compare its similarity to the correct step text versus other steps from the same goal. The model is rewarded for assigning higher similarity to the true step, capturing whether each image accurately follows the intended instruction at that point in the process.

Cross-Image Consistency (CIC). This metric assesses the visual coherence among images generated for different steps of the same goal. Ideally, all step images in a sequence should depict a consistent visual world, for example, maintaining the same objects, colors, and scene layout across steps. We measure this using two versions of the metric, both based on DINO image embeddings:

- ℓ_2 **distance:** Computes the average pairwise ℓ_2 distance between DINO embeddings of step images for each goal. A lower ℓ_2 distance indicates greater consistency in visual appearance across steps.
- **Cosine similarity:** Computes the average pairwise cosine similarity between DINO embeddings of step images. A higher cosine similarity indicates stronger alignment of visual features across steps.

6

Results and Discussion

In this section, we describe the performance of BeamDiffusion based on both human and automatic evaluations. We begin by analyzing the human evaluation, proceed to discussing the automatic evaluations, then focus on specifics of our approach such as hyperparameters and non-linear decoding characteristics, and finally provide a qualitative analysis.

6.1 General Results

Table 6.1 compares method preferences across Recipes, DIYs, and VIST, as judged by both human annotators and an LLM-based evaluator. Across all domains, **BeamDiffusion achieves the highest selection rates**, demonstrating robust alignment with human judgments while maintaining strong performance at scale when assessed by the LLM.

In the **Recipes** domain, BeamDiffusion is chosen in half of the human evaluations (50.0%) and 37.6% of LLM-as-a-judge evaluations, outperforming all baselines. Nucleus Sampling emerges as the closest competitor (30.0% human, 23.0% LLM), while Greedy decoding is notably preferred by the LLM judge (27.6%) but is never selected by humans. This discrepancy suggests that the LLM judge may prioritize surface-level fluency or step-wise alignment, whereas human evaluators focus more on global coherence and visual consistency.

The **DIY** domain presents the most competitive results. BeamDiffusion is again preferred by humans (50.0%) and the LLM judge (30.0%), but Nucleus Sampling is unusually strong here, reaching 37.5% in human evaluations and 27.7% in the LLM judge evaluation. Unlike Recipes, humans occasionally favor sampling-based outputs in this less structured domain, which may reflect the higher variability and open-ended nature of DIY sequences. Still, BeamDiffusion retains a clear edge.

In the **VIST** domain, BeamDiffusion achieves its strongest human preference, being selected in 58.3% of cases. It also receives 34.9% of LLM-judge selections, maintaining a clear lead over Nucleus Sampling (38.3% human, 26.6% LLM). This indicates that BeamDiffusion is particularly effective in narrative-driven, visually grounded storytelling

Table 6.1: Frequency of method selection by human annotators (H) and Gemini (G).

	Recipes		DIY		VIST	
Method	**H (%)**	**G (%)**	**H (%)**	**G (%)**	**H (%)**	**G (%)**
GILL [46]	0.0	2.0	0.0	20.0	3.3	16.1
StackedDiffusion [66]	20.0	9.8	12.5	15.6	0.0	3.2
	Image Sequence Decoding					
Greedy / $\text{CoSeD}_{\text{len=1}}$ [55]	0.0	27.6	0.0	23.3	0.0	26.3
Nucleus Sampling [67]	30.0	23.0	37.5	27.7	38.3	26.6
BeamDiffusion	**50.0**	**37.6**	**50.0**	**30.0**	**58.3**	**34.9**

Table 6.2: Comparison of CLIP-I, DINO-I and CLIP-T across different methods with the standard deviation

Method	CLIP-I	DINO-I	CLIP-T	CLIP*	DINO*
GILL [46]	0.55 ± 0.30	0.49 ± 0.26	0.43 ± 0.08	0.23	0.21
StackedDiffusion [66]	0.76 ± 0.12	0.46 ± 0.14	0.47 ± 0.09	0.36	0.22
	Image Sequence Decoding				
Greedy / $\text{CoSeD}_{\text{len=1}}$ [55]	0.81 ± 0.13	0.54 ± 0.14	**0.48 ± 0.07**	0.39	0.26
Nucleus Sampling [67]	0.77 ± 0.15	0.53 ± 0.15	**0.48 ± 0.08**	0.37	0.25
BeamDiffusion	**0.83 ± 0.14**	**0.62 ± 0.18**	**0.48 ± 0.09**	**0.40**	**0.30**

tasks.

Among the baselines, **GILL** consistently performs the worst, almost never selected by humans and only modestly by the LLM judge. **StackedDiffusion** improves over GILL but remains well behind BeamDiffusion and decoding-based methods. By contrast, **Nucleus Sampling** is the most competitive baseline, particularly in DIY, though it never surpasses BeamDiffusion. The **Greedy** approach stands out for its evaluator discrepancy: while humans rarely favor it, the LLM judge assigns it relatively high preference in Recipes and VIST, reinforcing the notion that, despite the overall good alignment, automatic judgments may emphasize different qualities than human perception.

Finally, it is important to note that the **LLM judge evaluated far more sequences than humans**, producing smoother, more statistically stable distributions of preferences. Human annotations, although noisier due to smaller sample sizes, offer a more direct measure of perceptual quality. The consistency of BeamDiffusion's superiority across both evaluators, despite these methodological differences, strengthens confidence in its effectiveness.

6.2 Automatic Metrics

To assess the semantic and visual quality of generated image sequences, we conduct automatic evaluations using CLIP and DINO based metrics, explained in Section 5.6. As

Table 6.3: Win percentages across datasets for each method and metric.

Method	CLIP-I	DINO-I	CLIP-T	CLIP*	DINO*
GILL [46]	**35.00**	26.00	7.00	13.50	18.00
StackedDiffusion [66]	14.50	23.00	14.00	14.50	13.50
Image Sequence Decoding					
Greedy / $\text{CoSeD}_{\text{len=1}}$ [55]	18.50	13.50	13.50	12.50	13.50
Nucleus Sampling [67]	5.00	7.00	**42.50**	14.00	11.50
BeamDiffusion	27.00	**30.50**	23.00	**45.50**	**43.50**

summarized in Table 6.2, BeamDiffusion achieves the highest overall mean scores on both CLIP* (0.40) and DINO* (0.30), indicating a strong balance between semantic alignment and visual consistency.

In terms of individual metrics, BeamDiffusion also obtains the best scores for image consistency, with CLIP-I at 0.83 and DINO-I at 0.62. These metrics assess the similarity between consecutive images, indicating that BeamDiffusion produces the most visually coherent sequences. For image-text alignment, as measured by CLIP-T, BeamDiffusion attains the highest score of 0.48, though this peak is shared with Greedy and Nucleus methods. This overlap is expected since Greedy and Nucleus explicitly use CLIP-T to guide output selection based on textual alignment.

Among the baseline models, StackedDiffusion consistently outperforms GILL across almost all metrics but remains behind BeamDiffusion and other decoding-based approaches. GILL exhibits the weakest performance in both semantic and visual metrics, with notably low CLIP* (0.23) and DINO* (0.21) scores.

Complementing the mean scores, the table also presents the standard deviation (σ) of these metrics to assess stability and consistency. A lower σ indicates more stable performance across diverse sequences, while a higher σ suggests variability in quality or responsiveness to task complexity. GILL has the highest variability in CLIP-I and DINO-I, reflecting inconsistent visual coherence and fluctuating similarity between consecutive images. In contrast, StackedDiffusion displays low variability, suggesting reliable and consistent behavior across tasks. BeamDiffusion achieves a favorable balance, coupling strong mean scores with moderate variability, which highlights its robustness and adaptability.

It is also worth noting that CLIP-T scores generally exhibit lower standard deviations across models, likely because CLIP-T is directly optimized during diffusion model training. As a result, CLIP-T tends to be more stable across different tasks and methods.

Overall, the combined analysis of average and variance metrics confirms that BeamDiffusion offers the most robust performance for sequential image generation. It excels in maintaining high semantic fidelity, prompt alignment, and visual consistency while avoiding redundancy, validating the effectiveness of the beam-based decoding strategy.

Win Rate Analysis and Metric Misalignment. Across CLIP and DINO based metrics, we further analyze win percentages, the proportion of sequences in which each method achieves the highest score for a given metric. Table 6.3 presents these win rates, revealing important insights into how different methods balance visual consistency and textual alignment, and how this trade-off impacts their overall performance.

GILL achieves the highest win rate in CLIP-I and a strong showing in DINO-I, indicating that it often generates sequences with visually coherent frames. However, its very low win rate in CLIP-T highlights poor alignment with the step-level text instructions. This imbalance results in middling combined metric win rates, reflecting GILL's failure to balance both visual consistency and semantic alignment effectively. Despite producing visually consistent images, GILL struggles to follow the textual narrative properly, which limits its overall performance.

StackedDiffusion shows more balanced but moderate performance across the board. Its win rates in CLIP-I, DINO-I, and CLIP-T are all modest, and its combined metric win rates remain low. This suggests that while StackedDiffusion manages some balance between visual and textual aspects, it does not excel strongly in either, further supporting the idea that successful sequential image generation requires careful equilibrium between maintaining frame-to-frame consistency and accurately representing textual instructions.

Greedy and Nucleus Sampling display contrasting patterns. Greedy sampling has relatively higher visual consistency wins but only moderate CLIP-T wins, which limits its combined metric success. Nucleus sampling, in contrast, achieves the highest win rate in CLIP-T by far, showing strong alignment to textual instructions. However, it scores very low in visual consistency, which drags down its combined metric performance. This highlights the challenge of maintaining visual quality without sacrificing textual fidelity.

BeamDiffusion achieves the best balance between these competing demands. It attains high win rates in both visual consistency metrics and maintains a respectable win rate in CLIP-T. This balanced performance leads to dominant combined metric wins, confirming that a method must optimize across both image consistency and text alignment to generate high-quality sequential outputs.

When we compare these automatic win rates with human and Gemini preferences (Table 6.1), notable discrepancies emerge. GILL and StackedDiffusion, despite their moderate win rates in CLIP* and DINO*, receive almost no preference from human annotators and low scores from Gemini. This suggests that CLIP* and DINO* metrics may overvalue superficial image-text or intra-sequence similarities while missing the broader narrative coherence and procedural correctness that humans and Gemini prioritize.

Conversely, Nucleus Sampling, which scores poorly in CLIP* and DINO* metrics, is often preferred by humans and Gemini. This indicates that human evaluators appreciate qualities such as diversity and contextual richness that automatic metrics sometimes penalize.

BeamDiffusion uniquely demonstrates strong agreement between automatic metrics and human judgments. It achieves the highest win rates in CLIP* and DINO* metrics

Table 6.4: Comparison of different methods across Goal Faithfulness (GF), Step Faithfulness (SF), and Cross-Image Consistency metrics (CIC).

Method	GF ↑	SF ↑	CIC ↓ (ℓ_2 distance)	CIC ↑ (Cosine similarity)
GILL [46]	0.5229	0.5505	1.5803	0.5854
StackedDiffusion [66]	0.6330	0.6697	1.4228	0.6203
Image Sequence Decoding				
Greedy / $\text{CoSeD}_{\text{len=1}}$ [55]	0.7523	0.8624	1.4224	0.6535
Nucleus Sampling [67]	0.7156	0.8624	**1.4041**	0.6459
BeamDiffusion	**0.8165**	**0.8991**	1.4278	**0.6643**

and is consistently preferred by humans and Gemini across all domains. This alignment suggests that while existing automatic metrics have limitations, they can provide valuable insights when a method performs robustly across multiple quality dimensions.

Overall, this analysis shows that a balanced performance across CLIP-I, DINO-I (which capture visual consistency), and CLIP-T (which captures textual alignment) is crucial. Methods excelling in one aspect but not the other tend to underperform in combined metrics and are less favored by human evaluators.

Global Semantic and Procedural Evaluation. Table 6.4 reports Goal Faithfulness, Step Faithfulness, and Cross-Image Consistency metrics, which capture the alignment of generated sequences with overall goals, individual steps, and visual coherence across frames.

BeamDiffusion consistently outperforms all other methods in both Goal Faithfulness (0.8165) and Step Faithfulness (0.8991), demonstrating its superior ability to generate images that faithfully follow the intended narrative from start to finish. This highlights its effectiveness at preserving both the global intent and the step-by-step progression of the input instructions. In terms of Cross-Image Consistency, measured by DINO embeddings, BeamDiffusion achieves the highest cosine similarity (0.6643), indicating strong visual coherence throughout the sequence. Although its ℓ_2 distance (1.4278) is slightly higher than Nucleus Sampling's best score (1.4041), this does not necessarily affect BeamDiffusion's overall consistency. This suggests that BeamDiffusion may preserve stable visual features across steps while remaining aligned with the task objectives.

Together, these results complement the earlier CLIP and DINO based metrics by validating that BeamDiffusion not only excels at frame-level semantic alignment but also ensures procedural correctness and smooth temporal transitions, confirming its robustness in sequential image generation.

The Need for Better Automatic Metrics. Importantly, the challenge of finding reliable automatic metrics that align well with human judgment, especially for complex tasks

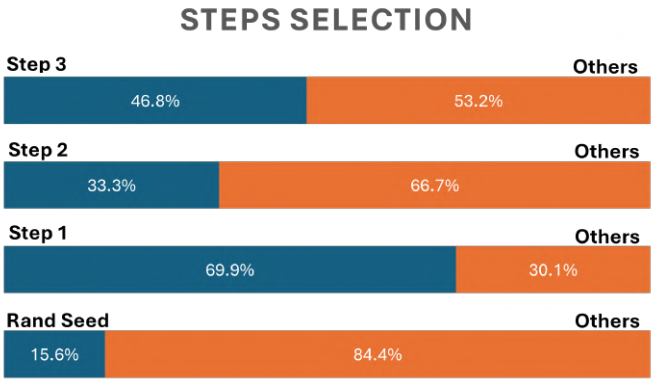


Figure 6.1: Non-linear steps selection, according to selection opportunities.

like sequential visual generationis a longstanding problem in computer vision. Although CLIP* and DINO* are tailored to capture sequence-level qualities, they still fail to fully assess temporal coherence, procedural correctness, and narrative flow dimensions better recognized by humans and multimodal models like Gemini. Developing more comprehensive, human-aligned evaluation metrics remains a critical and ongoing research frontier for advancing sequential visual generation.

6.3 Non-Linear Sequence Denoising

To analyze how BeamDiffusion explores the latent space, we measured how often a sequence step contributes latents to the generation of subsequent steps in sequences of four steps (Figure 6.1). A more detailed analysis at the level of denoising iterations is provided in Section 6.6.2.

Focusing on the four-step sequences, we observed a strong preference for early steps. Latents from Step 1 were selected 69.9% of the times they were eligible, highlighting the central role of the first step in shaping the sequence. Selection rates declined for later steps, with Step 3 chosen in 46.8% of opportunities and Step 2 in only 33.3%.

These percentages are computed relative to the number of opportunities each step had to be selected, rather than as a simple additive total. Since earlier steps naturally have more chances of being reused, their higher rates reflect both their greater availability and their stronger influence on subsequent steps. This effect is also consistent with the fact that most of the core elements of the sequence are introduced in the first step. Beam search therefore prioritizes preserving the latents from Step 1 to maintain coherence, while later steps contribute more selectively refining or extending what was already established rather than redefining it.

Overall, these findings suggest that beam search anchors the generation process around the earliest steps, particularly Step 1, while strategically incorporating intermediate steps to balance stability with flexibility in the evolving sequence.

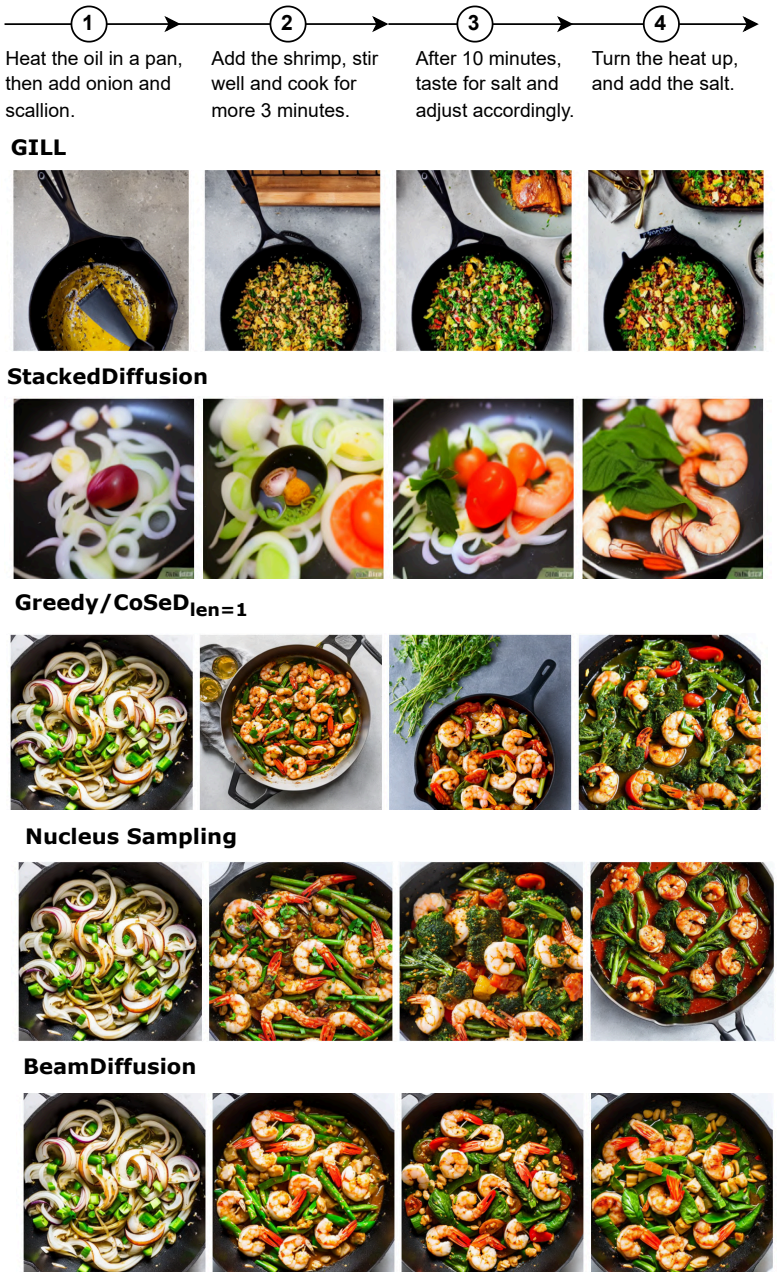


Figure 6.2: Example of the Recipes domain task *"Jamaican Callaloo With Shrimp"*, with a sequence of 4 steps

6.4 Qualitative Analysis

To highlight how different decoding methods and models impact the semantic and visual consistency of generated outputs, we present several examples in Figures 6.2–6.4 and Appendix A.2. We compare three decoding methods (Greedy, Nucleus, and Beam) with two models (GILL and StackedDiffusion).

In Figure 6.2, BeamDiffusion consistently maintains the same pan across all steps, a feature not always seen in Greedy or Nucleus methods. While GILL struggles to follow the step text and results in more inconsistencies, BeamDiffusion performs better in preserving

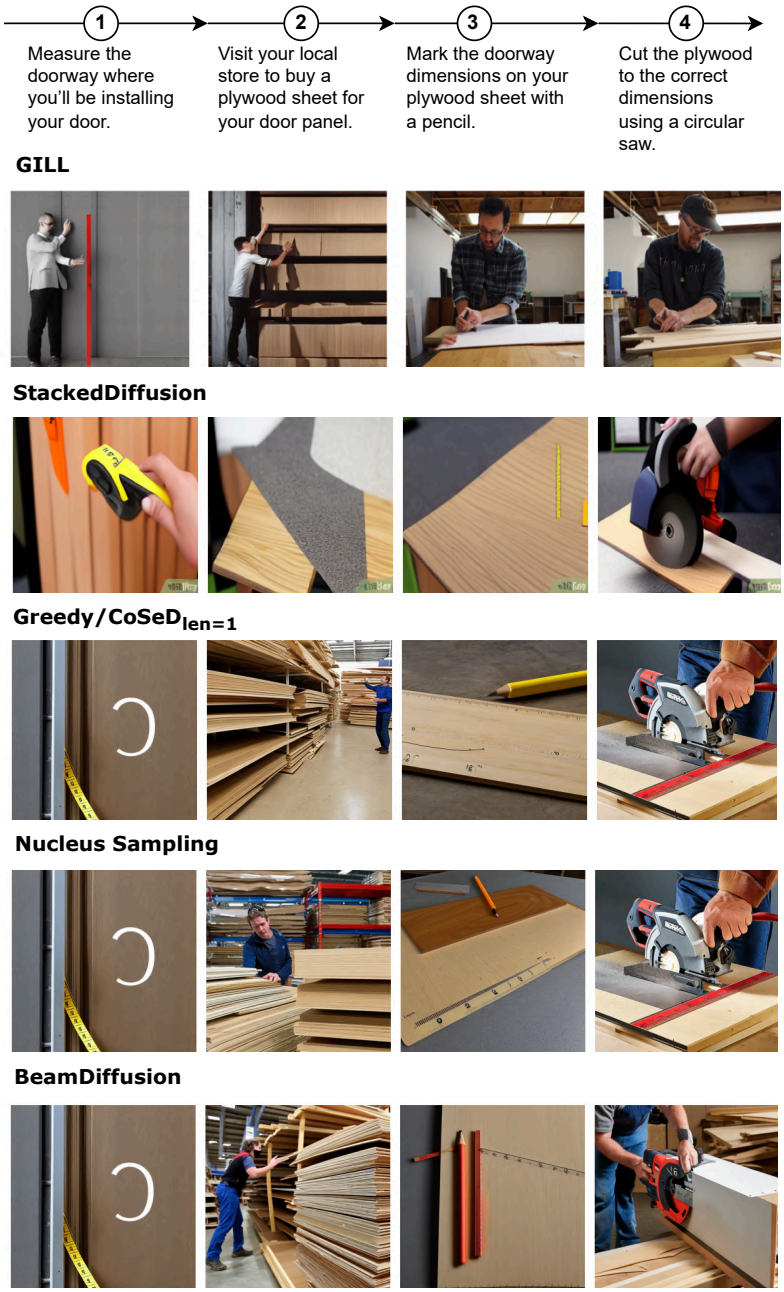


Figure 6.3: Example of the out-of-domain DIY task *"How to Make a Door"*, with a sequence of 4 steps.

visual coherence compared to Greedy and Nucleus, though it still faces some challenges. For DIY tasks, as shown in Figure 6.3, BeamDiffusion can preserve basic elements such as the color of the plywood and a basic adherence to the text. However, overall, all methods face challenges in capturing precise details. In the VIST dataset, Figure 6.4, Greedy and Nucleus alternate between black-and-white and color images, while BeamDiffusion ensures color consistency throughout the sequence. Furthermore, background elements, such as bushes, are maintained in the first three steps, and the individuals in the first and third steps closely resemble each other. Overall, BeamDiffusion excels in maintaining

Figure 6.4: Example of the out-of-domain VIST story *"July 4th Barbecue"*, with a sequence of 4 steps.

both semantic and visual consistency, even in complex out-of-domain tasks, In contrast, methods such as StackedDiffusion and GILL encounter challenges in keeping a balance between text alignment and consistency across images.

6.5 BeamDiffusion Hyperparameters

The hyperparameters *beam width* (w in Eq.4.6) and *steps back* (m in Eq.4.2) play a critical role in the performance of BeamDiffusion. This section reports an exhaustive evaluation

of how varying these values, beam widths $2, 3, 4, 6$ and steps back $2, 3, 4$, affects latent selection and the overall quality of the generated sequences. A more detailed analysis of latent selection behavior is provided in Section 6.6.

To determine the optimal hyperparameters, we conducted a human evaluation in which annotators compared pairs of beam search configurations. Five annotators were presented with 24 unique sequences, each corresponding to a distinct configuration generated by combining three key factors:

- **Steps Back** (m in Eq.4.2): Number of previous latent representations used. We tested values of 2, 3, and 4.
- **Latents Indexes**: Two sets were evaluated, 1, 3, 5, 7 and 0, 1, 2, 3, to explore different levels of latent details.
- **Beam Width** (w in Eq.4.6): Number of beams retained at each step, with values 2, 3, 4, and 6.

Evaluating all 24 configurations at once was cognitively demanding. To simplify the task, we used an elimination-style annotation method. Annotators compared two configurations at a time and chose the one that produced better semantic and visual coherence. If both configurations were equally good or equally bad, they could mark them as *both good* or *both bad*. In those cases, the computationally cheaper option advanced (see Figure 5.1). This process was repeated until no further improvements were possible, yielding the most effective configuration.

We first conducted a human evaluation on a diverse set of generated sequences (see Section 5.3). Poorly performing configurations, including all variants using the latent index set $\{1, 3, 5, 7\}$, were eliminated early. Table 6.5 shows that the best-performing setup was *2 steps back* with a *beam width of 4*, which was selected in 25% of comparisons. Two other configurations, *3 steps back, beam width 2* and *4 steps back, beam width 2*, also performed well, each selected in 16.67% of cases. By contrast, configurations using a beam width of 6 consistently underperformed across all settings.

To test the reliability of these findings, we first measured the agreement between Gemini and human annotators on a set of six tasks. The resulting 85% Fleiss' Kappa [68] indicated near-perfect consistency. Having established this level of agreement, we then used Gemini to evaluate the remaining tasks. The results (Table 6.6) confirmed that *2 steps back, beam width 4* was still the strongest configuration, now selected in 48.57% of comparisons, nearly double its initial performance. The *3 steps back, beam width 2* configuration also remained competitive, with a 28.57% selection rate. Configurations with larger beam widths, such as 6, again underperformed, showing that simply increasing beam width does not guarantee better results.

Table 6.5: Influence of different hyperparameter values based on human annotations. Values are percentage of times each configuration was selected as best.

Steps Back	Beam Width			
	2	**3**	**4**	**6**
2	8.33	8.33	**25.00**	8.33
3	16.67	-	-	-
4	16.67	8.33	8.33	-

Table 6.6: Influence of different hyperparameter values based on Gemini annotations. Values are percentage of times each configuration was selected as best.

Steps Back	Beam Width			
	2	**3**	**4**	**6**
2	5.71	2.86	**48.57**	5.71
3	28.57	-	2.86	-
4	-	-	2.86	-

6.6 Impact of Latent Selection on Beam Search Performance

Latent selection plays an important role in shaping the quality and coherence of image sequences generated through beam search. Examining how different latent groups contribute to generation performance can provide insights into their effect on semantic alignment and visual consistency. In particular, the choice of latent affects the steerability of the diffusion process: during early denoising iterations, the selected latent can still be guided by the prompt of the next step, while in later iterations the process becomes more constrained as it converges toward a particular solution.

This aligns with recent work on interpreting diffusion models. For example, [71] ground textual descriptions in image regions, while [72] and [73] explore how textual prompts influence generation through attention analysis and partial information decomposition. In contrast, our method contributes to interpretability by examining how latent space traversal, guided by beam search, affects the generation of coherent, semantically aligned image sequences.

To systematically investigate the effects of latent selection, we evaluate several sequences, each consisting of four steps. This allows us to study how different latent choices influence the quality and coherence of generated image sequences, as well as how information is preserved and manipulated throughout the denoising process. Using a combination of automatic evaluation metrics and human annotations, we focus on two complementary aspects:

- Latent groups: analyze different latent configurations to determine which groups contribute most to generating coherent and semantically meaningful sequences.

- Early vs late denoising iterations: in the second experiment we analyzed how early or late in the denoising process latents preserve the information they carry and their steerability, that is, the extent to which a latent can be guided or modified by a new prompt, allowing them to be reused in subsequent denoising steps.

6.6.1 Impact of Latent Groups on Performance

Annotations. Following the initial analysis of beam search configurations (Sec. 6.5), where we used latent indexes 0, 1, 2, and 3, we further investigated the impact of using different latent configurations on the generated sequences. While latents 0, 1, 2, and 3 show promising results, latents 1, 3, 5, and 7 exhibit significantly poorer performance. In particular, beam search configurations for these underperforming latents result in very limited selection rates, indicating they are rarely chosen in both human and Gemini annotations. For human annotations across 6 tasks, the only configuration that yielded any selection was the one with *4 steps back* and a *beam width of 3*, achieving a selection rate of 8.33%. All other configurations resulted in 0% selection. In contrast, none of the configurations showed any non-zero selection rate under Gemini annotations, reinforcing the lack of preference for these latents.

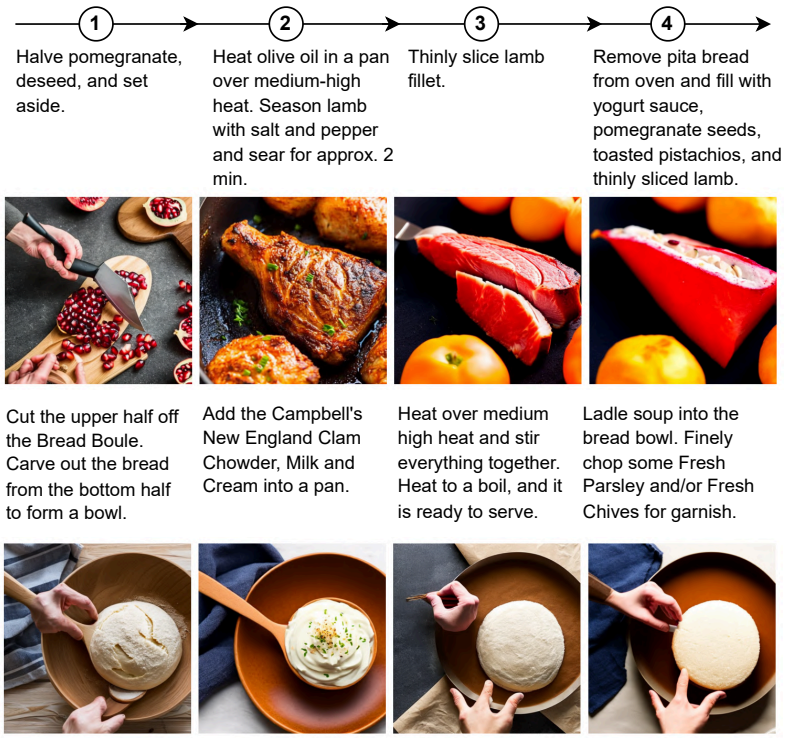


Figure 6.5: Impact of using 1, 3, 5, and 7 latent configurations on image generation.

Similarly, for Gemini annotations evaluated on the same 6 sequences, the configurations do not yield any notable results, with no selection rates. When the evaluation is extended to 35 sequences, the performance for latents 1, 3, 5, and 7 remains sparse, with a single configuration (*2 steps back, beam width of 3*) being selected, with a selection rate of 2.86%. Figure 6.5 visually demonstrates the poor results, where the images appear nearly identical, further supporting the findings of limited diversity and low performance, which aligns with the findings of [48, 55].

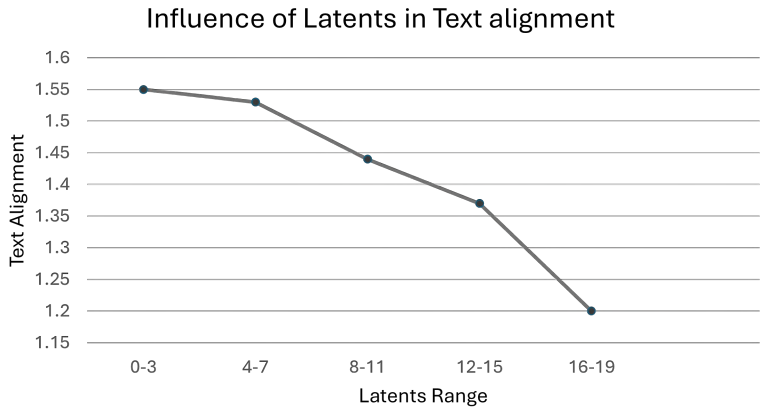


Figure 6.6: The average influence of latent indices on text alignment across task data.

Explainability Maps. Given the observed negative impact of higher latents on performance in human annotations, we investigated how latents influence text-image alignment using the DAAM (Diffusion Attention Attribution Map) framework [72], described in Section 3.2. DAAM interprets text-to-image diffusion models by analyzing cross-attention maps during the denoising process, generating heat maps that reveal how strongly each prompt token affects different regions of the output image.

By applying DAAM, we examined whether higher latents distort or weaken the influence of key tokens during generation. The resulting heat maps allowed us to visually assess token-level contributions, helping us understand whether the model continues to attend to the most semantically important parts of the prompt under varying latent configurations.

To quantify how well the generated images aligned with the text, we calculated the average heat map maximum intensity for all tokens, providing a numerical measure of how well the model preserved the key textual elements in the generated image. Then, for each sequence of generated images, we computed the average alignment score across all images in the sequence. While this serves as an objective metric for evaluating text adherence, it remains a partial measure, as it focuses solely on text alignment without accounting for other important aspects of image quality, such as coherence, realism, or artistic fidelity. Figure 6.6 shows the average impact of different latent groups on text alignment. The results reveal a clear pattern, latents from earlier iterations contribute more effectively to aligning the text with the generated image, while text alignment weakens as we move to higher latent ranges. Specifically, latents in the 0–3 range showed the strongest alignment, with an average score of 1.544, followed by the 4–7 range (1.523). In contrast, latents from the highest index range (16–19) had the weakest influence on alignment, with a significant drop in average scores. This suggests that earlier-stage latents play a crucial role in preserving text fidelity, whereas later-stage latents may dilute or distort the contribution of key tokens.

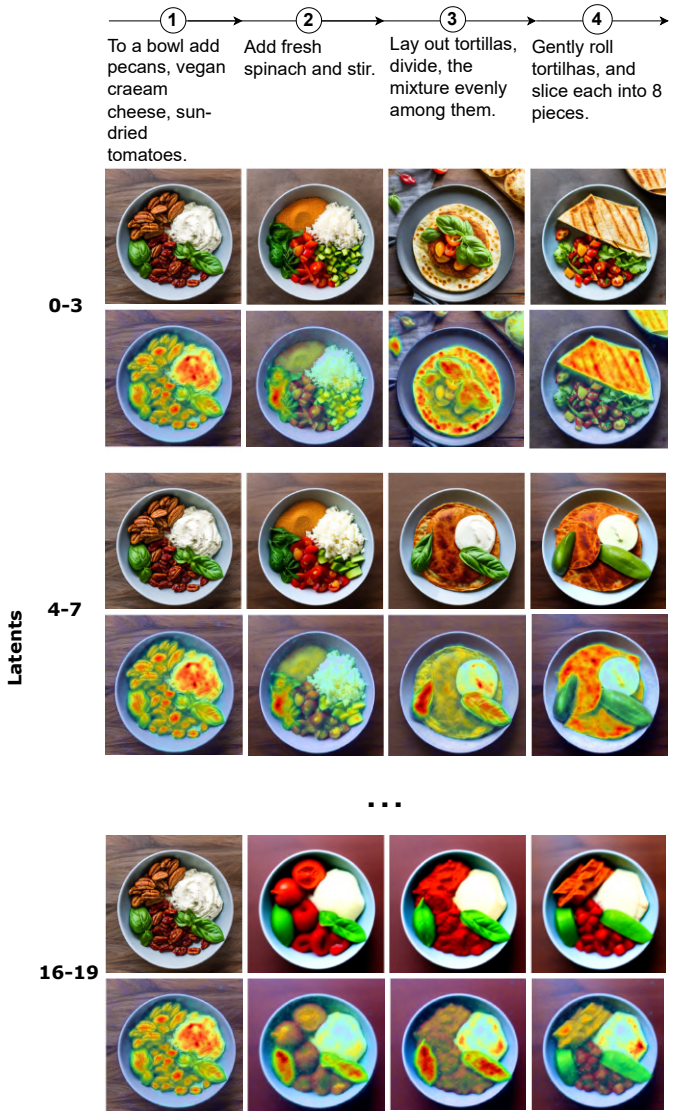


Figure 6.7: Visualization of cross-attention in latent diffusion across different sets of latents. The heat maps reveal how attention patterns evolve at various sequence steps, highlighting the alignment between text prompts and distinct latent representations.

In Figure 6.7, we analyze text alignment across multiple steps using heat maps corresponding to different latent groups, revealing how the connection between the input text and the generated image evolves throughout the process. Certain areas, such as the background and the bowl, show little to no heat map activation, indicating that these regions are less influenced by the text and more shaped by the model's learned priors and overall scene composition. Since the model generates new images by building on latents from previous steps, these early decisions about text relevance persist throughout the process, influencing later refinements. Additionally, when examining generations using later latent groups, we find that this information remains more stable across all steps. This suggests that latents at these stages carry more structural and stylistic details, focusing on consistency rather than continuous adjustments based on the input text.

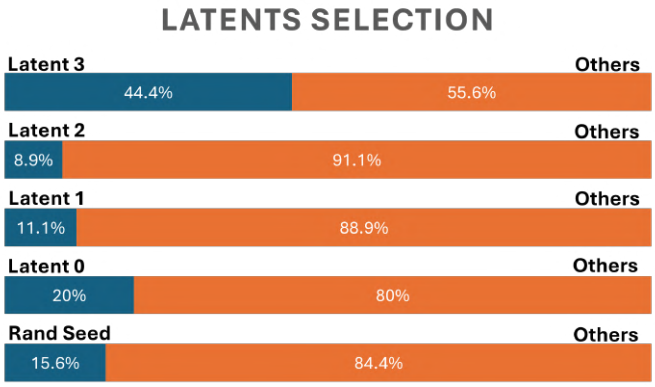


Figure 6.8: The average number of times that the model selected a given **latent** to generate the next image.

While later latents are effective at preserving visual consistency from previous steps, they are less capable of incorporating new details from the input prompts. For example, later latent groups (16-19) maintain stability in visual elements throughout the process, preserving previously established features. However, this comes at the expense of adapting to new or changing textual instructions. In contrast, early latent groups (0-3) demonstrate strong alignment between the text and the generated image, with consistently high heat map intensity across all steps. This highlights their role in grounding the image in the input text. For later latent groups (4-7), however, we observe a drop in heat map intensity, particularly by step 3, suggesting the model shifts focus away from strict textual adherence and toward refining finer visual details.

6.6.2 Latent Selection on Image Sequence Generation

After identifying Latents 0–3 as the most effective group for maintaining text-image alignment, we further analyzed the individual contributions of each latent within this group. Figure 6.8 shows the selection frequencies for each of these latents during sequence generation. Latent 3 was selected most frequently (44.4%), followed by Latent 0 (20%). In contrast, Latents 1 and 2 were chosen far less often (11.1% and 8.9%, respectively).

This uneven selection pattern shows that Latents 0 and 3 play complementary but crucial roles in generating sequences with strong text alignment. When Latent 0 is selected, it already carries contextual information from the prompt, making it responsive to the current sequence. Latent 3, on the other hand, contains additional information beyond the context, reinforcing key visual elements and providing stronger grounding. From a steerability perspective, these characteristics explain why beam search frequently selects these latents: they effectively guide the generation process while maintaining semantic and visual consistency.

Meanwhile, the lower usage of Latents 1 and 2 may indicate that they contribute less relevant or more redundant information, offering minimal benefit to prompt-driven

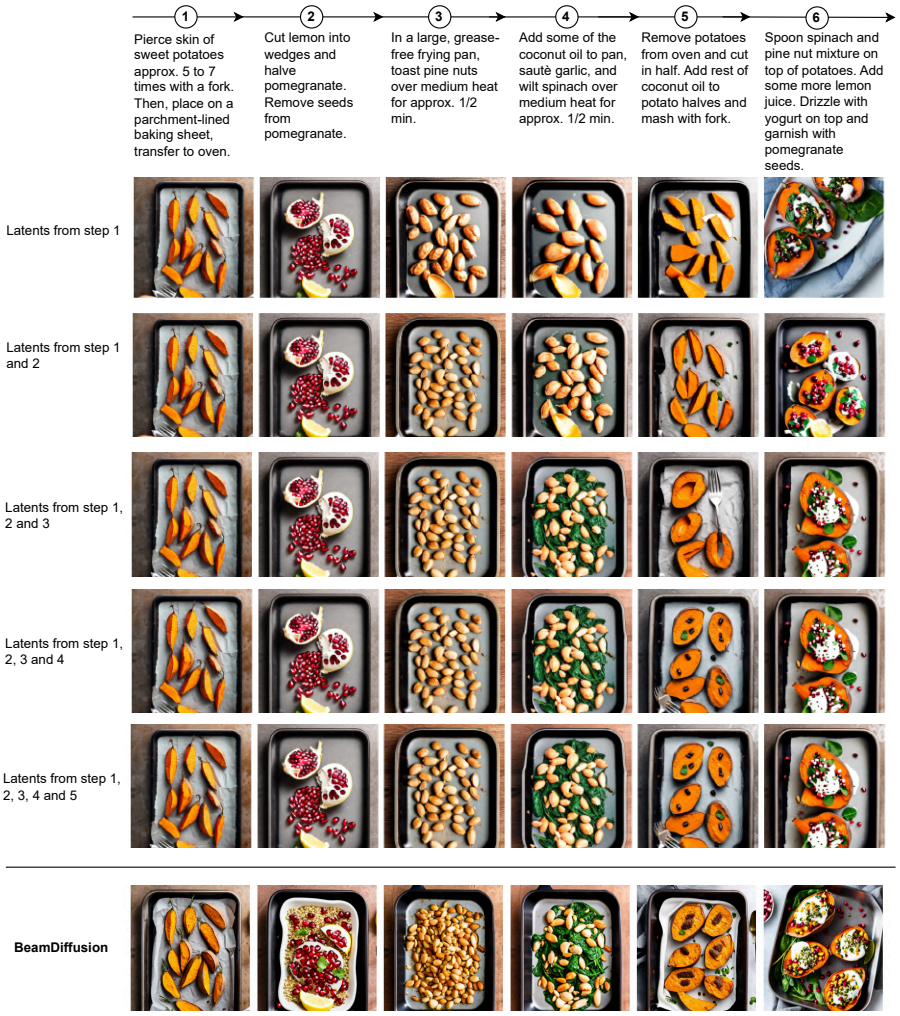


Figure 6.9: Visual comparison of sequences generated with cumulative latent access. Each row shows a sequence generated using latents from only the first step, the first two steps, and so on, up to all previous steps. The bottom row shows the result from BeamDiffusion, which uses latents from only the two preceding steps.

generation. Interestingly, the randomized latent seed (Rand Seed), which introduces variability into the generation process, accounted for 15.6% of selections. This highlights the importance of diversity: while beam search prefers reliable latents like 0 and 3, it also leverages stochasticity to explore alternative generation paths. This helps avoid local optima and contributes to the overall robustness and creativity of the output.

6.6.3 Latent Constrained BeamDiffusion

To investigate how access to latents from previous steps influences the quality of generation, we conducted an ablation study in which we constrain the set of available latents throughout the generation of the sequence. Specifically, we evaluate performance when the model is limited to latents from only the first step, then from the first two steps, and so on, up to including latents from all previous steps (i.e., full cumulative access).

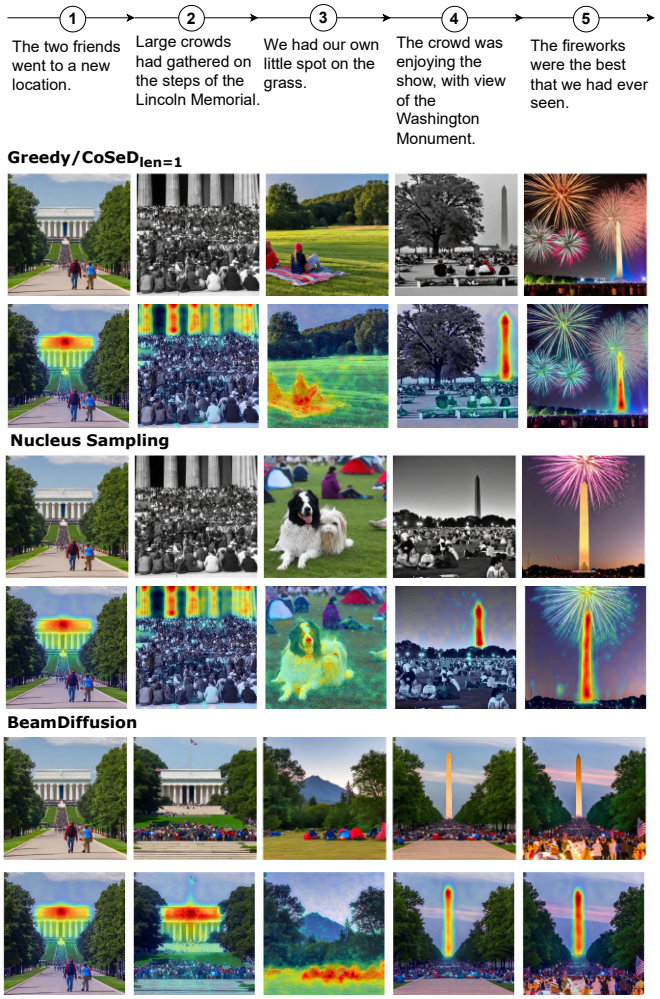


Figure 6.10: Heat maps illustrating cross-attention in latent diffusion across different decoding methods. The visualizations show variations in how each approach preserves or redistributes attention to textual cues throughout the generation process.

As shown in Figure 6.9, limiting the model to latents from only the first step results in weak text-image alignment and incoherent visual transitions. Gradually expanding the latent set to include more steps improves sequence quality, particularly when moving from two to three steps. However, we observe that including latents from all previous steps does not yield further improvements in coherence, suggesting diminishing returns with increasing temporal access.

In contrast, BeamDiffusion uses latents from only the two most recent steps and achieves results that are comparable, or even superior, to the full cumulative case. This indicates that a narrow temporal window is sufficient to maintain strong alignment and visual consistency. By avoiding the computational cost of considering all previous steps, BeamDiffusion finds a favorable trade-off between quality and efficiency.

Overall, this experiment highlights that cumulative latent access enhances generation quality up to a point, but full historical memory is unnecessary. Strategies such as BeamDiffusion, which take advantage of a limited but informative temporal context, can

produce high-quality sequences while significantly reducing computational overhead. These findings also underscore the importance of the non-linear sequence denoising strategy introduced in Section 6.3, which enables more targeted and effective reuse of past latents without the need for full sequence access.

6.7 Latent Search Methods and Text-Image Alignment

Figure 6.10 illustrates how different latent search methods affect text-image alignment. The heat maps show how attention patterns shift depending on the decoding strategy, revealing variations in how well each method preserves focus on key textual cues. Notably, in step 3, BeamDiffusion exhibits superior alignment, maintaining a stronger connection to the prompt and ensuring greater sequence coherence compared to other methods.

While Beam Diffusion generally achieves better visual consistency and stronger text adherence in earlier steps, it slightly underperforms in step 5, failing to capture the "fireworks" mentioned in the prompt. This highlights how even methods with strong overall performance can exhibit localized weaknesses, underscoring the importance of analyzing alignment step by step.

Overall, these findings highlight the significance of latent selection strategies in optimizing image generation. The results indicate that prioritizing the most significant latents could enhance the coherence of visual sequences and improve text alignment. These observations underline the influence of latent search strategies on the semantic accuracy and coherence of generated outputs.

7

Conclusions

This thesis introduced **BeamDiffusion, a novel method for navigating the latent space of diffusion models** using a beam search strategy to generate coherent and high-quality image sequences. By adapting principles from text-based sequence decoding, a framework was developed that maintains semantic alignment with textual prompts while preserving visual coherence across frames. **BeamDiffusion extends the capabilities of latent diffusion by treating image generation as a structured decoding task**, enabling more deliberate and interpretable control over sequence generation.

In the following sections we will discuss the key challenges that were overcomed, the main contributions of this thesis, the limitations and the broad impact of the current work. We conclude by offering a list of new research directions.

7.1 Research Challenges

While researching the problem of generating visual sequences, we encountered several challenges that required careful consideration and innovation. These challenges spanned multiple aspects, ranging from the difficulty of generating accurate and concise captions, to identifying suitable data for testing, and addressing the computational complexity of the generation process.

- **Latent spaces.** The method operates under several assumptions: BeamDiffusion relies on the **access to intermediate latent states during the denoising process**, which are necessary for steering the reverse diffusion trajectory. This design assumption, prevents it from being directly applied to frameworks that do not expose or allow access to such latents.

- **Domains.** While BeamDiffusion offers improved consistency, coherence, and contextual alignment, not all domains possess the complexity or variability necessary to demonstrate these strengths. Simpler tasks may fail to reveal the value of adaptive latent selection or hypothesis exploration, making careful domain selection essential for meaningful evaluation.

- **Efficiency.** Efficiency aspects particularly in large-scale or latency-sensitive applications is a key aspect of BeamDiffusion. Although strategies such as pruning and heuristic guidance can reduce cost, managing the quality-efficiency trade-off remains a challenge in practical settings.

While we were able to propose solutions for some of these challenges, others remain open and present ongoing areas of research. as will be discussed in the following sections.

7.2 Contributions and Achievements

This thesis research leveraged a key property of image synthesis methods: **the latent space of diffusion models can be actively guided**, rather than passively sampled. BeamDiffusion exploits this property by applying a non-linear, hypothesis-driven selection mechanism that navigates candidate latents, thus significantly enhancing generation quality. This approach bridges the gap between the flexibility of generative diffusion models and the structured reasoning required for multi-step visual synthesis.

On a broader level, the research establishes the **potential of integrating NLP decoding strategies into generative visual models**, offering a more unified paradigm for joint language-vision generation tasks. This opens up new possibilities for building systems that reason compositionally over both modalities. Furthermore, the method's modular and training-free nature positions it as a flexible component for future architectures focused on controllable and interpretable image generation.

Through extensive experimentation automatic and manual, it was demonstrated that **beam search improves sequence-level consistency** compared to traditional sampling-based approaches. **Sequential decoding was shown to be better suited for aligning multi-step textual instructions with image sequences**, as it can reason over intermediate representations and make globally informed decisions, which are essential for long-range coherence in narrative contexts.

A related study exploring similar ideas, including additional backbone architectures such as Diffusion Transformers (DiTs) and comparisons with more baselines, has been submitted to WACV 2026. For the full submitted paper, see Appendix B.

7.3 Limitations

The BeamDiffusion model generates multiple candidate image sequences by exploring diverse latents across a backward window of m steps and N latents per beam. We analyze the computational complexity across this generation process.

During the early steps, complexity is highest because pruning is deferred until after the second step. For each beam, the model explores $N \cdot m$ latents, and the total search space grows rapidly. Specifically, during step j, the complexity is proportional to $O(N \cdot m)^j$ for $j \leq 2$. For instance, at $j = 1$, the complexity scales with the number of initial random seeds.

By $j = 2$, the space becomes increasingly expensive due to combinatorial growth. After the second step, BeamDiffusion prunes the candidate set, retaining only the top w beams. This significantly reduces the search space, resulting in a complexity of approximately $O(N \cdot m \cdot w)$ in subsequent steps.

To further reduce computational cost, a heuristic based on text relevance can be employed to predict the top P most relevant past images. By restricting latent exploration to these P candidates, the model improves early-stage efficiency without sacrificing alignment. Although the initial steps remain compute-intensive, the combination of pruning and relevance heuristics provides a practical balance between quality and performance.

7.4 Broader Impact

The improvements offered by our work, can benefit applications in animation, visual storytelling, educational media, and accessibility, for instance, by enabling the automatic generation of illustrative sequences from text, thereby enhancing communication and understanding.

At the same time, the improved realism and consistency of generated content raise concerns around potential misuse. The ability to create highly coherent image sequences could facilitate the production of deceptive visual narratives, such as disinformation campaigns, propaganda, or deepfakes. Safeguards such as watermarking, provenance tracking, and responsible release strategies should be considered to mitigate these risks.

A continuous dialogue with stakeholders across policy, ethics, and technology domains will be essential for the responsible deployment of this work.

7.5 Future Work

This thesis demonstrated that sequence-level reasoning in latent diffusion models is both feasible and beneficial, enabling more structured, controllable, and semantically faithful image generation. Applications in areas such as **visual storytelling**, **instructional content**, and **step-by-step demonstrations** highlight the model's utility in scenarios requiring coherent visual narratives. However, several ideas remain unexplored, particularly regarding the fidelity of alignment between textual inputs and the resulting image sequences:

- **Early detection and correction of inconsistencies during denoising**. Rather than relying solely on the final generated outputs, future systems could assess alignment between text and visual content at intermediate stages of the diffusion process. **Vision-language models such as CLIP or VQAScore** may be adapted to evaluate semantic similarity during denoising, providing a real-time signal for guiding generation.

- **Compositional editing in latent space** may be explored as a means to guide generation and improve continuity. By manipulating latent components from earlier scenes,

such as background elements, character features, or object configurations, it becomes possible to apply fine-grained control and promote stronger coherence across sequences. This direction could enable new modes of interactive or conditional image editing within the generative process.

Taken together, these future directions aim to improve the **factual consistency, controllability, and reliability** of diffusion-based image generation systems. Such advances will be crucial for enabling safe and effective use of these models in **high-stakes, user-facing, or narrative-driven applications**, where consistency and correctness are essential.

Bibliography

[1] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[2] P. Grifoni and P. Grifoni. *Multimodal Human Computer Interaction and Pervasive Services*. USA: IGI Global, 2009. ISBN: 1605663867 (cit. on p. 1).

[3] R. Ahmed. "Reading the Visual: An Introduction to Teaching Multimodal Literacy : (Book Review)". In: *Journal of English Studies in Arabia Felix* 3.1 (2024-03), pp. 24–27. DOI: 10.56540/jesaf.v3i1.92. URL: https://journals.arafa.org/index.php/jesaf/article/view/92 (cit. on p. 1).

[4] R. Rombach et al. "High-Resolution Image Synthesis with Latent Diffusion Models". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 2022, pp. 10674–10685. DOI: 10.1109/CVPR52688.2022.01042. URL: https://doi.org/10.1109/CVPR52688.2022.01042 (cit. on pp. 1, 17, 19, 33, 42, 45).

[5] C. Saharia et al. "Photorealistic text-to-image diffusion models with deep language understanding". In: *Advances in neural information processing systems* 35 (2022), pp. 36479–36494 (cit. on p. 1).

[6] D. Podell et al. "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis". In: (2024). URL: https://openreview.net/forum?id=di52zR8xgf (cit. on p. 1).

[7] A. Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by I. Guyon et al. 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (cit. on pp. 6–8, 19, 33).

[8] S. University. *CS231n: Convolutional Neural Networks for Visual Recognition - Recurrent Neural Networks*. https://cs231n.github.io/rnn/. Accessed: 2025-01-17. n.d. (Cit. on p. 6).

[9] C. Olah. *LSTM Neural Network Tutorial*. https://web.stanford.edu/class/cs379c/archive/2018/class_messages_listing/content/Artificial_Neural_Network_Technology_Tutorials/OlahLSTM-NEURAL-NETWORK-TUTORIAL-15.pdf. Accessed: 2025-01-17. 2018 (cit. on p. 6).

[10] D. Jurafsky and J. H. Martin. *Speech and Language Processing (3rd ed.)* https://web.stanford.edu/~jurafsky/slp3/6.pdf. Accessed: 2025-01-17. 2025 (cit. on p. 6).

[11] A. Jaegle et al. "Perceiver IO: A General Architecture for Structured Inputs & Outputs". In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL: https://openreview.net/forum?id=fILj7WpI-g (cit. on p. 8).

[12] A. Jaegle et al. "Perceiver: General perception with iterative attention". In: *International conference on machine learning*. PMLR. 2021, pp. 4651–4664 (cit. on p. 8).

[13] S. Sonkar and R. G. Baraniuk. "Investigating the Role of Feed-Forward Networks in Transformers Using Parallel Attention and Feed-Forward Net Design". In: *CoRR* abs/2305.13297 (2023). DOI: 10.48550/ARXIV.2305.13297. arXiv: 2305.13297. URL: https://doi.org/10.48550/arXiv.2305.13297 (cit. on p. 9).

[14] J. Lederer. "Activation Functions in Artificial Neural Networks: A Systematic Overview". In: *CoRR* abs/2101.09957 (2021). arXiv: 2101.09957. URL: https://arxiv.org/abs/2101.09957 (cit. on p. 9).

[15] T. Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2013. URL: http://arxiv.org/abs/1301.3781 (cit. on p. 9).

[16] J. Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by J. Burstein, C. Doran, and T. Solorio. Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/V1/N19-1423. URL: https://doi.org/10.18653/v1/n19-1423 (cit. on p. 10).

[17] A. Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9 (cit. on p. 10).

[18] A. Zhang et al. "The Encoder-Decoder Architecture". In: *Dive into Deep Learning*. 2018. URL: https://d2l.ai/chapter_recurrent-modern/encoder-decoder.html (cit. on p. 10).

[19] C. Meister, G. Wiher, and R. Cotterell. "On Decoding Strategies for Neural Text Generators". In: *Trans. Assoc. Comput. Linguistics* 10 (2022), pp. 997–1012. DOI: 10.1162/TACL_A_00502. URL: https://doi.org/10.1162/tacl%5C_a%5C_00502 (cit. on p. 11).

[20] C. Shi et al. "A Thorough Examination of Decoding Methods in the Era of LLMs". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*. Ed. by Y. Al-Onaizan, M. Bansal, and Y. Chen. Association for Computational Linguistics, 2024, pp. 8601–8629. URL: https://aclanthology.org/2024.emnlp-main.489 (cit. on p. 11).

[21] K. Cho. "Noisy Parallel Approximate Decoding for Conditional Recurrent Language Model". In: *CoRR* abs/1605.03835 (2016). arXiv: 1605.03835. URL: http://arxiv.org/abs/1605.03835 (cit. on p. 11).

[22] A. Holtzman et al. "The Curious Case of Neural Text Degeneration". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=rygGQyrFvH (cit. on p. 12).

[23] M. Renze. "The Effect of Sampling Temperature on Problem Solving in Large Language Models". In: *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*. Ed. by Y. Al-Onaizan, M. Bansal, and Y. Chen. Association for Computational Linguistics, 2024, pp. 7346–7356. URL: https://aclanthology.org/2024.findings-emnlp.432 (cit. on p. 13).

[24] A. Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763. URL: http://proceedings.mlr.press/v139/radford21a.html (cit. on pp. 13, 14, 48).

[25] J. Li et al. "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation". In: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*. Ed. by K. Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 12888–12900. URL: https://proceedings.mlr.press/v162/li22n.html (cit. on p. 14).

[26] M. Oquab et al. "DINOv2: Learning Robust Visual Features without Supervision". In: *Trans. Mach. Learn. Res.* 2024 (2024). URL: https://openreview.net/forum?id=a68SUt6zFt (cit. on pp. 15, 48).

[27] D. Jiang et al. "From clip to dino: Visual encoders shout in multi-modal large language models". In: *arXiv preprint arXiv:2310.08825* (2023) (cit. on p. 15).

[28] J. Sohl-Dickstein et al. "Deep unsupervised learning using nonequilibrium thermodynamics". In: *International conference on machine learning*. PMLR. 2015, pp. 2256–2265 (cit. on p. 16).

[29] J. Ho, A. Jain, and P. Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by H. Larochelle et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html (cit. on p. 16).

[30] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*. Ed. by N. Navab et al. Vol. 9351. Lecture Notes in Computer Science. Springer, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28. URL: https://doi.org/10.1007/978-3-319-24574-4%5C_28 (cit. on pp. 17, 33).

[31] AssemblyAI. *How Imagen Actually Works*. Accessed: 2025-01-31. 2023-04. URL: https://www.assemblyai.com/blog/how-imagen-actually-works/ (cit. on p. 18).

[32] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer. 2015, pp. 234–241 (cit. on p. 18).

[33] D. Bank, N. Koenigstein, and R. Giryes. *Autoencoders*. 2021. arXiv: 2003.05991 [cs.LG]. URL: https://arxiv.org/abs/2003.05991 (cit. on p. 19).

[34] R. Tang et al. *What the DAAM: Interpreting Stable Diffusion Using Cross Attention*. Ed. by A. Rogers, J. L. Boyd-Graber, and N. Okazaki. 2023. DOI: 10.18653/V1/2023.ACL-LONG.310. URL: https://doi.org/10.18653/v1/2023.acl-long.310 (cit. on pp. 20, 21).

[35] S. Dewan et al. "DiffusionPID: Interpreting Diffusion via Partial Information Decomposition". In: *CoRR* abs/2406.05191 (2024). DOI: 10.48550/ARXIV.2406.05191. arXiv: 2406.05191. URL: https://doi.org/10.48550/arXiv.2406.05191 (cit. on p. 20).

[36] Y. Park et al. "Understanding the Latent Space of Diffusion Models through the Lens of Riemannian Geometry". In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Ed. by A. Oh et al. 2023. URL: http://papers.nips.cc/paper%5C_files/paper/2023/hash/4bfcebedf7a2967c410b64670f27f904-Abstract-Conference.html (cit. on p. 21).

[37] J. Mao, X. Wang, and K. Aizawa. "Guided Image Synthesis via Initial Image Editing in Diffusion Model". In: *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*. Ed. by

A. El-Saddik et al. ACM, 2023, pp. 5321–5329. DOI: 10.1145/3581783.3612191. URL: https://doi.org/10.1145/3581783.3612191 (cit. on pp. 22, 23).

[38] I. J. Goodfellow et al. "Generative Adversarial Networks". In: *CoRR* abs/1406.2661 (2014). arXiv: 1406.2661. URL: http://arxiv.org/abs/1406.2661 (cit. on p. 23).

[39] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2014. URL: http://arxiv.org/abs/1312.6114 (cit. on p. 23).

[40] K. Tian et al. "Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction". In: *CoRR* abs/2404.02905 (2024). DOI: 10.48550/ARXIV.2404.02905. arXiv: 2404.02905. URL: https://doi.org/10.48550/arXiv.2404.02905 (cit. on p. 23).

[41] W. Peebles and S. Xie. "Scalable Diffusion Models with Transformers". In: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 2023, pp. 4172–4182. DOI: 10.1109/ICCV51070.2023.00387. URL: https://doi.org/10.1109/ICCV51070.2023.00387 (cit. on p. 23).

[42] M. Geyer et al. "TokenFlow: Consistent Diffusion Features for Consistent Video Editing". In: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL: https://openreview.net/forum?id=lKK50q2MtV (cit. on p. 24).

[43] Y. Liu et al. "SyncDreamer: Generating Multiview-consistent Images from a Single-view Image". In: (2024). URL: https://openreview.net/forum?id=MN3yH2ovHb (cit. on p. 24).

[44] W. Chen et al. "ID-Aligner: Enhancing Identity-Preserving Text-to-Image Generation with Reward Feedback Learning". In: *arXiv preprint arXiv:2404.15449* (2024) (cit. on p. 24).

[45] L. Khachatryan et al. "Text2video-zero: Text-to-image diffusion models are zero-shot video generators". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 15954–15964 (cit. on p. 24).

[46] J. Y. Koh, D. Fried, and R. Salakhutdinov. "Generating Images with Multimodal Language Models". In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*. Ed. by A. Oh et al. 2023. URL: http://papers.nips.cc/paper%5C_files/paper/2023/hash/43a69d143273bd8215578bde887bb552-Abstract-Conference.html (cit. on pp. 24, 25, 45, 52, 53, 55).

[47] T. Rahman et al. "Make-A-Story: Visual Memory Conditioned Consistent Story Generation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 2493–2502. DOI: 10.1109/CVPR52729.2023.00246. URL: https://doi.org/10.1109/CVPR52729.2023.00246 (cit. on pp. 25, 33).

[48] J. Bordalo et al. "Generating Coherent Sequences of Visual Illustrations for Real-World Manual Tasks". In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*. Ed. by L. Ku, A. Martins, and V. Srikumar. Association for Computational Linguistics, 2024, pp. 12777–12797. DOI: 10.18653/V1/2024.ACL-LONG.690. URL: https://doi.org/10.18653/v1/2024.acl-long.690 (cit. on pp. 26, 33, 42, 44, 48, 62).

[49] X. Pan et al. "Synthesizing Coherent Story with Auto-Regressive Latent Diffusion Models". In: *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2024, Waikoloa, HI, USA, January 3-8, 2024*. IEEE, 2024, pp. 2908–2918. DOI: 10.1109/WACV57701.2024.00290. URL: https://doi.org/10.1109/WACV57701.2024.00290 (cit. on p. 26).

[50] Y. Shen et al. "Many-to-many Image Generation with Auto-regressive Diffusion Models". In: *CoRR* abs/2404.03109 (2024). DOI: 10.48550/ARXIV.2404.03109. arXiv: 2404.03109. URL: https://doi.org/10.48550/arXiv.2404.03109 (cit. on p. 27).

[51] S. Menon, I. Misra, and R. Girdhar. "Generating Illustrated Instructions". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*. IEEE, 2024, pp. 6274–6284. DOI: 10.1109/CVPR52733.2024.00600. URL: https://doi.org/10.1109/CVPR52733.2024.00600 (cit. on p. 28).

[52] H. Bansal et al. "TALC: Time-Aligned Captions for Multi-Scene Text-to-Video Generation". In: *CoRR* abs/2405.04682 (2024). DOI: 10.48550/ARXIV.2405.04682. arXiv: 2405.04682. URL: https://doi.org/10.48550/arXiv.2405.04682 (cit. on pp. 28, 29).

[53] A. Blattmann et al. "Align Your Latents: High-Resolution Video Synthesis with Latent Diffusion Models". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 22563–22575. DOI: 10.1109/CVPR52729.2023.02161. URL: https://doi.org/10.1109/CVPR52729.2023.02161 (cit. on p. 29).

[54] S. Yin et al. "NUWA-XL: Diffusion over Diffusion for eXtremely Long Video Generation". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*. Ed. by A. Rogers, J. L. Boyd-Graber, and N. Okazaki. Association for Computational

Linguistics, 2023, pp. 1309–1320. DOI: 10.18653/V1/2023.ACL-LONG.73. URL: https://doi.org/10.18653/v1/2023.acl-long.73 (cit. on p. 29).

[55] V. Ramos et al. "Contrastive Sequential-Diffusion Learning: An approach to Multi-Scene Instructional Video Synthesis". In: *CoRR* abs/2407.11814 (2024). DOI: 10.48550/ARXIV.2407.11814. arXiv: 2407.11814. URL: https://doi.org/10.48550/arXiv.2407.11814 (cit. on pp. 29, 30, 33, 38, 45, 48, 52, 53, 55, 62).

[56] H. Lin et al. "VideoDirectorGPT: Consistent Multi-scene Video Generation via LLM-Guided Planning". In: *CoRR* abs/2309.15091 (2023). DOI: 10.48550/ARXIV.2309.15091. arXiv: 2309.15091. URL: https://doi.org/10.48550/arXiv.2309.15091 (cit. on pp. 30, 31).

[57] Y. Zhou et al. "StoryDiffusion: Consistent Self-Attention for Long-Range Image and Video Generation". In: *CoRR* abs/2405.01434 (2024). DOI: 10.48550/ARXIV.2405.01434. arXiv: 2405.01434. URL: https://doi.org/10.48550/arXiv.2405.01434 (cit. on p. 30).

[58] J. Wang et al. "ModelScope Text-to-Video Technical Report". In: *CoRR* abs/2308.06571 (2023). DOI: 10.48550/ARXIV.2308.06571. arXiv: 2308.06571. URL: https://doi.org/10.48550/arXiv.2308.06571 (cit. on p. 30).

[59] M. Smilevski, I. Lalkovski, and G. Madjarov. "Stories for images-in-sequence by using visual and narrative components". In: *ICT Innovations 2018. Engineering and Life Sciences: 10th International Conference, ICT Innovations 2018, Ohrid, Macedonia, September 17–19, 2018, Proceedings 10*. Springer. 2018, pp. 148–159 (cit. on p. 33).

[60] C. Liu et al. "Intelligent Grimm - Open-ended Visual Storytelling via Latent Diffusion Models". In: *CVPR*. 2024. DOI: 10.1109/CVPR52733.2024.00592. URL: https://doi.org/10.1109/CVPR52733.2024.00592 (cit. on pp. 33, 48).

[61] T. Soucek et al. "GenHowTo: Learning to Generate Actions and State Transformations from Instructional Videos". In: *CVPR*. 2024. DOI: 10.1109/CVPR52733.2024.00627. URL: https://doi.org/10.1109/CVPR52733.2024.00627 (cit. on p. 33).

[62] A. Bertsch et al. "It's MBR All the Way Down: Modern Generation Techniques Through the Lens of Minimum Bayes Risk". In: *Proceedings of the Big Picture Workshop*. Ed. by Y. Elazar et al. Singapore: Association for Computational Linguistics, 2023-12, pp. 108–122. DOI: 10.18653/v1/2023.bigpicture-1.9. URL: https://aclanthology.org/2023.bigpicture-1.9 (cit. on p. 36).

[63] S. Wiseman and A. M. Rush. "Sequence-to-Sequence Learning as Beam-Search Optimization". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. Ed. by J. Su, X. Carreras, and K. Duh. The Association for Computational Linguistics, 2016, pp. 1296–1306. DOI: 10.18653/V1/D16-1137. URL: https://doi.org/10.18653/v1/d16-1137 (cit. on p. 37).

[64] M. Reid et al. "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context". In: *CoRR* abs/2403.05530 (2024). DOI: 10.48550/ARXIV.2403.05530. arXiv: 2403.05530. URL: https://doi.org/10.48550/arXiv.2403.05530 (cit. on pp. 41, 44).

[65] T.-H. K. Huang et al. "Visual Storytelling". In: *15th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*. 2016 (cit. on pp. 42, 44).

[66] S. Menon, I. Misra, and R. Girdhar. "Generating Illustrated Instructions". In: *CVPR*. 2024. DOI: 10.1109/CVPR52733.2024.00600. URL: https://doi.org/10.1109/CVPR52733.2024.00600 (cit. on pp. 45, 49, 52, 53, 55).

[67] A. Holtzman et al. "The Curious Case of Neural Text Degeneration". In: *International Conference on Learning Representations*. 2020 (cit. on pp. 45, 52, 53, 55).

[68] J. Fleiss. "Measuring Nominal Scale Agreement Among Many Raters". In: *Psychological Bulletin* 76 (1971-11), pp. 378–. DOI: 10.1037/h0031619 (cit. on pp. 48, 60).

[69] Z. Wang et al. "T2IShield: Defending Against Backdoors on Text-to-Image Diffusion Models". In: *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXV*. Ed. by A. Leonardis et al. Vol. 15143. Lecture Notes in Computer Science. Springer, 2024, pp. 107–124. DOI: 10.1007/978-3-031-73013-9_7. URL: https://doi.org/10.1007/978-3-031-73013-9%5C_7 (cit. on p. 48).

[70] S. Lin et al. "Explore the Power of Synthetic Data on Few-shot Object Detection". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 638–647. DOI: 10.1109/CVPRW59228.2023.00071. URL: https://doi.org/10.1109/CVPRW59228.2023.00071%7D (cit. on p. 48).

[71] J. Pont-Tuset et al. "Connecting vision and language with localized narratives". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer. 2020, pp. 647–664 (cit. on p. 61).

[72] R. Tang et al. "What the DAAM: Interpreting Stable Diffusion Using Cross Attention". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*. Ed. by A. Rogers, J. L. Boyd-Graber, and N. Okazaki. Association for Computational Linguistics, 2023, pp. 5644–5659. DOI: 10.18653/V1/2023.ACL-LONG.310. URL: https://doi.org/10.18653/v1/2023.acl-long.310 (cit. on pp. 61, 63).

[73] S. Dewan et al. "DiffusionPID: Interpreting Diffusion via Partial Information Decomposition". In: *CoRR* abs/2406.05191 (2024). DOI: 10.48550/ARXIV.2406.05191. arXiv: 2406.05191. URL: https://doi.org/10.48550/arXiv.2406.05191 (cit. on p. 61).

[74] R. E. Mayer. "Multimedia learning". In: vol. 41. Psychology of Learning and Motivation. Academic Press, 2002, pp. 85–139. DOI: https://doi.org/10.1016/S0079-7421(02)80005-6. URL: https://www.sciencedirect.com/science/article/pii/S0079742102800056 (cit. on p. 82).

[75] Y. Lu et al. "Multimodal Procedural Planning via Dual Text-Image Prompting". In: *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*. Ed. by Y. Al-Onaizan, M. Bansal, and Y. Chen. Association for Computational Linguistics, 2024, pp. 10931–10954. URL: https://aclanthology.org/2024.findings-emnlp.641 (cit. on p. 82).

A

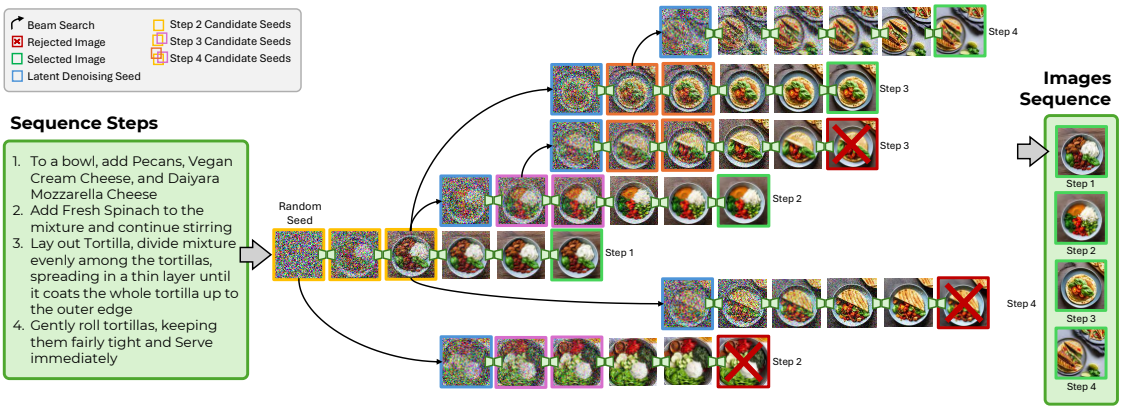
Application Examples

This section presents additional visual examples and qualitative results that complement the main paper. Images and other visual aids are essential tools that significantly improve comprehension across a variety of domains. Whether breaking down complex tasks, illustrating a sequence of events, or clarifying intricate concepts, visual representations provide clear, step-by-step guidance that enhances understanding [74]. From technical procedures and instructional design to storytelling and educational content, multi-scene image sequences help readers grasp information more intuitively, transforming even the most complex or dense topics into something more accessible. However, generating a sequence of images that not only aligns with each textual instruction but also maintains overall coherence remains a major challenge [75]. In this appendix, we include expanded examples to further illustrate how our model addresses this issue. These include additional visualizations of the beam search process and extended qualitative comparisons across domains such as recipes, visual storytelling, and DIY tasks. Together, these examples highlight the strengths and limitations of current approaches in achieving textual alignment and visual consistency in sequential image generation.

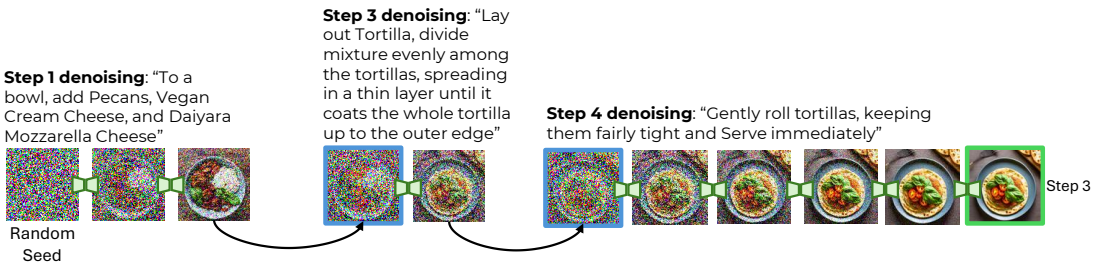
A.1 Beam Search Tree

In Figure A.1a, we provide another example of how the latent search occurs within the beam denoising process. This illustrates the BeamDiffusion model at work, showing the denoising of the initial seed to latents and how different beam paths are decoded. The figure highlights how our method explores the latent space, iterating through various latents based on the beam search strategy.

In Figure A.1b, we observe that the generation of the selected image at step 4 behaves similarly to an isolated diffusion process. However, it is important to note that this process is influenced by multiple prompts. Specifically, the image generation at step 4 is conditioned on the prompts from steps 1, 3, and 4. This demonstrates how the latent search process incorporates context from various steps, showing that even though each generation step seems independent, it is shaped by previous steps, enabling a more



(a) The beam denoising search tree.



(b) Example of one complete decoded denoising process using contextual captions.

Figure A.1: The beam diffusion model works in the denoising latent space by denoising the initial seed to latents and them decoding different beam paths. This shows how our method evolves in the latent space and how it can explore the latent space.

coherent and consistent generation of the final image.

A.2 Sequence Examples

To provide a comprehensive comparison between all models, we present several examples across different domains in Figures A.2-A.13. These figures show case how each method performs in terms of maintaining sequence coherence throughout sequential image generation.

In our overall qualitative analysis, we observe that both Greedy/CoSeD$_{len=1}$ and Nucleus Sampling methods are generally good at following the textual input across steps, as shown in Figures A.2, A.5, A.8, and A.9. They align well with the step instructions but often struggle with visual consistency, as the generated images show noticeable shifts or inconsistencies between steps. This lack of visual cohesion detracts from the overall coherence of the sequence, even though they adhere reasonably well to the textual prompt. In contrast, GILL and StackedDiffusion exhibit a different behavior. While both maintain high visual consistency, generating sequences in which the images are visually

similar, they struggle to accurately follow the step-by-step textual instructions. GILL, in particular, tends to produce images that lack the necessary variability to incorporate new information from each step, resulting in sequences that are visually cohesive but contextually incorrect or incomplete. StackedDiffusion performs slightly better than GILL in this regard, capturing some of the step-wise changes, but still often fails to fully reflect the evolving textual input across the sequence.

Finally, BeamDiffusion strikes a strong balance between text adherence and visual consistency. As shown in Figures A.2, A.5, A.8, and A.9, it generally produces the most coherent sequences compared to other methods. In most cases, BeamDiffusion effectively follows the textual instructions while maintaining visual consistency across the steps. This combination of textual alignment and visual stability leads to more coherent and contextually accurate image sequences. While not without its limitations, BeamDiffusion demonstrates a more reliable progression throughout the image generation process than the other approaches. However, it is important to note that all methods face challenges in maintaining coherence within the DIY domain, as seen in Figures A.10, A.11, and A.13. The complexity and diversity of DIY tasks present unique difficulties for the models, making it harder to both accurately follow textual instructions and ensure consistency throughout the sequence. While the models perform well in other domains, the intricacies of the DIY domain result in some inconsistencies, particularly when capturing step changes.

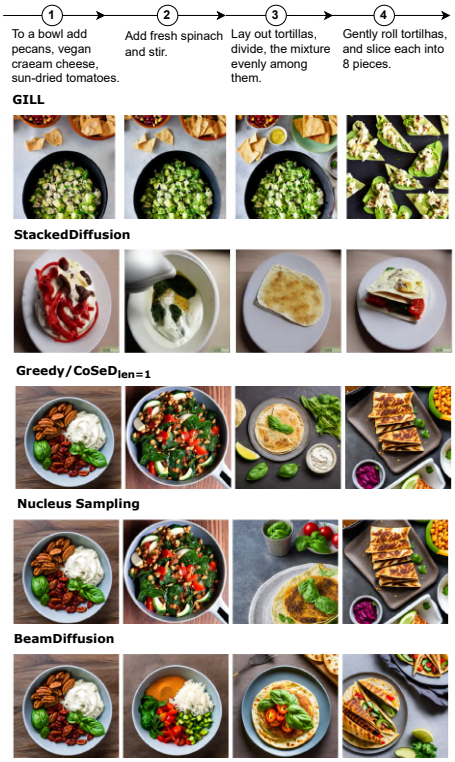


Figure A.2: Example of the Recipes domain task *"Vegan Basil Pesto and Sun-Dried Tomato Pinwheels"*, with a sequence of 4 steps.

Figure A.3: Example of the Recipes domain task *"Tangy lemon sheet cake"*, with a sequence of 4 steps.

Figure A.4: Example of the Recipes domain task *"Zucchini, Pecorino & Mint Salad"*, with a sequence of 4 steps.

Figure A.5: Example of the Recipes domain task *"Bone Broth"*, with a sequence of 3 steps.

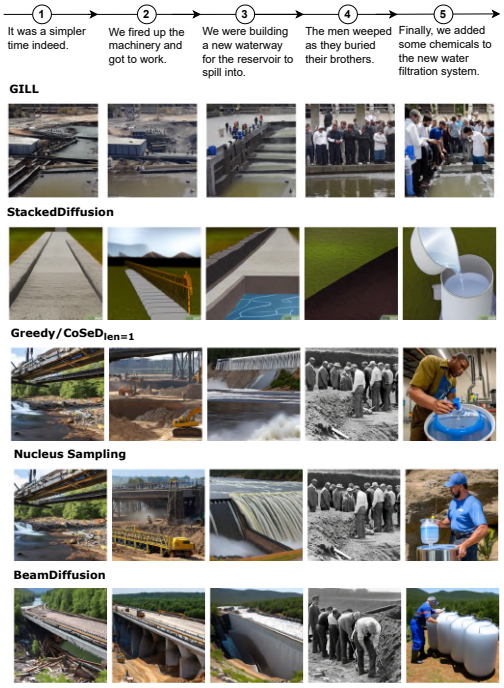


Figure A.6: Example of the out-of-domain VIST story *"Klamath Project Construction"*, with a sequence of 5 steps.

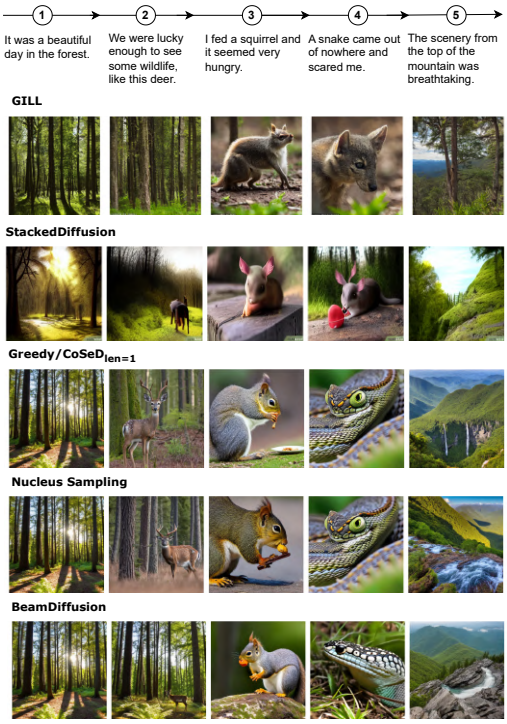


Figure A.7: Example of the out-of-domain VIST story *"Yosemite Weekend"*, with a sequence of 5 steps.

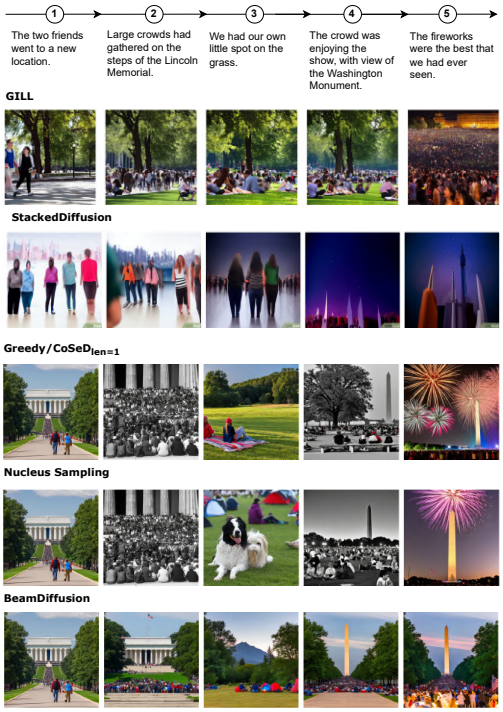


Figure A.8: Example of the out-of-domain VIST story *"Fireworks - 4th July in Washington, DC"*, with a sequence of 5 steps.

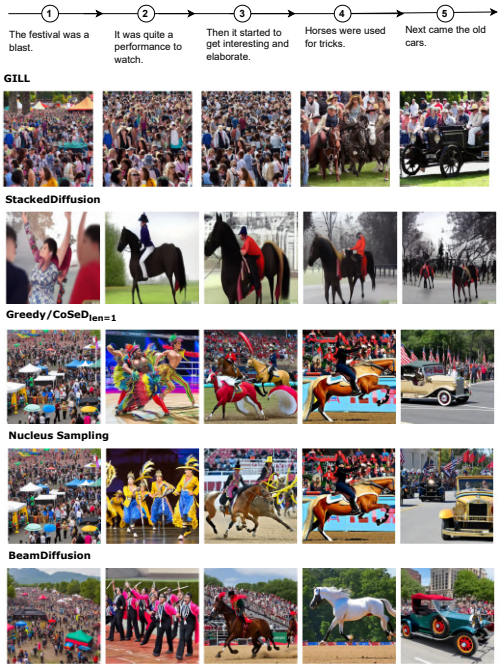


Figure A.9: Example of the out-of-domain VIST story *"Independence Day Parade"*, with a sequence of 5 steps.

Figure A.10: Example of the out-of-domain DIY task *"How to Clean Silk Rugs"*, with a sequence of 2 steps.

Figure A.11: Example of the out-of-domain DIY task *"How to Polish Wood"*, with a sequence of 3 steps.

Figure A.12: Example of the out-of-domain DIY task *"How to Preserve Basil"*, with a sequence of 6 steps.

Figure A.13: Example of the out-of-domain DIY task *"How to Clean Natural Stone"*, with a sequence of 5 steps.

B

Submitted Paper

This appendix includes a paper that has been submitted to **The IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) 2026** in connection with the main thesis idea. While the paper shares the same core concept, it extends the study by evaluating additional backbone architectures, including Diffusion Transformers (DiTs), and by comparing with a broader set of baseline methods. The submitted work serves as supporting material for the analyses and results presented in the main chapters of this thesis.

Latent Beam Diffusion Models for Decoding Image Sequences

Anonymous WACV Algorithms Track submission

Paper ID *****

Abstract

While diffusion models excel at generating high-quality images from text prompts, they struggle with visual consistency in image sequences. Existing methods generate each image independently, leading to disjointed narratives — a challenge further exacerbated in non-linear storytelling, where scenes must connect beyond adjacent frames. We introduce a novel beam search strategy for latent space exploration, enabling conditional generation of full image sequences with beam search decoding. Unlike prior approaches that use fixed latent priors, our method dynamically searches for an optimal sequence of latent representations, ensuring coherent visual transitions. As the latent denoising space is explored, the beam search graph is pruned with a cross-attention mechanism that efficiently scores search paths, prioritizing alignment with both textual prompts and visual context. Human and automatic evaluations confirm that BeamDiffusion outperforms other baseline methods, producing full sequences with superior coherence, visual continuity, and textual alignment. All data and code implementations will be released after publication.

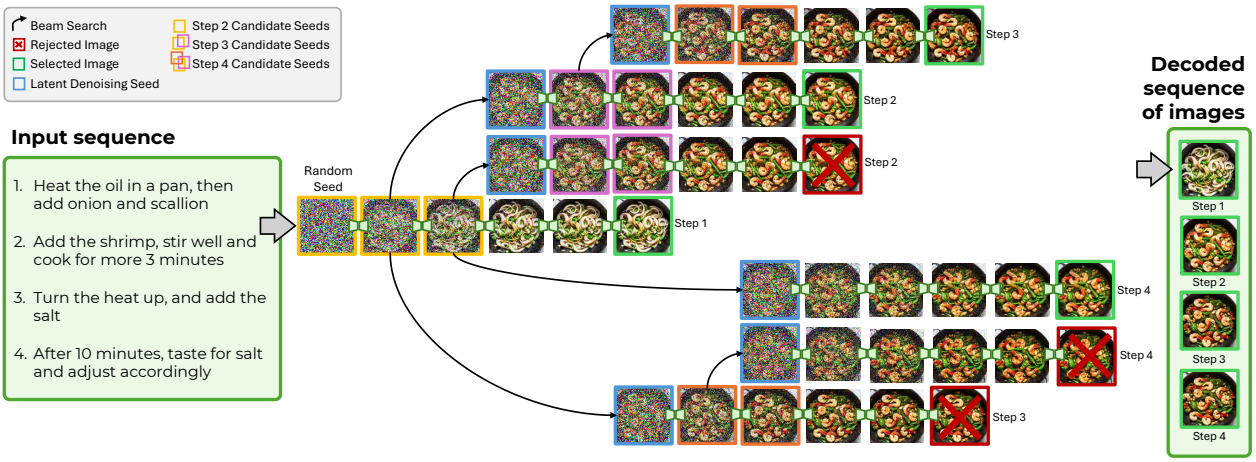
1. Introduction

Image diffusion models [11, 47, 48] have made significant advancements in generating high-quality images. These models, especially when combined with text prompts, have shown great potential in producing detailed and contextually accurate visuals [33, 39, 42]. Despite the impressive results at generating individual images based on specific text prompts, diffusion models face challenges when it comes to creating coherent sequences of images. In most cases, each image is generated independently based on its own prompt, which does not naturally ensure visual continuity across a *sequence of correlated prompts*. This problem is further compounded when the narrative is non-linear, where a scene may be connected not only to the immediate prior scene but also to earlier scenes in the sequence. This dynamic can result in sequences that lack coherence and visual consistency over time [3].

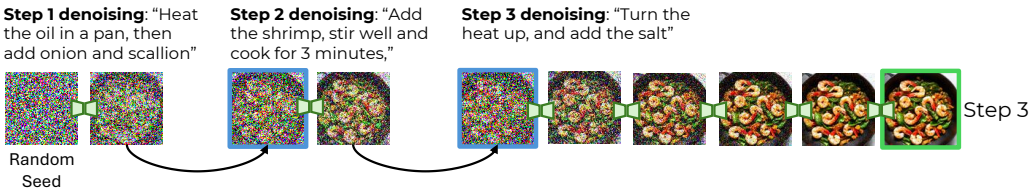
We propose to utilize beam search as a potential solution to this problem. Beam search is a well-established technique in text generation tasks like machine translation [16, 19, 41, 54], allowing the model to explore multiple possible paths during the generation process, evaluating different trajectories and refining its predictions. However, despite its success in other domains, it has been barely explored in this context, leaving its potential untapped. In full image sequence generation, beam search could help navigate the complex latent space, refining coherence by adjusting to multiple text prompts or adding semantic cues.

In this work, we explore beam search for latent prior selection in image sequence generation. While previous research has used latent representations as priors to improve sequence consistency [3, 37], it remains unclear which specific representations yield the best overall sequence. We build on these ideas by introducing beam search to explore the latent space, enabling iterative refinement of image sequences. As illustrated in Figure 1a, our approach leverages search in the latent space to identify the best sequence of images, minimizing the risk of falling into suboptimal or incoherent paths, while maintaining visual continuity across images. As the beam diffusion algorithm traverses the latent space, the number of hypothesis grow and the least probably search paths are pruned to keep a fixed number of candidate paths. To this end, we incorporate a cross-attention mechanism [21] that evaluates and scores candidate image paths based on their alignment with both the textual description of the next step and the visual context of previous steps. This provides a reliable quality measure for each path, allowing us to retain the best ones while pruning the least suitable, ensuring both adherence to instructions and visual continuity throughout the sequence.

We evaluated the effectiveness of BeamDiffusion in two domains: the Recipes and in the Visual Storytelling domains. Results obtained with human annotations, automatic evaluations and LLM-as-a-judge, all showed that the proposed approach outperforms baseline methods for generated sequence quality, showcasing its adaptability to diverse contexts.



(a) The beam denoising search tree.



(b) Example of one complete decoded denoising process using contextual captions.

Figure 1. The BeamDiffusion model works in the denoising latent space by denoising the initial seed to latents and them decoding different beam paths. This shows how our method evolves in the latent space and how it can explore the latent space.

2. Related Work

Decoding methods are crucial for enhancing the quality of large language models (LLMs). Techniques such as beam search, top-k and nucleus sampling are widely studied for their effectiveness in balancing computational cost with generating coherent, contextually appropriate outputs [9, 12, 44]. Beam search has also proven effective in tasks like machine translation [28], image captioning [17, 51], and visual storytelling [14].

Ensuring coherence remains a major challenge in multimodal sequence generation [5, 10, 18, 25]. Some approaches, like AR-LDM [31], encode the context of caption-image pairs into multimodal representations to maintain consistency across outputs. Likewise, Rahman et al. [36] incorporates the history of U-net [40, 43] latent vectors to preserve narrative coherence. Other approaches, such as Gill [20], combine LLMs with image encoder-decoder systems. StoryDiffusion [55], changes the U-Net's cross-attention mechanism enabling training-free consistency by sampling tokens from previous images to maintain coherence in newly generated frames. StoryGen [23] denoises the previous image and the resulting features serve as conditioning for the subsequent generation of images. GenHowTo [49] uses a control net during inference to project the previous image directly into the latent space to guide the diffusion process for the next image step. StackedDiffusion [29] processes stacked image sequences as a single latent representation allowing the attention mechanism to consider all steps together, enhancing coherence across steps. In [3], authors proposed to use latent information from previous steps to guide coherent generation of subsequent images relying on a greedy heuristic-based selection. CoSeD [37] improved on this idea and introduced a contrastive learning approach [4] to rank images generated from different latent seeds. However, both approaches focused on local refinements rather than optimizing throughout the sequence. Our beam search approach extends these local selection criteria to perform global optimization.

While effective for certain tasks, these models suffer from alignment issues, with retrieved or generated images failing to match the narrative context, and they too often require complex training. Our method addresses these previous constraints by generating images directly from the guiding text, maintaining consistency via a beam search for optimal seeds, similar to methods such as [45] that view image generation as a search process.

3. Generating Non-Linear Visual Sequences

Generating a full sequence of visually coherent images from textual scene descriptions is a challenging research problem. Given a sequence of steps $S = \{s_1, s_2, \ldots, s_L\}$, where each sequence step s_j is a textual description, our goal is to generate a corresponding sequence of images $\{I_1, I_2, \ldots, I_L\}$ that maintain consistency across all steps of the sequence. Diffusion Models generate high-quality images in a compressed latent space, but typically treat each image independently from a randomly initialized latent z_T.

Within this framework, Latent Diffusion Models (LDMs) [39] and Diffusion Transformers (DiTs) [32] share a common structure: both encode an input image I into a latent representation z using an encoder $\mathcal{E}$ and reconstruct it through a decoder $\mathcal{D}$ [39]. The reverse diffusion process is performed by a denoising backbone ϵ_θ, which can be a U-Net [40] (for LDMs) or a transformer [32] (for DiT), and incorporates cross-attention to condition the denoising iterations on an external input, e.g., the output embeddings of a text encoder $\tau_\theta(s)$. Both models are trained to iteratively estimate a denoised latent with the loss:

$$L_{\text{diff}} = \mathbb{E}_{\mathcal{E}(I),s,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(s))\|_2^2\right] \quad (1)$$

This formulation highlights that each denoising process is *agnostic to other generations*, making it challenging to maintain coherence across a sequence of related prompts. This is particularly important in non-linear narratives [46], where preserving character identity, lighting, and environmental details is crucial. To address these challenges, prior work [3, 23, 36, 37, 49] has proposed *reusing latent denoising iterations from earlier steps*, raising the key question: *how can we determine the optimal set of latent states that maximizes coherence across different generations?* Our approach, BeamDiffusion, explores this combinatorial search space to find an optimal solution, as we will discuss next.

3.1. Beam Diffusion Model

To address the limitations of standard diffusion models in generating coherent image sequences, we introduce the BeamDiffusion model. The proposed approach aims to enhance both the consistency and quality of image sequences by adopting a strategy inspired by beam search, but with a twist, it operates on latent denoising space. BeamDiffusion incorporates latent representations from previously generated images. These latent representations guide the generation of new candidates, ensuring that each new image aligns with the context of the evolving sequence. This approach allows visual elements from previous images to be carried over or referenced, maintaining contextual consistency and continuity, while also adapting to the next scene description.

In the remainder of this article, the term *step* will always refer to an element in the input sequence; and the term *iteration* will refer to one denoising iteration of the diffusion model.

Algorithm 1 BeamDiffusion Algorithm

1: **Input parameters:**
$\{s_1, \ldots, s_L\}$: sequence of steps; r: number of random seeds; m: steps back; w: beam width.
2: $\hat{B}_1 \leftarrow \{\text{SD}(s_1, rnd)\}^r$ ▷ Beam initialization with r different images.
3: **for** $j = 2$ **to** L **do**
4: **for all** beam $b \in \hat{B}_{j-1}$ **do**
5: $\mathcal{Z}_j \leftarrow \{\hat{B}_{j-n}, \ldots, \hat{B}_{j-1}\}$ ▷ Read denoising latents from previous steps.
6: **for all** latent $z \in \mathcal{Z}_j$ **do**
7: $I_j \leftarrow \text{SD}(s_j, z)$ ▷ Generate candidate a image.
8: $\hat{B}_{j-1} \leftarrow \hat{B}_{j-1} \bigcup I_j$ ▷ Extend beam with candidate image.
9: **end for**
10: **end for**
11: $\text{score}(I_j) \leftarrow \varphi(I_j \in \hat{B}_j)$ ▷ Score the new image of the beam.
12: $B_j \leftarrow \arg\max_{|B_j|=w} \text{score}(\hat{B}_j)$ ▷ Prune beams and keep top w beams.
13: **end for**
14: **return** best beam $B^* \leftarrow \arg\max_{B_L} \text{score}(B_L)$

3.2. Latent Space Exploration with Diffusion Beams

To generate coherent and diverse image sequences, we explore the latent space of diffusion models using a beam search-inspired strategy. For each sequence step j we generate a set of candidate images $\mathcal{I}_j = \{I_j^{(1)}, I_j^{(2)}, \ldots\}$, where each $I_j^{(k)}$ is conditioned on a different beam path.

Sampling the Denoising Latent Space. In our approach, the denoising process for each sequence step is conditioned by representations from prior steps, helping the model maintain coherence across the sequence. In the first step of the sequence, s_1, we generate multiple candidate images by running the diffusion process with r random noise seeds (Alg. 1, line 2). This strategy enables broader exploration of the latent space and reduces Bayes risk [1], as detailed in Appendix C.

For every generation process corresponding to a sequence step s_i, we store the first N latent space vectors $z_i = \{z_{i,t}\}_{t\in\subseteq\{1,\ldots,N\}}$ produced during the denoising iterations. On each step prompt $s_{j>1}$ we need to generate diverse samples that (a) explore different denoising trajectories in the latent space, and (b) are strongly correlated with previous samples. Leveraging the stored latent space vectors, we sample the latent subspace of n previous denoising processes. Hence, for steps $j > 1$, to enable contextual consistency, we

first accumulate representations from the previous m steps:

$$\mathcal{Z}_j = \bigcup_{i \in \{j-m,\ldots,j-1\}} \{z_i\}. \quad (2)$$

To generate new candidate images $\mathcal{I}_j = \{I_j^{(1)}, \ldots, I_j^{(k)}, \ldots, I_j^{(N \cdot m)}\}$ at step j, we condition the diffusion model on the latent trajectories in $\mathcal{Z}_j$. Specifically, for each latent trajectory $z \in \mathcal{Z}_j$, the model performs a new diffusion process to produce a corresponding image candidate $I_j^{(k)}$, Alg. 1, line 5 and 7. This step establishes a dependency between generations, where latents from earlier steps guide the synthesis of visually coherent transitions over time, as illustrated in Figure 1.

Diffusion Beams. Inspired by beam search [53], which maintains and extends the most promising candidates at each sequence step, we introduce *diffusion beams* to structure sequential image generation. At each step j, the set of candidate images $\mathcal{I}_j$ is used to compute the set of candidate diffusion beams $\hat{B}_j$: for each step j we consider all the beams existing in step $j-1$ and extend each one of such b beams $\hat{B}_{j-1}^{(b)}$ by pairing it with a new candidate image $I_j^{(k)} \in \mathcal{I}_j$. The resulting beams are defined as

$$\hat{B}_j = \bigcup_{(b,k)} \left\{ \hat{B}_{j-1}^{(b)} \oplus I_j^{(b \cdot k)} \mid I_j^{(b \cdot k)} \in \mathcal{I}_j \right\}, \quad (3)$$

where $\oplus$ denotes the concatenation of a prior beam with a new candidate image, and $(b \cdot k)$ indexes the pairing of beam b and candidate image $I_j^{(k)}$. This process builds a tree of candidate sequences, Alg. 1, line 8, enabling the model to explore diverse yet coherent visual narratives over time.

3.3. Pruning Diffusion Beam Paths

Tracking multiple *diffusion beams* enables BeamDiffusion to maintain visual coherence while exploring diverse continuations across a sequence. However, as the sequence length increases, the number of candidate beams grows rapidly, making it computationally infeasible to retain them all. To address this, we apply a pruning strategy based on a learned contrastive scoring function, which allows us to retain only the most promising beams while preserving diversity.

At each sequence step j, we construct a set of candidate beams $\hat{B}_j$. Each candidate beam $\hat{B}_j^{(b)}$ is formed by extending a beam from the previous step $\hat{B}_{j-1}^{(b)}$ with a newly generated image $I_j^{(k)}$. A complete candidate beam at step j is a sequence of images $\hat{B}_j^{(b)} = [I_1^{(b)}, I_2^{(b)}, \ldots, I_j^{(b)}]$, one per step. To evaluate the overall quality of a beam, we use a contrastive classifier [37], denoted by φ. Specifically, for each image $I_i^{(b)}$ in the beam, the classifier φ computes a compatibility score based on the image's prompt s_i and the previous images in the beam $B_{i-1}^{(b)}$, which includes earlier images and prompts in the same beam. The total beam score is then the sum of these per-image scores over all sequence steps up to j:

$$score(\hat{B}_j^{(b)}) = \sum_{i=1}^{j} \log \left(\frac{\exp\left(\varphi(I_i^{(b)}, s_i, B_{i-1}^{(b)})\right)}{\sum_{I_i' \in \mathcal{I}_i} \exp\left(\varphi(I_i', s_i, B_{i-1}^{(b)})\right)} \right), \quad (4)$$

where $\mathcal{I}_i$ is the set of all candidate images at sequence step i. The contrastive formulation ensures that each image is evaluated not in isolation, but relative to other candidates, capturing how well it fits the evolving sequence context.

Once scored, we prune the beam candidates by selecting the top w scoring beams:

$$B_j = \text{argmax}_{B_j \subseteq \hat{B}_j, |B_j| = w} \left(\sum_{i=1}^{w} score(\hat{B}_j^{(i)}) \right), \quad (5)$$

where B_j denotes the final set of retained beams for sequence step j.

To balance exploration and efficiency, pruning is disabled for the initial steps, allowing all candidates to be retained initially. After a few step, we apply pruning to keep only the top w beams. This strategy ensures wide exploration early on and focused, high-quality continuation in later sequence steps.

Training. The contrastive classifier $\varphi(I_j | \{s_1, \ldots, s_j\}, \{I_1, \ldots, I_{j-1}\})$ is trained as a *next image prediction* task to estimate how well align an image is with the previously sequence of steps and images. For a given sequence t and sequence step j, the ground-truth label $l_{t,j}$ indicates whether image I_j belongs to the sequence given its prompt s_j and context $B_{j-1}^{(b)}$. Negative examples are hallucinated from other sequences. The model minimizes the cross-entropy loss encouraging the model to promote candidates that best match the sequence's semantic and visual dynamics. The classifier is trained on the Recipes dataset [3]. See appendix G for implementation details.

4. Experimental Setup

In this section, we describe the experimental setup for evaluating BeamDiffusion's performance in generating coherent image sequences. We first outline the different decoding strategies that are considered. We then present the datasets and baseline methods used, and describe the evaluation protocols, which include human judgments, automatic metrics, and LLM-as-a-judge evaluations using Gemini 2.5 [6], all designed to assess semantic and visual consistency across generated sequences.

Decoding strategies. Our proposed method, BeamDiffusion, uses a beam search over latent sequences to generate coherent image sequences from textual prompts. To evaluate

the effectiveness of this decoding strategy, we also compare it against two alternative approaches applied to the same model: **Greedy / CoSeD**$_{len=1}$ [37], which selects at each step the top-scoring image according to CLIP similarity from the first few latents of the previous step, focusing on immediate best-match selection; and **Nucleus Sampling** [13], a probabilistic strategy that samples images from the subset whose cumulative CLIP probability exceeds a threshold p, encouraging diversity while still prioritizing high-probability outcomes.

All strategies are applied to the same sequences of prompts using the same underlying model, allowing a direct comparison of their ability to maintain sequence-level coherence.

Datasets. We used a dataset consisting of publicly available manual tasks in the Recipes domain [3], as well as the Visual Storytelling (VIST) dataset [15]. Recipes serve as the in-domain dataset, providing detailed, step-by-step instructions, while only a small portion of VIST tasks was used as out-of-domain data to train the classifier. Using both in-domain and out-of-domain data ensures that the classifier can generalize beyond the primary domain, allowing us to evaluate BeamDiffusion's performance on both familiar and novel types of tasks. Together, these datasets enable a thorough assessment of sequence coherence, semantic alignment, and visual consistency. More details can be found in Appendix E.

Baselines. To assess the performance of BeamDiffusion, we compared against several competitive baselines that aim to generate coherent image sequences from text: **GILL [20]**, **StackedDiffusion [29]**, **StoryDiffusion [55]**, and **One-Prompt-One-Story (1P1S) [24]**.

Human Evaluation. To find the optimal hyperparameters, the annotators compared pairs of beam search configurations to select the optimal beam width (w in Eq. 5) and steps back (m in Eq. 2). The winning configuration is compared against other configurations until a new configuration is selected as better. The selection criteria focused on semantic alignment and visual consistency. Semantic alignment assesses how well the image sequence matches the corresponding text, while visual consistency evaluates the coherence of the visuals across the sequence (e.g., keeping backgrounds, objects, or ingredients consistent).

Having chosen the optimal configuration, we compared BeamDiffusion to other methods. In the second annotation experiment, BeamDiffusion was compared with other baseline models. Annotators independently assessed the sequences, selecting the best method according to semantic and visual consistency with the possibility of choosing multiple options or none. For further information, see Appendix D.

Gemini Evaluation. To further validate our findings and annotate a wider range of tasks, we used Gemini, a large language model (LLM), as a judge to evaluate both configurations and baselines. Its selection process closely mirrored that of human annotators. For details, please refer to Appendix D.

Human-Gemini Agreement. We conducted an alignment assessment to measure the agreement between human annotators and Gemini. For this, human annotators were asked to rate 16 Recipes and 10 VIST tasks. These ratings were then compared to the evaluations made by Gemini. To quantify the agreement between the two, we computed Fleiss' kappa [8], obtaining a value of 70%, which indicates a substantial level of agreement. Based on this strong alignment, we extended the evaluation to the remaining sequences using Gemini.

Automatic Metrics. Following prior work [3, 23, 37], we evaluate each method using two complementary criteria: (1) *Visual Consistency*, measured via average image similarity using DINO [30] (DINO-I) and CLIP [35] (CLIP-I); and (2) *Semantic Alignment*, computed with CLIP (CLIP-T) by comparing each image to the average of all step texts up to and including that step. To ensure comparability across metrics, we normalize CLIP-I, CLIP-T, and DINO-I by their respective maxima. We further define CLIP* as the product of CLIP-I and CLIP-T, and DINO* as the product of DINO-I and CLIP-T, balancing visual and semantic quality while emphasizing alignment with the stepwise input text. To better capture real performance, the results for DINO-I, CLIP-I, CLIP-T, CLIP*, and DINO* are reported as *win percentages* rather than mean values. This avoids overestimating methods that perform very well on some sequences but poorly on others, which could otherwise produce artificially high averages despite inconsistent quality.

In addition, we adopt metrics from StackedDiffusion [29] to evaluate task-specific aspects of the generated sequences: *Goal Faithfulness*, measuring alignment of the final image with the overall textual goal; *Step Faithfulness*, measuring alignment of each intermediate image with its corresponding step text; and *Cross-Image Consistency*, capturing visual coherence across images in the sequence. These metrics complement the previous ones by explicitly assessing semantic fidelity and sequence-level consistency. See Appendix D for further details.

5. Results and Discussion

In this section, we evaluate the effectiveness of BeamDiffusion through three complementary approaches: human evaluations, LLM based judgments, and automatic metrics. We begin by comparing decoding strategies, selecting the

method that achieves the highest human preference scores for subsequent experiments. We then highlight key performance aspects of BeamDiffusion, including visual and semantic consistency, and conclude with a qualitative analysis. Additional details on the annotation setup are provided in Appendix D.

5.1. Comparison of Decoding Methods

To evaluate the effectiveness of different decoding strategies, we compare our proposed method, BeamDiffusion, against Greedy/CoSeD$_{len=1}$ [37] and Nucleus Sampling [13] using the same sequences of prompts. We perform experiments with two underlying diffusion models (backbones), FLUX.1-dev [2] and SD2.1 [39], to test the generality of our approach.

Table 1. Human (H) and Gemini (G) evaluation comparison of decoding strategies across two backbones.

	FLUX.1-dev		SD2.1	
Method	**H (%)**	**G (%)**	**H (%)**	**G (%)**
Greedy/CoSeD$_{len=1}$ [37]	16.35	10.00	11.11	10.00
Nucleus Sampling [13]	24.04	16.67	33.33	26.67
BeamDiffusion	**59.61**	**73.33**	**55.56**	**63.33**

As shown in Table 1, BeamDiffusion consistently achieves the highest preference scores across both backbones, according to both human annotators (H) and Gemini (G). In particular, BeamDiffusion substantially outperforms Greedy decoding and Nucleus Sampling, indicating that beam-based decoding produces sequences that are not only more coherent but also better aligned with human and automatic judgments. While Nucleus Sampling improves over Greedy decoding by encouraging diversity, its performance remains notably below that of BeamDiffusion in both evaluation settings. The agreement between H and G further strengthens the evidence for the robustness of our decoding strategy. Given these results, we adopt BeamDiffusion as our default decoding method in subsequent experiments (see Appendix B.4 for qualitative and automatic evaluation results).

5.2. General Results

Table 2 compares methods across Recipes and VIST. BeamDiffusion dominates in both domains, leading Recipes with 64.52% (H) and 73.33% (G), and VIST with 41.10% (H) and 48.28% (G). StoryDiffusion consistently ranks second, with 22.58% (H) and 20.00% (G) in Recipes, and 36.99% (H) and 41.38% (G) in VIST. In contrast, 1P1S receives moderate preference (12.90% H in Recipes, 19.18% H in VIST), while GILL and StackedDiffusion record the lowest selection rates.

These results show that BeamDiffusion achieves the strongest performance across settings, with especially large margins in Recipes where BeamDiffusion surpasses the next-best method by over 40% for human preference. The consistently low selection rates of GILL and StackedDiffusion highlight that these methods are rarely preferred by either humans or Gemini, in contrast to the stronger performance of BeamDiffusion and StoryDiffusion. Overall, the findings highlight that effective search and selection strategies are central to producing coherent and engaging narratives, enabling models to maintain quality even in complex domains such as VIST.

Table 2. Frequency of method selection by human annotators (H) and Gemini (G).

	Recipes		VIST	
Method	**H (%)**	**G (%)**	**H (%)**	**G (%)**
GILL [20]	0.00	0.00	2.74	0.00
StackedDiffusion [29]	0.00	3.33	0.00	0.00
StoryDiffusion [55]	22.58	20.00	36.99	41.38
1P1S [24]	12.90	3.33	19.18	10.34
BeamDiffusion	**64.52**	**73.33**	**41.10**	**48.28**

5.3. Automatic Metrics

Table 6 compares all methods using metrics that measure sequence quality, including CLIP-I, DINO-I, CLIP-T, CLIP*, DINO*, Goal Faithfulness, Step Faithfulness, and Cross-Image Consistency.

BeamDiffusion achieves the highest combined scores (CLIP*: 40.00%, DINO*: 30.00%), reflecting strong performance in both visual and semantic consistency. This is confirmed by task-specific metrics, where BeamDiffusion leads in Goal Faithfulness (0.82), Step Faithfulness (0.92), and Cross-Image Consistency (0.61).

StoryDiffusion performs well on CLIP-T (35.00%) and achieves high task-specific scores (Goal Faithfulness: 0.78, Step Faithfulness: 0.87, Cross-Image Consistency: 0.61), showing strong semantic alignment and visual coherence despite lower combined metrics. 1P1S achieves the highest DINO-I (30.00%) but lower CLIP-T (10.00%) and moderate task-specific metrics, indicating strong visual consistency but weaker semantic alignment. StackedDiffusion maintains balanced performance across CLIP-I, CLIP-T, and DINO-I (around 25.00%) with moderate task-specific scores, while GILL consistently underperforms across both automatic and task-specific metrics.

Overall, the relationships between CLIP-I/DINO-I and Cross-Image Consistency, and between CLIP-T and Goal/Step Faithfulness, show that the combined metrics CLIP* and DINO* reliably reflect both semantic and visual consistency. Together, these results demonstrate that BeamDiffusion is the most effective method for generating coherent and semantically faithful sequences of images.

Table 3. Comparison of different methods across embedding-based sequence quality metrics. CLIP-I, DINO-I, CLIP-T, CLIP*, and DINO* are reported as win percentages, while Goal Faithfulness, Step Faithfulness, and Cross-Image Consistency are reported as absolute scores.

Method	CLIP-I	DINO-I	CLIP-T	CLIP*	DINO*	Goal Faith.↑	Step Faith.↑	Cross-Image Cons.↑
GILL [20]	0.00	0.00	0.00	0.00	0.00	0.67	0.52	0.54
StackedDiffusion [29]	25.00	25.00	5.00	25.00	25.00	0.60	0.67	0.57
StoryDiffusion [55]	20.00	25.00	35.00	25.00	20.00	0.78	0.87	**0.61**
1P1S [24]	25.00	**30.00**	10.00	10.00	25.00	0.75	0.83	0.57
BeamDiffusion	**30.00**	20.00	**50.00**	**40.00**	**30.00**	**0.82**	**0.92**	**0.61**

5.4. Best Beam Search Configuration

The hyperparameters beam width (w in Eq. 5) and steps back (m in Eq. 2), play a crucial role in the performance of BeamDiffusion. In this section, we report an exaustive evaluation on how different beam widths (2, 3, 4, and 6) and steps back (2, 3, and 4) improve the latent selection and the overall quality of the generation. A more detailed discussion on the impact of the latents selection is provided in Appendix B.

We first perform human annotations on a set of sequences to identify the optimal beam search configuration, see Appendix D for details. To determine the best configuration, annotations were used to discard non-optimal configurations. The results in Table 4 showed that the best performing configuration was 2 steps back with a beam width of 4, achieving a selection rate of 25%. Other configurations, such as 3 steps back with a beam width of 2 and 4 steps back with a beam width of 2, also showed a strong performance (16.67%), but none outperformed the 2 steps back and beam width of 4 configuration. The configurations with beam width 6 performed the worst in all the settings tested.

To assess the reliability of these findings, we used Gemini for the same evaluation on six tasks, achieving an 85% Fleiss' Kappa agreement [8], indicating near-perfect alignment. The results from the extended Gemini evaluation (Table 5) confirmed that the optimal configuration, 2 steps back with a beam width of 4, remained the best, now achieving a 48.57% selection rate, a significant improvement over the 33.33% from the initial six-task evaluation. Further, the configuration with 3 steps back and a beam width of 2 still performed well, yielding a 28.57% selection rate, confirming its robustness as an alternative. However, larger beam widths, such as beam width 6, continued to underperform, reinforcing that, in this task, a higher beam width does not always improve performance.

5.5. Non-Linear Sequence Denoising

To analyze how BeamDiffusion explores the latent space, we measured the number of times that a sequence step provides latents for the generation of subsequent steps of the sequence. In this experiment we consider sequences of four steps, (Figure 2). Appendix B.2 provides an analysis at the level of the denoising iterations. Latents from step 1 were chosen 69.9% of

the times they had a chance of being selected, highlighting the role of step 1 in shaping the sequence. Selection rates decrease for later steps, with Step 3 at 46.8% and Step 2 at 33.3%. Earlier steps, such as Step 1, have more chances of being chosen, leading to a higher selection rate. Thus, the percentages reflect how often each step is selected given the different possibilities for their inclusion, rather than a simple additive total. This pattern demonstrates how beam search prioritizes crucial steps, particularly Step 1, to establish sequence coherence while strategically weighing intermediate sequence steps.

5.6. Qualitative Analysis

To highlight the impact of our approach on semantic and visual consistency, we present two examples, one from the Recipes domain (Figure 3a) and one from VIST (Figure 3b). In the recipe example, BeamDiffusion avoids major semantic errors and maintains consistent object shapes across steps; for instance, while StoryDiffusion and 1P1S show inconsistencies in the shape of the tins between steps 3 and 4, BeamDiffusion preserves them accurately. In the VIST example, BeamDiffusion maintains both semantic alignment and visual coherence throughout the sequence, accurately depicting two friends and preserving consistent colors and lighting across all steps. In contrast, baselines such as 1P1S introduce a temporal inconsistency, shifting from sunset to night and back to sunset, while StoryDiffusion misinterprets the step text by showing three individuals instead of two friends.

6. Limitations

Exploring the latent denoising space with BeamDiffusion requires Diffusion Models that provide access to fine-grain latents across denoising iterations, where BeamDiffusion still has the opportunity to steer the reverse diffusion process.

The theoretical complexity of BeamDiffusion is linear with the number of beams that we configure the model to explore. The drawback of this strategy is that exploring the space of hypothesis increases the computational complexity of the process, discussed in Appendix F. However, this

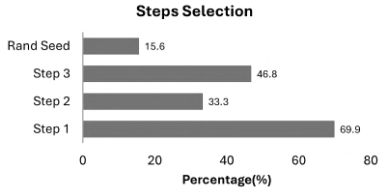
Table 4. Influence of different hyperparameter values according to human annotations.

Steps Back	Beam Width			
	2	3	4	6
2	8.33	8.33	**25.00**	8.33
3	16.67	-	-	-
4	16.67	8.33	8.33	-

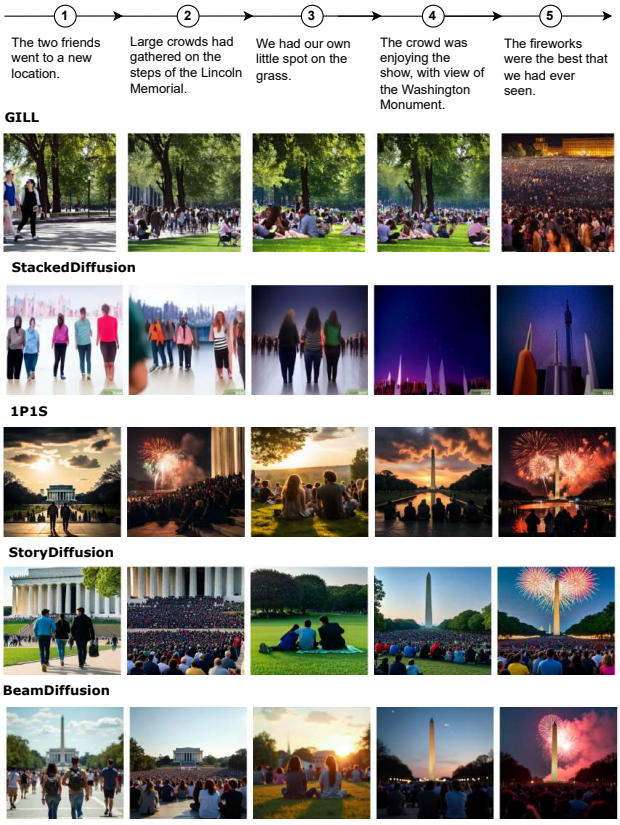
Table 5. Influence of different hyperparameter values according to Gemini annotations.

Steps Back	Beam Width			
	2	3	4	6
2	5.71	2.86	**48.57**	5.71
3	28.57	-	2.86	-
4	-	-	2.86	-

Figure 2. Non-linear steps selection, according to selection opportunities.



(a) Recipes domain task.



(b) VIST story.

Figure 3. Example tasks from two different domains: Recipes (*"Crema Catalana with salted caramel sauce"*) and VIST (*"Fireworks - 4th July in Washington, DC"*).

provides a solid ground to explore exciting new ideas where stronger constraints can be enforced in the explored beams.

7. Conclusions

In this paper, we proposed a novel method to explore the latent diffusion space with a beam search strategy, to generate image sequences. BeamDiffusion is a theoretically sound framework, drawing on the strengths of text-based decoding techniques [44], with clear experimental gains over alternative methods, as we demonstrated in generating image sequences with enhanced coherence and consistency.

Through both human and automatic evaluation, we show that beam search performs effectively in generating high-quality and coherent image sequences. The insights drawn from these evaluations reveal that beam search aligns closely with textual instructions while ensuring visual continuity. Most methods can align images with text, however, we demonstrated that *sequence decoding methods perform naturally better at aligning their output with the input sequence*, and BeamDiffusion in particular achieves an optimal balance between the sequence of text instructions and the visual

coherence of the generated images. Furthermore, our results underscore the significance of the non-linear selection process in beam search, highlighting how prioritizing certain steps and latents can optimize the quality of the generation process.

This work shows that the latent diffusion space can be steered, offering more reliable and factual outputs. As future work, we plan to explore constraint decoding method that assess the likelihood of each denoise latent with respect to the target prompt. Solving this challenge, will improve the correlation between all elements in a sequence of scenes.

References

[1] Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew R. Gormley. It's MBR all the way down: Modern generation techniques through the lens of minimum bayes risk. *CoRR*, abs/2310.01387, 2023. 3

[2] black-forest labs. flux: Official inference repo for flux.1 models. https://github.com/black-forest-labs/flux, 2024. GitHub repository. 6, 26

[3] João Bordalo, Vasco Ramos, Rodrigo Valerio, Diogo Glória-Silva, Yonatan Bitton, Michal Yarom, Idan Szpektor, and João Magalhães. Generating coherent sequences of visual illustrations for real-world manual tasks. pages 12777–12797, 2024. 1, 2, 3, 4, 5, 20

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[5] Weifeng Chen, Jiacheng Zhang, Jie Wu, Hefeng Wu, Xuefeng Xiao, and Liang Lin. Id-aligner: Enhancing identity-preserving text-to-image generation with reward feedback learning. *arXiv preprint arXiv:2404.15449*, 2024. 2

[6] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, Krishna Haridasan, Ahmed Omran, Nikunj Saunshi, Dara Bahri, Gaurav Mishra, Eric Chu, Toby Boyd, Brad Hekman, Aaron Parisi, Chaoyi Zhang, Kornraphop Kawintiranon, Tania Bedrax-Weiss, Oliver Wang, Ya Xu, Ollie Purkiss, Uri Mendlovic, Ilaï Deutel, Nam Nguyen, Adam Langley, Flip Korn, Lucia Rossazza, Alexandre Ramé, Sagar Waghmare, Helen Miller, Nathan Byrd, Ashrith Sheshan, Raia Hadsell Sangnie Bhardwaj, Pawel Janus, Tero Rissa, Dan Horgan, Sharon Silver, Ayzaan Wahid, Sergey Brin, Yves Raimond, Klemen Kloboves, Cindy Wang, Nitesh Bharadwaj Gundavarapu, Ilia Shumailov, Bo Wang, Mantas Pajarskas, Joe Heyward, Martin Nikoltchev, Maciej Kula, Hao Zhou, Zachary Garrett, Sushant Kafle, Sercan Arik, Ankita Goel, Mingyao Yang, Jiho Park, Koji Kojima, Parsa Mahmoudieh, Koray Kavukcuoglu, Grace Chen, Doug Fritz, Anton Bulyenov, Sudeshna Roy, Dimitris Paparas, Hadar Shemtov, Bo-Juen Chen, Robin Strudel, David Reitter, Aurko Roy, Andrey Vlasov, Changwan Ryu, Chas Leichner, Haichuan Yang, Zelda Mariet, Denis Vnukov, Tim Sohn, Amy Stuart, Wei Liang, Minmin Chen, Praynaa Rawlani, Christy Koh, JD Co-Reyes, Guangda Lai, Praseem Banzal, Dimitrios Vytiniotis, Jieru Mei, Mu Cai, Mohammed Badawi, Corey Fry, Ale Hartman, Daniel Zheng, Eric Jia, James Keeling, Annie Louis, Ying Chen, Efren Robles, Wei-Chih Hung, Howard Zhou, Nikita Saxena, Sonam Goenka, Olivia Ma, Zach Fisher, Mor Hazan Taege, Emily Graves, David Steiner, Yujia Li, Sarah Nguyen, Rahul Sukthankar, Joe Stanton, Ali Eslami, Gloria Shen, Berkin Akin, Alexey Guseynov, Yiqian Zhou, Jean-Baptiste Alayrac, Armand Joulin, Efrat Farkash, Ashish Thapliyal, Stephen Roller, Noam Shazeer, Todor Davchev, Terry Koo, Hannah Forbes-Pollard, Kartik Audhkhasi, Greg Farquhar, Adi Mayrav Gilady, Maggie Song, John Aslanides, Piermaria Mendolicchio, Alicia Parrish, John Blitzer, Pramod Gupta, Xiaoen Ju, Xiaochen Yang, Puranjay Datta, Andrea Tacchetti, Sanket Vaibhav Mehta, Gregory Dibb, Shubham Gupta, Federico Piccinini, Raia Hadsell, Sujee Rajayogam, Jiepu Jiang, Patrick Griffin, Patrik Sundberg, Jamie Hayes, Alexey Frolov, Tian Xie, Adam Zhang, Kingshuk Dasgupta, Uday Kalra, Lior Shani, Klaus Macherey, Tzu-Kuo Huang, Liam MacDermed, Karthik Duddu, Paulo Zacchello, Zi Yang, Jessica Lo, Kai Hui, Matej Kastelic, Derek Gasaway, Qijun Tan, Summer Yue, Pablo Barrio, John Wieting, Weel Yang, Andrew Nystrom, Solomon Demmessie, Anselm Levskaya, Fabio Viola, Chetan Tekur, Greg Billock, George Necula, Mandar Joshi, Rylan Schaeffer, Swachhand Lokhande, Christina Sorokin, Pradeep Shenoy, Mia Chen, Mark Collier, Hongji Li, Taylor Bos, Nevan Wichers, Sun Jae Lee, Angéline Pouget, Santhosh Thangaraj, Kyriakos Axiotis, Phil Crone, Rachel Sterneck, Nikolai Chinaev, Victoria Krakovna, Oleksandr Ferludin, Ian Gemp, Stephanie Winkler, Dan Goldberg, Ivan Korotkov, Kefan Xiao, Malika Mehrotra, Sandeep Mariserla, Vihari Piratla, Terry Thurk, Khiem Pham, Hongxu Ma, Alexandre Senges, Ravi Kumar, Clemens Meyer, Ellie Talius, Nuo Wang Pierse, Ballie Sandhu, Horia Toma, Kuo Lin, Swaroop Nath, Tom Stone, Dorsa Sadigh, Nikita Gupta, Arthur Guez, Avi Singh, Matt Thomas, Tom Duerig, Yuan Gong, Richard Tanburn, Lydia Lihui Zhang, Phuong Dao, Mohamed Hammad, Sirui Xie, Shruti Rijhwani, Ben Murdoch, Duhyeon Kim, Will Thompson, Heng-Tze Cheng, Daniel Sohn, Pablo Sprechmann, Qiantong Xu, Srinivas Tadepalli, Peter Young, Ye Zhang, Hansa Srinivasan, Miranda Aperghis, Aditya Ayyar, Hen Fitoussi, Ryan Burnell, David Madras, Mike Dusenberry, Xi Xiong, Tayo Oguntebi, Ben Albrecht, Jörg Bornschein, Jovana Mitrović, Mason Dimarco, Bhargav Kanagal Shamanna, Premal Shah, Eren Sezener, Shyam Upadhyay, Dave Lacey, Craig Schiff, Sebastien Baur, Sanjay Ganapathy, Eva Schnider, Mateo Wirth, Connor Schenck, Andrey Simanovsky, Yi-Xuan Tan, Philipp Fränken, Dennis Duan, Bharath Mankalale, Nikhil Dhawan, Kevin Sequeira, Zichuan Wei, Shivanker Goel, Caglar Unlu, Yukun Zhu, Haitian Sun, Ananth Balashankar, Kurt Shuster, Megh Umekar, Mahmoud Alnahlawi, Aäron van den Oord, Kelly Chen, Yuexiang Zhai, Zihang Dai, Kuang-Huei Lee, Eric Doi, Lukas Zilka, Rohith Vallu, Disha Shrivastava, Jason Lee, Hisham Husain, Honglei Zhuang, Vincent Cohen-Addad, Jarred Barber, James Atwood, Adam Sadovsky, Quentin

Wellens, Steven Hand, Arunkumar Rajendran, Aybuke Turker, CJ Carey, Yuanzhong Xu, Hagen Soltau, Zefei Li, Xinying Song, Conglong Li, Iurii Kemaev, Sasha Brown, Andrea Burns, Viorica Patraucean, Piotr Stanczyk, Renga Aravamudhan, Mathieu Blondel, Hila Noga, Lorenzo Blanco, Will Song, Michael Isard, Mandar Sharma, Reid Hayes, Dalia El Badawy, Avery Lamp, Itay Laish, Olga Kozlova, Kelvin Chan, Sahil Singla, Srinivas Sunkara, Mayank Upadhyay, Chang Liu, Aijun Bai, Jarek Wilkiewicz, Martin Zlocha, Jeremiah Liu, Zhuowan Li, Haiguang Li, Omer Barak, Ganna Raboshchuk, Jiho Choi, Fangyu Liu, Erik Jue, Mohit Sharma, Andreea Marzoca, Robert Busa-Fekete, Anna Korsun, Andre Elisseeff, Zhe Shen, Sara Mc Carthy, Kay Lamerigts, Anahita Hosseini, Hanzhao Lin, Charlie Chen, Fan Yang, Kushal Chauhan, Mark Omernick, Dawei Jia, Karina Zainullina, Demis Hassabis, Danny Vainstein, Ehsan Amid, Xiang Zhou, Ronny Votel, Eszter Vértes, Xinjian Li, Zongwei Zhou, Angeliki Lazaridou, Brendan McMahan, Arjun Narayanan, Hubert Soyer, Sujoy Basu, Kayi Lee, Bryan Perozzi, Qin Cao, Leonard Berrada, Rahul Arya, Ke Chen, Katrina, Xu, Matthias Lochbrunner, Alex Hofer, Sahand Sharifzadeh, Renjie Wu, Sally Goldman, Pranjal Awasthi, Xuezhi Wang, Yan Wu, Claire Sha, Biao Zhang, Maciej Mikuła, Filippo Graziano, Siobhan Mcloughlin, Irene Giannoumis, Youhei Namiki, Chase Malik, Carey Radebaugh, Jamie Hall, Ramiro Leal-Cavazos, Jianmin Chen, Vikas Sindhwani, David Kao, David Greene, Jordan Griffith, Chris Welty, Ceslee Montgomery, Toshihiro Yoshino, Liangzhe Yuan, Noah Goodman, Assaf Hurwitz Michaely, Kevin Lee, KP Sawhney, Wei Chen, Zheng Zheng, Megan Shum, Nikolay Savinov, Etienne Pot, Alex Pak, Morteza Zadimoghaddam, Sijal Bhatnagar, Yoad Lewenberg, Blair Kutzman, Ji Liu, Lesley Katzen, Jeremy Selier, Josip Djolonga, Dmitry Lepikhin, Kelvin Xu, Jacky Liang, Jiewen Tan, Benoit Schillings, Muge Ersoy, Pete Blois, Bernd Bandemer, Abhimanyu Singh, Sergei Lebedev, Pankaj Joshi, Adam R. Brown, Evan Palmer, Shreya Pathak, Komal Jalan, Fedir Zubach, Shuba Lall, Randall Parker, Alok Gunjan, Sergey Rogulenko, Sumit Sanghai, Zhaoqi Leng, Zoltan Egyed, Shixin Li, Maria Ivanova, Kostas Andriopoulos, Jin Xie, Elan Rosenfeld, Auriel Wright, Ankur Sharma, Xinyang Geng, Yicheng Wang, Sam Kwei, Renke Pan, Yujing Zhang, Gabby Wang, Xi Liu, Chak Yeung, Elizabeth Cole, Aviv Rosenberg, Zhen Yang, Phil Chen, George Polovets, Pranav Nair, Rohun Saxena, Josh Smith, Shuo yiin Chang, Aroma Mahendru, Svetlana Grant, Anand Iyer, Irene Cai, Jed McGiffin, Jiaming Shen, Alanna Walton, Antonious Girgis, Oliver Woodman, Rosemary Ke, Mike Kwong, Louis Rouillard, Jinmeng Rao, Zhihao Li, Yuntao Xu, Flavien Prost, Chi Zou, Ziwei Ji, Alberto Magni, Tyler Liechty, Dan A. Calian, Deepak Ramachandran, Igor Krivokon, Hui Huang, Terry Chen, Anja Hauth, Anastasija Ilić, Weijuan Xi, Hyeontaek Lim, Vlad-Doru Ion, Pooya Moradi, Metin Toksoz-Exley, Kalesha Bullard, Miltos Allamanis, Xiaomeng Yang, Sophie Wang, Zhi Hong, Anita Gergely, Cheng Li, Bhavishya Mittal, Vitaly Kovalev, Victor Ungureanu, Jane Labanowski, Jan Wassenberg, Nicolas Lacasse, Geoffrey Cideron, Petar Dević, Annie Marsden, Lynn Nguyen, Michael Fink, Yin Zhong, Tatsuya Kiyono, Desi Ivanov, Sally Ma, Max Bain, Kiran Yalasangi, Jennifer She, Anastasia Petrushkina, Mayank Lunayach, Carla Bromberg, Sarah Hodkinson, Vilobh Meshram, Daniel Vlasic, Austin Kyker, Steve Xu, Jeff Stanway, Zuguang Yang, Kai Zhao, Matthew Tung, Seth Odoom, Yasuhisa Fujii, Justin Gilmer, Eunyoung Kim, Felix Halim, Quoc Le, Bernd Bohnet, Seliem El-Sayed, Behnam Neyshabur, Malcolm Reynolds, Dean Reich, Yang Xu, Erica Moreira, Anuj Sharma, Zeyu Liu, Mohammad Javad Hosseini, Naina Raisinghani, Yi Su, Ni Lao, Daniel Formoso, Marco Gelmi, Almog Gueta, Tapomay Dey, Elena Gribovskaya, Domagoj Ćevid, Sidharth Mudgal, Garrett Bingham, Jianling Wang, Anurag Kumar, Alex Cullum, Feng Han, Konstantinos Bousmalis, Diego Cedillo, Grace Chu, Vladimir Magay, Paul Michel, Ester Hlavnova, Daniele Calandriello, Setareh Ariafar, Kaisheng Yao, Vikash Sehwag, Arpi Vezer, Agustin Dal Lago, Zhenkai Zhu, Paul Kishan Rubenstein, Allen Porter, Anirudh Baddepudi, Oriana Riva, Mihai Dorin Istin, Chih-Kuan Yeh, Zhi Li, Andrew Howard, Nilpa Jha, Jeremy Chen, Raoul de Liedekerke, Zafarali Ahmed, Mikel Rodriguez, Tanuj Bhatia, Bangju Wang, Ali Elqursh, David Klinghoffer, Peter Chen, Pushmeet Kohli, Te I, Weiyang Zhang, Zack Nado, Jilin Chen, Maxwell Chen, George Zhang, Aayush Singh, Adam Hillier, Federico Lebron, Yiqing Tao, Ting Liu, Gabriel Dulac-Arnold, Jingwei Zhang, Shashi Narayan, Buhuang Liu, Orhan Firat, Abhishek Bhowmick, Bingyuan Liu, Hao Zhang, Zizhao Zhang, Georges Rotival, Nathan Howard, Anu Sinha, Alexander Grushetsky, Benjamin Beyret, Keerthana Gopalakrishnan, James Zhao, Kyle He, Szabolcs Payrits, Zaid Nabulsi, Zhaoyi Zhang, Weijie Chen, Edward Lee, Nova Fallen, Sreenivas Gollapudi, Aurick Zhou, Filip Pavetić, Thomas Köppe, Shiyu Huang, Rama Pasumarthi, Nick Fernando, Felix Fischer, Daria Ćurko, Yang Gao, James Svensson, Austin Stone, Haroon Qureshi, Abhishek Sinha, Apoorv Kulshreshtha, Martin Matysiak, Jieming Mao, Carl Saroufim, Aleksandra Faust, Qingnan Duan, Gil Fidel, Kaan Katircioglu, Raphaël Lopez Kaufman, Dhruv Shah, Weize Kong, Abhishek Bapna, Gellért Weisz, Emma Dunleavy, Praneet Dutta, Tianqi Liu, Rahma Chaabouni, Carolina Parada, Marcus Wu, Alexandra Belias, Alessandro Bissacco, Stanislav Fort, Li Xiao, Fantine Huot, Chris Knutsen, Yochai Blau, Gang Li, Jennifer Prendki, Juliette Love, Yinlam Chow, Pichi Charoenpanit, Hidetoshi Shimokawa, Vincent Coriou, Karol Gregor, Tomas Izo, Arjun Akula, Mario Pinto, Chris Hahn, Dominik Paulus, Jiaxian Guo, Neha Sharma, Cho-Jui Hsieh, Adaeze Chukwuka, Kazuma Hashimoto, Nathalie Rauschmayr, Ling Wu, Christof Angermueller, Yulong Wang, Sebastian Gerlach, Michael Pliskin, Daniil Mirylenka, Min Ma, Lexi Baugher, Bryan Gale, Shaan Bijwadia, Nemanja Rakićević, David Wood, Jane Park, Chung-Ching Chang, Babi Seal, Chris Tar, Kacper Krasowiak, Yiwen Song, Georgi Stephanov, Gary Wang, Marcello Maggioni, Stein Xudong Lin, Felix Wu, Shachi Paul, Zixuan Jiang, Shubham Agrawal, Bilal Piot, Alex Feng, Cheolmin Kim, Tulsee Doshi, Jonathan Lai, Chuqiao, Xu, Sharad Vikram, Ciprian Chelba, Sebastian Krause, Vincent Zhuang, Jack Rae, Timo Denk, Adrian Collister, Lotte Weerts, Xianghong Luo, Yifeng Lu, Håvard Garnes, Nitish Gupta, Terry Spitz, Avinatan Hassidim, Lihao Liang, Izhak Shafran, Peter Humphreys, Kenny Vassigh, Phil Wal-

lis, Virat Shejwalkar, Nicolas Perez-Nieves, Rachel Hornung, Melissa Tan, Beka Westberg, Andy Ly, Richard Zhang, Brian Farris, Jongbin Park, Alec Kosik, Zeynep Cankara, Andrii Maksai, Yunhan Xu, Albin Cassirer, Sergi Caelles, Abbas Abdolmaleki, Mencher Chiang, Alex Fabrikant, Shravya Shetty, Luheng He, Mai Giménez, Hadi Hashemi, Sheena Panthaplackel, Yana Kulizhskaya, Salil Deshmukh, Daniele Pighin, Robin Alazard, Disha Jindal, Seb Noury, Pradeep Kumar S, Siyang Qin, Xerxes Dotiwalla, Stephen Spencer, Mohammad Babaeizadeh, Blake JianHang Chen, Vaibhav Mehta, Jennie Lees, Andrew Leach, Penporn Koanantakool, Ilia Akolzin, Ramona Comanescu, Junwhan Ahn, Alexey Svyatkovskiy, Basil Mustafa, David D'Ambrosio, Shiva Mohan Reddy Garlapati, Pascal Lamblin, Alekh Agarwal, Shuang Song, Pier Giuseppe Sessa, Pauline Coquinot, John Maggs, Hussain Masoom, Divya Pitta, Yaqing Wang, Patrick Morris-Suzuki, Billy Porter, Johnson Jia, Jeffrey Dudek, Raghavender R, Cosmin Paduraru, Alan Ansell, Tolga Bolukbasi, Tony Lu, Ramya Ganeshan, Zi Wang, Henry Griffiths, Rodrigo Benenson, Yifan He, James Swirhun, George Papamakarios, Aditya Chawla, Kuntal Sengupta, Yan Wang, Vedrana Milutinovic, Igor Mordatch, Zhipeng Jia, Jamie Smith, Will Ng, Shitij Nigam, Matt Young, Eugen Vušak, Blake Hechtman, Sheela Goenka, Avital Zipori, Kareem Ayoub, Ashok Popat, Trilok Acharya, Luo Yu, Dawn Bloxwich, Hugo Song, Paul Roit, Haiqiong Li, Aviel Boag, Nigamaa Nayakanti, Bilva Chandra, Tianli Ding, Aahil Mehta, Cath Hope, Jiageng Zhang, Idan Heimlich Shtacher, Kartikeya Badola, Ryo Nakashima, Andrei Sozanschi, Iulia Comşa, Ante Žužul, Emily Caveness, Julian Odell, Matthew Watson, Dario de Cesare, Phillip Lippe, Derek Lockhart, Siddharth Verma, Huizhong Chen, Sean Sun, Lin Zhuo, Aditya Shah, Prakhar Gupta, Alex Muzio, Ning Niu, Amir Zait, Abhinav Singh, Meenu Gaba, Fan Ye, Prajit Ramachandran, Mohammad Saleh, Raluca Ada Popa, Ayush Dubey, Frederick Liu, Sara Javanmardi, Mark Epstein, Ross Hemsley, Richard Green, Nishant Ranka, Eden Cohen, Chuyuan Kelly Fu, Sanjay Ghemawat, Jed Borovik, James Martens, Anthony Chen, Pranav Shyam, André Susano Pinto, Ming-Hsuan Yang, Alexandru Ţifrea, David Du, Boqing Gong, Ayushi Agarwal, Seungyeon Kim, Christian Frank, Saloni Shah, Xiaodan Song, Zhiwei Deng, Ales Mikhalap, Kleopatra Chatziprimou, Timothy Chung, Toni Creswell, Susan Zhang, Yennie Jun, Carl Lebsack, Will Truong, Slavica Andačić, Itay Yona, Marco Fornoni, Rong Rong, Serge Toropov, Afzal Shama Soudagar, Andrew Audibert, Salah Zaiem, Zaheer Abbas, Andrei Rusu, Sahitya Potluri, Shitao Weng, Anastasios Kementsietsidis, Anton Tsitsulin, Daiyi Peng, Natalie Ha, Sanil Jain, Tejasi Latkar, Simeon Ivanov, Cory McLean, Anirudh GP, Rajesh Venkataraman, Canoee Liu, Dilip Krishnan, Joel D'sa, Roey Yogev, Paul Collins, Benjamin Lee, Lewis Ho, Carl Doersch, Gal Yona, Shawn Gao, Felipe Tiengo Ferreira, Adnan Ozturel, Hannah Muckenhirn, Ce Zheng, Gargi Balasubramaniam, Mudit Bansal, George van den Driessche, Sivan Eiger, Salem Haykal, Vedant Misra, Abhimanyu Goyal, Danilo Martins, Gary Leung, Jonas Valfridsson, Four Flynn, Will Bishop, Chenxi Pang, Yoni Halpern, Honglin Yu, Lawrence Moore, Yuvein, Zhu, Sridhar Thiagarajan, Yoel Drori, Zhisheng Xiao, Lucio Dery, Rolf Jagerman, Jing Lu, Eric Ge, Vaibhav Aggarwal, Arjun Khare, Vinh Tran, Oded Elyada, Ferran Alet, James Rubin, Ian Chou, David Tian, Libin Bai, Lawrence Chan, Lukasz Lew, Karolis Misiunas, Taylan Bilal, Aniket Ray, Sindhu Raghuram, Alex Castro-Ros, Viral Carpenter, CJ Zheng, Michael Kilgore, Josef Broder, Emily Xue, Praveen Kallakuri, Dheeru Dua, Nancy Yuen, Steve Chien, John Schultz, Saurabh Agrawal, Reut Tsarfaty, Jingcao Hu, Ajay Kannan, Dror Marcus, Nisarg Kothari, Baochen Sun, Ben Horn, Matko Bošnjak, Ferjad Naeem, Dean Hirsch, Lewis Chiang, Boya Fang, Jie Han, Qifei Wang, Ben Hora, Antoine He, Mario Lučić, Beer Changpinyo, Anshuman Tripathi, John Youssef, Chester Kwak, Philippe Schlattner, Cat Graves, Rémi Leblond, Wenjun Zeng, Anders Andreassen, Gabriel Rasskin, Yue Song, Eddie Cao, Junhyuk Oh, Matt Hoffman, Wojtek Skut, Yichi Zhang, Jon Stritar, Xingyu Cai, Saarthak Khanna, Kathie Wang, Shriya Sharma, Christian Reisswig, Younghoon Jun, Aman Prasad, Tatiana Sholokhova, Preeti Singh, Adi Gerzi Rosenthal, Anian Ruoss, Françoise Beaufays, Sean Kirmani, Dongkai Chen, Johan Schalkwyk, Jonathan Herzig, Been Kim, Josh Jacob, Damien Vincent, Adrian N Reyes, Ivana Balazevic, Léonard Hussenot, Jon Schneider, Parker Barnes, Luis Castro, Spandana Raj Babbula, Simon Green, Serkan Cabi, Nico Duduta, Danny Driess, Rich Galt, Noam Velan, Junjie Wang, Hongyang Jiao, Matthew Mauger, Du Phan, Miteyan Patel, Vlado Galić, Jerry Chang, Eyal Marcus, Matt Harvey, Julian Salazar, Elahe Dabir, Suraj Satishkumar Sheth, Amol Mandhane, Hanie Sedghi, Jeremiah Willcock, Amir Zandieh, Shruthi Prabhakara, Aida Amini, Antoine Miech, Victor Stone, Massimo Nicosia, Paul Niemczyk, Ying Xiao, Lucy Kim, Sławek Kwasiborski, Vikas Verma, Ada Maksutaj Oflazer, Christoph Hirnschall, Peter Sung, Lu Liu, Richard Everett, Michiel Bakker, Ágoston Weisz, Yufei Wang, Vivek Sampathkumar, Uri Shaham, Bibo Xu, Yasemin Altun, Mingqiu Wang, Takaaki Saeki, Guanjie Chen, Emanuel Taropa, Shanthal Vasanth, Sophia Austin, Lu Huang, Goran Petrovic, Qingyun Dou, Daniel Golovin, Grigory Rozhdestvenskiy, Allie Culp, Will Wu, Motoki Sano, Divya Jain, Julia Proskurnia, Sébastien Cevey, Alejandro Cruzado Ruiz, Piyush Patil, Mahdi Mirzazadeh, Eric Ni, Javier Snaider, Lijie Fan, Alexandre Fréchette, AJ Pierigiovanni, Shariq Iqbal, Kenton Lee, Claudio Fantacci, Jinwei Xing, Lisa Wang, Alex Irpan, David Raposo, Yi Luan, Zhuoyuan Chen, Harish Ganapathy, Kevin Hui, Jiazhong Nie, Isabelle Guyon, Heming Ge, Roopali Vij, Hui Zheng, Dayeong Lee, Alfonso Castaño, Khuslen Baatarsukh, Gabriel Ibagon, Alexandra Chronopoulou, Nicholas FitzGerald, Shashank Viswanadha, Safeen Huda, Rivka Moroshko, Georgi Stoyanov, Prateek Kolhar, Alain Vaucher, Ishaan Watts, Adhi Kuncoro, Henryk Michalewski, Satish Kambala, Bat-Orgil Batsaikhan, Alek Andreev, Irina Jurenka, Maigo Le, Qihang Chen, Wael Al Jishi, Sarah Chakera, Zhe Chen, Aditya Kini, Vikas Yadav, Aditya Siddhant, Ilia Labzovsky, Balaji Lakshminarayanan, Carrie Grimes Bostock, Pankil Botadra, Ankesh Anand, Colton Bishop, Sam Conway-Rahman, Mohit Agarwal, Yani Donchev, Achintya Singhal, Félix de Chaumont Quitry, Natalia Ponomareva, Nishant Agrawal, Bin Ni, Kalpesh Krishna, Masha Samsikova, John Karro, Yilun

Du, Tamara von Glehn, Caden Lu, Christopher A. Choquette-Choo, Zhen Qin, Tingnan Zhang, Sicheng Li, Divya Tyam, Swaroop Mishra, Wing Lowe, Colin Ji, Weiyi Wang, Manaal Faruqui, Ambrose Slone, Valentin Dalibard, Arunachalam Narayanaswamy, John Lambert, Pierre-Antoine Manzagol, Dan Karliner, Andrew Bolt, Ivan Lobov, Aditya Kusupati, Chang Ye, Xuan Yang, Heiga Zen, Nelson George, Mukul Bhutani, Olivier Lacombe, Robert Riachi, Gagan Bansal, Rachel Soh, Yue Gao, Yang Yu, Adams Yu, Emily Nottage, Tania Rojas-Esponda, James Noraky, Manish Gupta, Ragha Kotikalapudi, Jichuan Chang, Sanja Deur, Dan Graur, Alex Mossin, Erin Farnese, Ricardo Figueira, Alexandre Moufarek, Austin Huang, Patrik Zochbauer, Ben Ingram, Tongzhou Chen, Zelin Wu, Adrià Puigdomènech, Leland Rechis, Da Yu, Sri Gayatri Sundara Padmanabhan, Rui Zhu, Chu ling Ko, Andrea Banino, Samira Daruki, Aarush Selvan, Dhruva Bhaswar, Daniel Hernandez Diaz, Chen Su, Salvatore Scellato, Jennifer Brennan, Woohyun Han, Grace Chung, Priyanka Agrawal, Urvashi Khandelwal, Khe Chai Sim, Morgane Lustman, Sam Ritter, Kelvin Guu, Jiawei Xia, Prateek Jain, Emma Wang, Tyrone Hill, Mirko Rossini, Marija Kostelac, Tautvydas Misiunas, Amit Sabne, Kyuyeun Kim, Ahmet Iscen, Congchao Wang, José Leal, Ashwin Sreevatsa, Utku Evci, Manfred Warmuth, Saket Joshi, Daniel Suo, James Lottes, Garrett Honke, Brendan Jou, Stefani Karp, Jieru Hu, Himanshu Sahni, Adrien Ali Taïga, William Kong, Samrat Ghosh, Renshen Wang, Jay Pavagadhi, Natalie Axelsson, Nikolai Grigorev, Patrick Siegler, Rebecca Lin, Guohui Wang, Emilio Parisotto, Sharath Maddineni, Krishan Subudhi, Eyal Ben-David, Elena Pochernina, Orgad Keller, Thi Avrahami, Zhe Yuan, Pulkit Mehta, Jialu Liu, Sherry Yang, Wendy Kan, Katherine Lee, Tom Funkhouser, Derek Cheng, Hongzhi Shi, Archit Sharma, Joe Kelley, Matan Eyal, Yury Malkov, Corentin Tallec, Yuval Bahat, Shen Yan, Xintian, Wu, David Lindner, Chengda Wu, Avi Caciularu, Xiyang Luo, Rodolphe Jenatton, Tim Zaman, Yingying Bi, Ilya Kornakov, Ganesh Mallya, Daisuke Ikeda, Itay Karo, Anima Singh, Colin Evans, Praneeth Netrapalli, Vincent Nallatamby, Isaac Tian, Yannis Assael, Vikas Raunak, Victor Carbune, Ioana Bica, Lior Madmoni, Dee Cattle, Snchit Grover, Krishna Somandepalli, Sid Lall, Amelio Vázquez-Reina, Riccardo Patana, Jiaqi Mu, Pranav Talluri, Maggie Tran, Rajeev Aggarwal, RJ Skerry-Ryan, Jun Xu, Mike Burrows, Xiaoyue Pan, Edouard Yvinec, Di Lu, Zhiying Zhang, Duc Dung Nguyen, Hairong Mu, Gabriel Barcik, Helen Ran, Lauren Beltrone, Krzysztof Choromanski, Dia Kharrat, Samuel Albanie, Sean Purser-haskell, David Bieber, Carrie Zhang, Jing Wang, Tom Hudson, Zhiyuan Zhang, Han Fu, Johannes Mauerer, Mohammad Hossein Bateni, AJ Maschinot, Bing Wang, Muye Zhu, Arjun Pillai, Tobias Weyand, Shuang Liu, Oscar Akerlund, Fred Bertsch, Vittal Premachandran, Alicia Jin, Vincent Roulet, Peter de Boursac, Shubham Mittal, Ndaba Ndebele, Georgi Karadzhov, Sahra Ghalebikesabi, Ricky Liang, Allen Wu, Yale Cong, Nimesh Ghelani, Sumeet Singh, Bahar Fatemi, Warren, Chen, Charles Kwong, Alexey Kolganov, Steve Li, Richard Song, Chenkai Kuang, Sobhan Miryoosefi, Dale Webster, James Wendt, Arkadiusz Socala, Guolong Su, Artur Mendonça, Abhinav Gupta, Xiaowei Li, Tomy Tsai, Qiong, Hu, Kai Kang, Angie Chen, Sertan Girgin, Yongqin Xian, Andrew Lee, Nolan Ramsden, Leslie Baker, Madeleine Clare Elish, Varvara Krayvanova, Rishabh Joshi, Jiri Simsa, Yao-Yuan Yang, Piotr Ambroszczyk, Dipankar Ghosh, Arjun Kar, Yuan Shangguan, Yumeya Yamamori, Yaroslav Akulov, Andy Brock, Haotian Tang, Siddharth Vashishtha, Rich Munoz, Andreas Steiner, Kalyan Andra, Daniel Eppens, Qixuan Feng, Hayato Kobayashi, Sasha Goldshtein, Mona El Mahdy, Xin Wang, Jilei, Wang, Richard Killam, Tom Kwiatkowski, Kavya Kopparapu, Serena Zhan, Chao Jia, Alexei Bendebury, Sheryl Luo, Adrià Recasens, Timothy Knight, Jing Chen, Mohak Patel, YaGuang Li, Ben Withbroe, Dean Weesner, Kush Bhatia, Jie Ren, Danielle Eisenbud, Ebrahim Songhori, Yanhua Sun, Travis Choma, Tasos Kementsietsidis, Lucas Manning, Brian Roark, Wael Farhan, Jie Feng, Susheel Tatineni, James Cobon-Kerr, Yunjie Li, Lisa Anne Hendricks, Isaac Noble, Chris Breaux, Nate Kushman, Liqian Peng, Fuzhao Xue, Taylor Tobin, Jamie Rogers, Josh Lipschultz, Chris Alberti, Alexey Vlaskin, Mostafa Dehghani, Roshan Sharma, Tris Warkentin, Chen-Yu Lee, Benigno Uria, Da-Cheng Juan, Angad Chandorkar, Hila Sheftel, Ruibo Liu, Elnaz Davoodi, Borja De Balle Pigem, Kedar Dhamdhere, David Ross, Jonathan Hoech, Mahdis Mahdieh, Li Liu, Qiujia Li, Liam McCafferty, Chenxi Liu, Markus Mircea, Yunting Song, Omkar Savant, Alaa Saade, Colin Cherry, Vincent Hellendoorn, Siddharth Goyal, Paul Pucciarelli, David Vilar Torres, Zohar Yahav, Hyo Lee, Lars Lowe Sjoesund, Christo Kirov, Bo Chang, Deepanway Ghoshal, Lu Li, Gilles Baechler, Sébastien Pereira, Tara Sainath, Anudhyan Boral, Dominik Grewe, Afief Halumi, Nguyet Minh Phu, Tianxiao Shen, Marco Tulio Ribeiro, Dhriti Varma, Alex Kaskasoli, Vlad Feinberg, Navneet Potti, Jarrod Kahn, Matheus Wisniewski, Shakir Mohamed, Arnar Mar Hrafnkelsson, Bobak Shahriari, Jean-Baptiste Lespiau, Lisa Patel, Legg Yeung, Tom Paine, Lantao Mei, Alex Ramirez, Rakesh Shivanna, Li Zhong, Josh Woodward, Guilherme Tubone, Samira Khan, Heng Chen, Elizabeth Nielsen, Catalin Ionescu, Utsav Prabhu, Mingcen Gao, Qingze Wang, Sean Augenstein, Neesha Subramaniam, Jason Chang, Fotis Iliopoulos, Jiaming Luo, Myriam Khan, Weicheng Kuo, Denis Teplyashin, Florence Perot, Logan Kilpatrick, Amir Globerson, Hongkun Yu, Anfal Siddiqui, Nick Sukhanov, Arun Kandoor, Umang Gupta, Marco Andreetto, Moran Ambar, Donnie Kim, Paweł Wesołowski, Sarah Perrin, Ben Limonchik, Wei Fan, Jim Stephan, Ian Stewart-Binks, Ryan Kappedal, Tong He, Sarah Cogan, Romina Datta, Tong Zhou, Jiayu Ye, Leandro Kieliger, Ana Ramalho, Kyle Kastner, Fabian Mentzer, Wei-Jen Ko, Arun Suggala, Tianhao Zhou, Shiraz Butt, Hana Strejček, Lior Belenki, Subhashini Venugopalan, Mingyang Ling, Evgenii Eltyshev, Yunxiao Deng, Geza Kovacs, Mukund Raghavachari, Hanjun Dai, Tal Schuster, Steven Schwarcz, Richard Nguyen, Arthur Nguyen, Gavin Buttimore, Shrestha Basu Mallick, Sudeep Gandhe, Seth Benjamin, Michal Jastrzebski, Le Yan, Sugato Basu, Chris Apps, Isabel Edkins, James Allingham, Immanuel Odisho, Tomas Kocisky, Jewel Zhao, Linting Xue, Apoorv Reddy, Chrysovalantis Anastasiou, Aviel Atias, Sam Redmond, Kieran Milan, Nicolas Heess, Herman Schmit, Allan Dafoe, Daniel Andor, Tynan Gangwani, Anca Dragan, Sheng

Zhang, Ashyana Kachra, Gang Wu, Siyang Xue, Kevin Aydin, Siqi Liu, Yuxiang Zhou, Mahan Malihi, Austin Wu, Siddharth Gopal, Candice Schumann, Peter Stys, Alek Wang, Mirek Olšák, Dangyi Liu, Christian Schallhart, Yiran Mao, Demetra Brady, Hao Xu, Tomas Mery, Chawin Sitawarin, Siva Velusamy, Tom Cobley, Alex Zhai, Christian Walder, Nitzan Katz, Ganesh Jawahar, Chinmay Kulkarni, Antoine Yang, Adam Paszke, Yinan Wang, Bogdan Damoc, Zalán Borsos, Ray Smith, Jinning Li, Mansi Gupta, Andrei Kapishnikov, Sushant Prakash, Florian Luisier, Rishabh Agarwal, Will Grathwohl, Kuangyuan Chen, Kehang Han, Nikhil Mehta, Andrew Over, Shekoofeh Azizi, Lei Meng, Niccolò Dal Santo, Kelvin Zheng, Jane Shapiro, Igor Petrovski, Jeffrey Hui, Amin Ghafouri, Jasper Snoek, James Qin, Mandy Jordan, Caitlin Sikora, Jonathan Malmaud, Yuheng Kuang, Aga Świetlik, Ruoxin Sang, Chongyang Shi, Leon Li, Andrew Rosenberg, Shubin Zhao, Andy Crawford, Jan-Thorsten Peter, Yun Lei, Xavier Garcia, Long Le, Todd Wang, Julien Amelot, Dave Orr, Praneeth Kacham, Dana Alon, Gladys Tyen, Abhinav Arora, James Lyon, Alex Kurakin, Mimi Ly, Theo Guidroz, Zhipeng Yan, Rina Panigrahy, Pingmei Xu, Thais Kagohara, Yong Cheng, Eric Noland, Jinhyuk Lee, Jonathan Lee, Cathy Yip, Maria Wang, Efrat Nehoran, Alexander Bykovsky, Zhihao Shan, Ankit Bhagatwala, Chaochao Yan, Jie Tan, Guillermo Garrido, Dan Ethier, Nate Hurley, Grace Vesom, Xu Chen, Siyuan Qiao, Abhishek Nayyar, Julian Walker, Paramjit Sandhu, Mihaela Rosca, Danny Swisher, Mikhail Dektiarev, Josh Dillon, George-Cristian Muraru, Manuel Tragut, Artiom Myaskovsky, David Reid, Marko Velic, Owen Xiao, Jasmine George, Mark Brand, Jing Li, Wenhao Yu, Shane Gu, Xiang Deng, François-Xavier Aubet, Soheil Hassas Yeganeh, Fred Alcober, Celine Smith, Trevor Cohn, Kay McKinney, Michael Tschannen, Ramesh Sampath, Gowoon Cheon, Liangchen Luo, Luyang Liu, Jordi Orbay, Hui Peng, Gabriela Botea, Xiaofan Zhang, Charles Yoon, Cesar Magalhaes, Paweł Stradomski, Ian Mackinnon, Steven Hemingray, Kumaran Venkatesan, Rhys May, Jaeyoun Kim, Alex Druinsky, Jingchen Ye, Zheng Xu, Terry Huang, Jad Al Abdallah, Adil Dostmohamed, Rachana Fellinger, Tsendsuren Munkhdalai, Akanksha Maurya, Peter Garst, Yin Zhang, Maxim Krikun, Simon Bucher, Aditya Srikanth Veerubhotla, Yaxin Liu, Sheng Li, Nishesh Gupta, Jakub Adamek, Hanwen Chen, Bernett Orlando, Aleksandr Zaks, Joost van Amersfoort, Josh Camp, Hui Wan, HyunJeong Choe, Zhichun Wu, Kate Olszewska, Weiren Yu, Archita Vadali, Martin Scholz, Daniel De Freitas, Jason Lin, Amy Hua, Xin Liu, Frank Ding, Yichao Zhou, Boone Severson, Katerina Tsihlas, Samuel Yang, Tammo Spalink, Varun Yerram, Helena Pankov, Rory Blevins, Ben Vargas, Sarthak Jauhari, Matt Miecnikowski, Ming Zhang, Sandeep Kumar, Clement Farabet, Charline Le Lan, Sebastian Flennerhag, Yonatan Bitton, Ada Ma, Arthur Bražinskas, Eli Collins, Niharika Ahuja, Sneha Kudugunta, Anna Bortsova, Minh Giang, Wanzheng Zhu, Ed Chi, Scott Lundberg, Alexey Stern, Subha Puttagunta, Jing Xiong, Xiao Wu, Yash Pande, Amit Jhindal, Daniel Murphy, Jon Clark, Marc Brockschmidt, Maxine Deines, Kevin R. McKee, Dan Bahir, Jiajun Shen, Minh Truong, Daniel McDuff, Andrea Gesmundo, Edouard Rosseel, Bowen Liang, Ken Caluwaerts, Jessica Hamrick, Joseph Kready, Mary Cassin, Rishikesh Ingale, Li Lao, Scott Pollom, Yifan Ding, Wei He, Lizzetth Bellot, Joana Iljazi, Ramya Sree Boppana, Shan Han, Tara Thompson, Amr Khalifa, Anna Bulanova, Blagoj Mitrevski, Bo Pang, Emma Cooney, Tian Shi, Rey Coaguila, Tamar Yakar, Marc'aurelio Ranzato, Nikola Momchev, Chris Rawles, Zachary Charles, Young Maeng, Yuan Zhang, Rishabh Bansal, Xiaokai Zhao, Brian Albert, Yuan Yuan, Sudheendra Vijayanarasimhan, Roy Hirsch, Vinay Ramasesh, Kiran Vodrahalli, Xingyu Wang, Arushi Gupta, DJ Strouse, Jianmo Ni, Roma Patel, Gabe Taubman, Zhouyuan Huo, Dero Gharibian, Marianne Monteiro, Hoi Lam, Shobha Vasudevan, Aditi Chaudhary, Isabela Albuquerque, Kilol Gupta, Sebastian Riedel, Chaitra Hegde, Avraham Ruderman, András György, Marcus Wainwright, Ashwin Chaugule, Burcu Karagol Ayan, Tomer Levinboim, Sam Shleifer, Yogesh Kalley, Vahab Mirrokni, Abhishek Rao, Prabakar Radhakrishnan, Jay Hartford, Jialin Wu, Zhenhai Zhu, Francesco Bertolini, Hao Xiong, Nicolas Serrano, Hamish Tomlinson, Myle Ott, Yifan Chang, Mark Graham, Jian Li, Marco Liang, Xiangzhu Long, Sebastian Borgeaud, Yanif Ahmad, Alex Grills, Diana Mincu, Martin Izzard, Yuan Liu, Jinyu Xie, Louis O'Bryan, Sameera Ponda, Simon Tong, Michelle Liu, Dan Malkin, Khalid Salama, Yuankai Chen, Rohan Anil, Anand Rao, Rigel Swavely, Misha Bilenko, Nina Anderson, Tat Tan, Jing Xie, Xing Wu, Lijun Yu, Oriol Vinyals, Andrey Ryabtsev, Rumen Dangovski, Kate Baumli, Daniel Keysers, Christian Wright, Zoe Ashwood, Betty Chan, Artem Shtefan, Yaohui Guo, Ankur Bapna, Radu Soricut, Steven Pecht, Sabela Ramos, Rui Wang, Jiahao Cai, Trieu Trinh, Paul Barham, Linda Friso, Eli Stickgold, Xiangzhuo Ding, Siamak Shakeri, Diego Ardila, Eleftheria Briakou, Phil Culliton, Adam Raveret, Jingyu Cui, David Saxton, Subhrajit Roy, Javad Azizi, Pengcheng Yin, Lucia Loher, Andrew Bunner, Min Choi, Faruk Ahmed, Eric Li, Yin Li, Shengyang Dai, Michael Elabd, Sriram Ganapathy, Shivani Agrawal, Yiqing Hua, Paige Kunkle, Sujeevan Rajayogam, Arun Ahuja, Arthur Conmy, Alex Vasiloff, Parker Beak, Christopher Yew, Jayaram Mudigonda, Bartek Wydrowski, Jon Blanton, Zhengdong Wang, Yann Dauphin, Zhuo Xu, Martin Polacek, Xi Chen, Hexiang Hu, Pauline Sho, Markus Kunesch, Mehdi Hafezi Manshadi, Eliza Rutherford, Bo Li, Sissie Hsiao, Iain Barr, Alex Tudor, Matija Kecman, Arsha Nagrani, Vladimir Pchelin, Martin Sundermeyer, Aishwarya P S, Abhijit Karmarkar, Yi Gao, Grishma Chole, Olivier Bachem, Isabel Gao, Arturo BC, Matt Dibb, Mauro Verzetti, Felix Hernandez-Campos, Yana Lunts, Matthew Johnson, Julia Di Trapani, Raphael Koster, Idan Brusilovsky, Binbin Xiong, Megha Mohabey, Han Ke, Joe Zou, Tea Sabolić, Víctor Campos, John Palowitch, Alex Morris, Linhai Qiu, Pranavaraj Ponnuramu, Fangtao Li, Vivek Sharma, Kiranbir Sodhia, Kaan Tekelioglu, Aleksandr Chuklin, Madhavi Yenugula, Erika Gemzer, Theofilos Strinopoulos, Sam El-Husseini, Huiyu Wang, Yan Zhong, Edouard Leurent, Paul Natsev, Weijun Wang, Dre Mahaarachchi, Tao Zhu, Songyou Peng, Sami Alabed, Cheng-Chun Lee, Anthony Brohan, Arthur Szlam, GS Oh, Anton Kovsharov, Jenny Lee, Renee Wong, Megan Barnes, Gregory Thornton, Felix Gimeno, Omer Levy, Martin Sevenich, Melvin Johnson,

Jonathan Mallinson, Robert Dadashi, Ziyue Wang, Qingchun Ren, Preethi Lahoti, Arka Dhar, Josh Feldman, Dan Zheng, Thatcher Ulrich, Liviu Panait, Michiel Blokzijl, Cip Baetu, Josip Matak, Jitendra Harlalka, Maulik Shah, Tal Marian, Daniel von Dincklage, Cosmo Du, Ruy Ley-Wild, Bethanie Brownfield, Max Schumacher, Yury Stuken, Shadi Noghabi, Sonal Gupta, Xiaoqi Ren, Eric Malmi, Felix Weissenberger, Blanca Huergo, Maria Bauza, Thomas Lampe, Arthur Douillard, Mojtaba Seyedhosseini, Roy Frostig, Zoubin Ghahramani, Kelvin Nguyen, Kashyap Krishnakumar, Chengxi Ye, Rahul Gupta, Alireza Nazari, Robert Geirhos, Pete Shaw, Ahmed Eleryan, Dima Damen, Jennimaria Palomaki, Ted Xiao, Qiyin Wu, Quan Yuan, Phoenix Meadowlark, Matthew Bilotti, Raymond Lin, Mukund Sridhar, Yannick Schroecker, Da-Woon Chung, Jincheng Luo, Trevor Strohman, Tianlin Liu, Anne Zheng, Jesse Emond, Wei Wang, Andrew Lampinen, Toshiyuki Fukuzawa, Folawiyo Campbell-Ajala, Monica Roy, James Lee-Thorp, Lily Wang, Iftekhar Naim, Tony, Nguy ên, Guy Bensky, Aditya Gupta, Dominika Rogozińska, Justin Fu, Thanumalayan Sankaranarayana Pillai, Petar Veličković, Shahar Drath, Philipp Neubeck, Vaibhav Tulsyan, Arseniy Klimovskiy, Don Metzler, Sage Stevens, Angel Yeh, Junwei Yuan, Tianhe Yu, Kelvin Zhang, Alec Go, Vincent Tsang, Ying Xu, Andy Wan, Isaac Galatzer-Levy, Sam Sobell, Abodunrinwa Toki, Elizabeth Salesky, Wenlei Zhou, Diego Antognini, Sholto Douglas, Shimu Wu, Adam Lelkes, Frank Kim, Paul Cavallaro, Ana Salazar, Yuchi Liu, James Besley, Tiziana Refice, Yiling Jia, Zhang Li, Michal Sokolik, Arvind Kannan, Jon Simon, Jo Chick, Avia Aharon, Meet Gandhi, Mayank Daswani, Keyvan Amiri, Vighnesh Birodkar, Abe Ittycheriah, Peter Grabowski, Oscar Chang, Charles Sutton, Zhixin, Lai, Umesh Telang, Susie Sargsyan, Tao Jiang, Raphael Hoffmann, Nicole Brichtova, Matteo Hessel, Jonathan Halcrow, Sammy Jerome, Geoff Brown, Alex Tomala, Elena Buchatskaya, Dian Yu, Sachit Menon, Pol Moreno, Yuguo Liao, Vicky Zayats, Luming Tang, SQ Mah, Ashish Shenoy, Alex Siegman, Majid Hadian, Okwan Kwon, Tao Tu, Nima Khajehnouri, Ryan Foley, Parisa Haghani, Zhongru Wu, Vaishakh Keshava, Khyatti Gupta, Tony Bruguier, Rui Yao, Danny Karmon, Luisa Zintgraf, Zhicheng Wang, Enrique Piqueras, Junehyuk Jung, Jenny Brennan, Diego Machado, Marissa Giustina, MH Tessler, Kamyu Lee, Qiao Zhang, Joss Moore, Kaspar Daugaard, Alexander Frömmgen, Jennifer Beattie, Fred Zhang, Daniel Kasenberg, Ty Geri, Danfeng Qin, Gaurav Singh Tomar, Tom Ouyang, Tianli Yu, Luowei Zhou, Rajiv Mathews, Andy Davis, Yaoyiran Li, Jai Gupta, Damion Yates, Linda Deng, Elizabeth Kemp, Ga-Young Joung, Sergei Vassilvitskii, Mandy Guo, Pallavi LV, Dave Dopson, Sami Lachgar, Lara McConnaughey, Himadri Choudhury, Dragos Dena, Aaron Cohen, Joshua Ainslie, Sergey Levi, Parthasarathy Gopavarapu, Polina Zablotskaia, Hugo Vallet, Sanaz Bahargam, Xiaodan Tang, Nenad Tomasev, Ethan Dyer, Daniel Balle, Hongrae Lee, William Bono, Jorge Gonzalez Mendez, Vadim Zubov, Shentao Yang, Ivor Rendulic, Yanyan Zheng, Andrew Hogue, Golan Pundak, Ralph Leith, Avishkar Bhoopchand, Michael Han, Mislav Žanić, Tom Schaul, Manolis Delakis, Tejas Iyer, Guanyu Wang, Harman Singh, Abdelrahman Abdelhamed, Tara Thomas, Siddhartha Brahma, Hilal Dib, Naveen Kumar, Wenxuan Zhou, Liang Bai, Pushkar Mishra, Jiao Sun, Valentin Anklin, Roykrong Sukkerd, Lauren Agubuzu, Anton Briukhov, Anmol Gulati, Maximilian Sieb, Fabio Pardo, Sara Nasso, Junquan Chen, Kexin Zhu, Tiberiu Sosea, Alex Goldin, Keith Rush, Spurthi Amba Hombaiah, Andreas Noever, Allan Zhou, Sam Haves, Mary Phuong, Jake Ades, Yi ting Chen, Lin Yang, Joseph Pagadora, Stan Bileschi, Victor Cotruta, Rachel Saputro, Arijit Pramanik, Sean Ammirati, Dan Garrette, Kevin Villela, Tim Blyth, Canfer Akbulut, Neha Jha, Alban Rrustemi, Arissa Wongpanich, Chirag Nagpal, Yonghui Wu, Morgane Rivière, Sergey Kishchenko, Pranesh Srinivasan, Alice Chen, Animesh Sinha, Trang Pham, Bill Jia, Tom Hennigan, Anton Bakalov, Nithya Attaluri, Drew Garmon, Daniel Rodriguez, Dawid Wegner, Wenhao Jia, Evan Senter, Noah Fiedel, Denis Petek, Yuchuan Liu, Cassidy Hardin, Harshal Tushar Lehri, Joao Carreira, Sara Smoot, Marcel Prasetya, Nami Akazawa, Anca Stefanoiu, Chia-Hua Ho, Anelia Angelova, Kate Lin, Min Kim, Charles Chen, Marcin Sieniek, Alice Li, Tongfei Guo, Sorin Baltateanu, Pouya Tafti, Michael Wunder, Nadav Olmert, Divyansh Shukla, Jingwei Shen, Neel Kovelamudi, Balaji Venkatraman, Seth Neel, Romal Thoppilan, Jerome Connor, Frederik Benzing, Axel Stjerngren, Golnaz Ghiasi, Alex Polozov, Joshua Howland, Theophane Weber, Justin Chiu, Ganesh Poomal Girirajan, Andreas Terzis, Pidong Wang, Fangda Li, Yoav Ben Shalom, Dinesh Tewari, Matthew Denton, Roee Aharoni, Norbert Kalb, Heri Zhao, Junlin Zhang, Angelos Filos, Matthew Rahtz, Lalit Jain, Connie Fan, Vitor Rodrigues, Ruth Wang, Richard Shin, Jacob Austin, Roman Ring, Mariella Sanchez-Vargas, Mehadi Hassen, Ido Kessler, Uri Alon, Gufeng Zhang, Wenhu Chen, Yenai Ma, Xiance Si, Le Hou, Azalia Mirhoseini, Marc Wilson, Geoff Bacon, Becca Roelofs, Lei Shu, Gautam Vasudevan, Jonas Adler, Artur Dwornik, Tayfun Terzi, Matt Lawlor, Harry Askham, Mike Bernico, Xuanyi Dong, Chris Hidey, Kevin Kilgour, Gaël Liu, Surya Bhupatiraju, Luke Leonhard, Siqi Zuo, Partha Talukdar, Qing Wei, Aliaksei Severyn, Vít Listík, Jong Lee, Aditya Tripathi, SK Park, Yossi Matias, Hao Liu, Alex Ruiz, Rajesh Jayaram, Jackson Tolins, Pierre Marcenac, Yiming Wang, Bryan Seybold, Henry Prior, Deepak Sharma, Jack Weber, Mikhail Sirotenko, Yunhsuan Sung, Dayou Du, Ellie Pavlick, Stefan Zinke, Markus Freitag, Max Dylla, Montse Gonzalez Arenas, Natan Potikha, Omer Goldman, Connie Tao, Rachita Chhaparia, Maria Voitovich, Pawan Dogra, Andrija Ražnatović, Zak Tsai, Chong You, Oleaser Johnson, George Tucker, Chenjie Gu, Jae Yoo, Maryam Majzoubi, Valentin Gabeur, Bahram Raad, Rocky Rhodes, Kashyap Kolipaka, Heidi Howard, Geta Sampemane, Benny Li, Chulayuth Asawaroengchai, Duy Nguyen, Chiyuan Zhang, Timothee Cour, Xinxin Yu, Zhao Fu, Joe Jiang, Po-Sen Huang, Gabriela Surita, Iñaki Iturrate, Yael Karov, Michael Collins, Martin Baeuml, Fabian Fuchs, Shilpa Shetty, Swaroop Ramaswamy, Sayna Ebrahimi, Qiuchen Guo, Jeremy Shar, Gabe Barth-Maron, Sravanti Addepalli, Bryan Richter, Chin-Yi Cheng, Eugénie Rives, Fei Zheng, Johannes Griesser, Nishanth Dikkala, Yoel Zeldes, Ilkin Safarli, Dipanjan Das, Himanshu Srivastava, Sadh MNM Khan, Xin

Li, Aditya Pandey, Larisa Markeeva, Dan Belov, Qiqi Yan, Mikołaj Rybiński, Tao Chen, Megha Nawhal, Michael Quinn, Vineetha Govindaraj, Sarah York, Reed Roberts, Roopal Garg, Namrata Godbole, Jake Abernethy, Anil Das, Lam Nguyen Thiet, Jonathan Tompson, John Nham, Neera Vats, Ben Caine, Wesley Helmholz, Francesco Pongetti, Yeongil Ko, James An, Clara Huiyi Hu, Yu-Cheng Ling, Julia Pawar, Robert Leland, Keisuke Kinoshita, Waleed Khawaja, Marco Selvi, Eugene Ie, Danila Sinopalnikov, Lev Proleev, Nilesh Tripuraneni, Michele Bevilacqua, Seungji Lee, Clayton Sanford, Dan Suh, Dustin Tran, Jeff Dean, Simon Baumgartner, Jens Heitkaemper, Sagar Gubbi, Kristina Toutanova, Yichong Xu, Chandu Thekkath, Keran Rong, Palak Jain, Annie Xie, Yan Virin, Yang Li, Lubo Litchev, Richard Powell, Tarun Bharti, Adam Kraft, Nan Hua, Marissa Ikonomidis, Ayal Hitron, Sanjiv Kumar, Loic Matthey, Sophie Bridgers, Lauren Lax, Ishaan Malhi, Ondrej Skopek, Ashish Gupta, Jiawei Cao, Mitchelle Rasquinha, Siim Põder, Wojciech Stokowiec, Nicholas Roth, Guowang Li, Michaël Sander, Joshua Kessinger, Vihan Jain, Edward Loper, Wonpyo Park, Michal Yarom, Liqun Cheng, Guru Guruganesh, Kanishka Rao, Yan Li, Catarina Barros, Mikhail Sushkov, Chun-Sung Ferng, Rohin Shah, Ophir Aharoni, Ravin Kumar, Tim McConnell, Peiran Li, Chen Wang, Fernando Pereira, Craig Swanson, Fayaz Jamil, Yan Xiong, Anitha Vijayakumar, Prakash Shroff, Kedar Soparkar, Jindong Gu, Livio Baldini Soares, Eric Wang, Kushal Majmundar, Aurora Wei, Kai Bailey, Nora Kassner, Chizu Kawamoto, Goran Žužić, Victor Gomes, Abhirut Gupta, Michael Guzman, Ishita Dasgupta, Xinyi Bai, Zhufeng Pan, Francesco Piccinno, Hadas Natalie Vogel, Octavio Ponce, Adrian Hutter, Paul Chang, Pan-Pan Jiang, Ionel Gog, Vlad Ionescu, James Manyika, Fabian Pedregosa, Harry Ragan, Zach Behrman, Ryan Mullins, Coline Devin, Aroonalok Pyne, Swapnil Gawde, Martin Chadwick, Yiming Gu, Sasan Tavakkol, Andy Twigg, Naman Goyal, Ndidi Elue, Anna Goldie, Srinivasan Venkatachary, Hongliang Fei, Ziqiang Feng, Marvin Ritter, Isabel Leal, Sudeep Dasari, Pei Sun, Alif Raditya Rochman, Brendan O'Donoghue, Yuchen Liu, Jim Sproch, Kai Chen, Natalie Clay, Slav Petrov, Sailesh Sidhwani, Ioana Mihailescu, Alex Panagopoulos, AJ Piergiovanni, Yunfei Bai, George Powell, Deep Karkhanis, Trevor Yacovone, Petr Mitrichev, Joe Kovac, Dave Uthus, Amir Yazdanbakhsh, David Amos, Steven Zheng, Bing Zhang, Jin Miao, Bhuvana Ramabhadran, Soroush Radpour, Shantanu Thakoor, Josh Newlan, Oran Lang, Orion Jankowski, Shikhar Bharadwaj, Jean-Michel Sarr, Shereen Ashraf, Sneha Mondal, Jun Yan, Ankit Singh Rawat, Sarmishta Velury, Greg Kochanski, Tom Eccles, Franz Och, Abhanshu Sharma, Ethan Mahintorabi, Alex Gurney, Carrie Muir, Vered Cohen, Saksham Thakur, Adam Bloniarz, Asier Mujika, Alexander Pritzel, Paul Caron, Altaf Rahman, Fiona Lang, Yasumasa Onoe, Petar Sirkovic, Jay Hoover, Ying Jian, Pablo Duque, Arun Narayanan, David Soergel, Alex Haig, Loren Maggiore, Shyamal Buch, Josef Dean, Ilya Figotin, Igor Karpov, Shaleen Gupta, Denny Zhou, Muhuan Huang, Ashwin Vaswani, Christopher Semturs, Kaushik Shivakumar, Yu Watanabe, Vinodh Kumar Rajendran, Eva Lu, Yanhan Hou, Wenting Ye, Shikhar Vashishth, Nana Nti, Vytenis Sakenas, Darren Ni, Doug DeCarlo, Michael Benderskey, Sumit Bagri, Nacho Cano, Elijah Peake, Simon Tokumine, Varun Godbole, Carlos Guía, Tanya Lando, Vittorio Selo, Seher Ellis, Danny Tarlow, Daniel Gillick, Alessandro Epasto, Siddhartha Reddy Jonnalagadda, Meng Wei, Meiyan Xie, Ankur Taly, Michela Paganini, Mukund Sundararajan, Daniel Toyama, Ting Yu, Dessie Petrova, Aneesh Pappu, Rohan Agrawal, Senaka Buthpitiya, Justin Frye, Thomas Buschmann, Remi Crocker, Marco Tagliasacchi, Mengchao Wang, Da Huang, Sagi Perel, Brian Wieder, Hideto Kazawa, Weiyue Wang, Jeremy Cole, Himanshu Gupta, Ben Golan, Seojin Bang, Nitish Kulkarni, Ken Franko, Casper Liu, Doug Reid, Sid Dalmia, Jay Whang, Kevin Cen, Prasha Sundaram, Johan Ferret, Berivan Isik, Lucian Ionita, Guan Sun, Anna Shekhawat, Muqthar Mohammad, Philip Pham, Ronny Huang, Karthik Raman, Xingyi Zhou, Ross Mcilroy, Austin Myers, Sheng Peng, Jacob Scott, Paul Covington, Sofia Erell, Pratik Joshi, João Gabriel Oliveira, Natasha Noy, Tajwar Nasir, Jake Walker, Vera Axelrod, Tim Dozat, Pu Han, Chun-Te Chu, Eugene Weinstein, Anand Shukla, Shreyas Chandrakaladharan, Petra Poklukar, Bonnie Li, Ye Jin, Prem Eruvbetine, Steven Hansen, Avigail Dabush, Alon Jacovi, Samrat Phatale, Chen Zhu, Steven Baker, Mo Shomrat, Yang Xiao, Jean Pouget-Abadie, Mingyang Zhang, Fanny Wei, Yang Song, Helen King, Yiling Huang, Yun Zhu, Ruoxi Sun, Juliana Vicente Franco, Chu-Cheng Lin, Sho Arora, Hui, Li, Vivian Xia, Luke Vilnis, Mariano Schain, Kaiz Alarakyia, Laurel Prince, Aaron Phillips, Caleb Habtegebriel, Luyao Xu, Huan Gui, Santiago Ontanon, Lora Aroyo, Karan Gill, Peggy Lu, Yash Katariya, Dhruv Madeka, Shankar Krishnan, Shubha Srinivas Raghvendra, James Freedman, Yi Tay, Gaurav Menghani, Peter Choy, Nishita Shetty, Dan Abolafia, Doron Kukliansky, Edward Chou, Jared Lichtarge, Ken Burke, Ben Coleman, Dee Guo, Larry Jin, Indro Bhattacharya, Victoria Langston, Yiming Li, Suyog Kotecha, Alex Yakubovich, Xinyun Chen, Petre Petrov, Tolly Powell, Yanzhang He, Corbin Quick, Kanav Garg, Dawsen Hwang, Yang Lu, Srinadh Bhojanapalli, Kristian Kjems, Ramin Mehran, Aaron Archer, Hado van Hasselt, Ashwin Balakrishna, JK Kearns, Meiqi Guo, Jason Riesa, Mikita Sazanovich, Xu Gao, Chris Sauer, Chengrun Yang, XiangHai Sheng, Thomas Jimma, Wouter Van Gansbeke, Vitaly Nikolaev, Wei Wei, Katie Millican, Ruizhe Zhao, Justin Snyder, Levent Bolelli, Maura O'Brien, Shawn Xu, Fei Xia, Wentao Yuan, Arvind Neelakantan, David Barker, Sachin Yadav, Hannah Kirkwood, Farooq Ahmad, Joel Wee, Jordan Grimstad, Boyu Wang, Matthew Wiethoff, Shane Settle, Miaosen Wang, Charles Blundell, Jingjing Chen, Chris Duvarney, Grace Hu, Olaf Ronneberger, Alex Lee, Yuanzhen Li, Abhishek Chakladar, Alena Butryna, Georgios Evangelopoulos, Guillaume Desjardins, Jonni Kanerva, Henry Wang, Averi Nowak, Nick Li, Alyssa Loo, Art Khurshudov, Laurent El Shafey, Nagabhushan Baddi, Karel Lenc, Yasaman Razeghi, Tom Lieber, Amer Sinha, Xiao Ma, Yao Su, James Huang, Asahi Ushio, Hanna Klimczak-Plucińska, Kareem Mohamed, JD Chen, Simon Osindero, Stav Ginzburg, Lampros Lamprou, Vasilisa Bashlovkina, Duc-Hieu Tran, Ali Khodaei, Ankit Anand, Yixian Di, Ramy Eskander, Manish Reddy Vuyyuru, Jasmine Liu, Aishwarya Kamath, Roman Goldenberg, Math-

ias Bellaiche, Juliette Pluto, Bill Rosgen, Hassan Mansoor, William Wong, Suhas Ganesh, Eric Bailey, Scott Baird, Dan Deutsch, Jinoo Baek, Xuhui Jia, Chansoo Lee, Abe Friesen, Nathaniel Braun, Kate Lee, Amayika Panda, Steven M. Hernandez, Duncan Williams, Jianqiao Liu, Ethan Liang, Arnaud Autef, Emily Pitler, Deepali Jain, Phoebe Kirk, Oskar Bunyan, Jaume Sanchez Elias, Tongxin Yin, Machel Reid, Aedan Pope, Nikita Putikhin, Bidisha Samanta, Sergio Guadarrama, Dahun Kim, Simon Rowe, Marcella Valentine, Geng Yan, Alex Salcianu, David Silver, Gan Song, Richa Singh, Shuai Ye, Hannah DeBalsi, Majd Al Merey, Eran Ofek, Albert Webson, Shibl Mourad, Ashwin Kakarla, Silvio Lattanzi, Nick Roy, Evgeny Sluzhaev, Christina Butterfield, Alessio Tonioni, Nathan Waters, Sudhindra Kopalle, Jason Chase, James Cohan, Girish Ramchandra Rao, Robert Berry, Michael Voznesensky, Shuguang Hu, Kristen Chiafullo, Sharat Chikkerur, George Scrivener, Ivy Zheng, Jeremy Wiesner, Wolfgang Macherey, Timothy Lillicrap, Fei Liu, Brian Walker, David Welling, Elinor Davies, Yangsibo Huang, Lijie Ren, Nir Shabat, Alessandro Agostini, Mariko Iinuma, Dustin Zelle, Rohit Sathyanarayana, Andrea D'olimpio, Morgan Redshaw, Matt Ginsberg, Ashwin Murthy, Mark Geller, Tatiana Matejovicova, Ayan Chakrabarti, Ryan Julian, Christine Chan, Qiong Hu, Daniel Jarrett, Manu Agarwal, Jeshwanth Challagundla, Tao Li, Sandeep Tata, Wen Ding, Maya Meng, Zhuyun Dai, Giulia Vezzani, Shefali Garg, Jannis Bulian, Mary Jasarevic, Honglong Cai, Harish Rajamani, Adam Santoro, Florian Hartmann, Chen Liang, Bartek Perz, Apoorv Jindal, Fan Bu, Sungyong Seo, Ryan Poplin, Adrian Goedeckemeyer, Badih Ghazi, Nikhil Khadke, Leon Liu, Kevin Mather, Mingda Zhang, Ali Shah, Alex Chen, Jinliang Wei, Keshav Shivam, Yuan Cao, Donghyun Cho, Angelo Scorza Scarpati, Michael Moffitt, Clara Barbu, Ivan Jurin, Ming-Wei Chang, Hongbin Liu, Hao Zheng, Shachi Dave, Christine Kaeser-Chen, Xiaobin Yu, Alvin Abdagic, Lucas Gonzalez, Yanping Huang, Peilin Zhong, Cordelia Schmid, Bryce Petrini, Alex Wertheim, Jifan Zhu, Hoang Nguyen, Kaiyang Ji, Yanqi Zhou, Tao Zhou, Fangxiaoyu Feng, Regev Cohen, David Rim, Shubham Milind Phal, Petko Georgiev, Ariel Brand, Yue Ma, Wei Li, Somit Gupta, Chao Wang, Pavel Dubov, Jean Tarbouriech, Kingshuk Majumder, Huijian Li, Norman Rink, Apurv Suman, Yang Guo, Yinghao Sun, Arun Nair, Xiaowei Xu, Mohamed Elhawaty, Rodrigo Cabrera, Guangxing Han, Julian Eisenschlos, Junwen Bai, Yuqi Li, Yamini Bansal, Thibault Sellam, Mina Khan, Hung Nguyen, Justin Mao-Jones, Nikos Parotsidis, Jake Marcus, Cindy Fan, Roland Zimmermann, Yony Kochinski, Laura Graesser, Feryal Behbahani, Alvaro Caceres, Michael Riley, Patrick Kane, Sandra Lefdal, Rob Willoughby, Paul Vicol, Lun Wang, Shujian Zhang, Ashleah Gill, Yu Liang, Gautam Prasad, Soroosh Mariooryad, Mehran Kazemi, Zifeng Wang, Kritika Muralidharan, Paul Voigtlaender, Jeffrey Zhao, Huanjie Zhou, Nina D'Souza, Aditi Mavalankar, Séb Arnold, Nick Young, Obaid Sarvana, Chace Lee, Milad Nasr, Tingting Zou, Seokhwan Kim, Lukas Haas, Kaushal Patel, Neslihan Bulut, David Parkinson, Courtney Biles, Dmitry Kalashnikov, Chi Ming To, Aviral Kumar, Jessica Austin, Alex Greve, Lei Zhang, Megha Goel, Yeqing Li, Sergey Yaroshenko, Max Chang, Abhishek Jindal, Geoff Clark, Hagai Taitelbaum, Dale Johnson, Ofir Roval, Jeongwoo Ko, Anhad Mohananey, Christian Schuler, Shenil Dodhia, Ruichao Li, Kazuki Osawa, Claire Cui, Peng Xu, Rushin Shah, Tao Huang, Ela Gruzewska, Nathan Clement, Mudit Verma, Olcan Sercinoglu, Hai Qian, Viral Shah, Masa Yamaguchi, Abhinit Modi, Takahiro Kosakai, Thomas Strohmann, Junhao Zeng, Beliz Gunel, Jun Qian, Austin Tarango, Krzysztof Jastrzębski, Robert David, Jyn Shan, Parker Schuh, Kunal Lad, Willi Gierke, Mukundan Madhavan, Xinyi Chen, Mark Kurzeja, Rebeca Santamaria-Fernandez, Dawn Chen, Alexandra Cordell, Yuri Chervonyi, Frankie Garcia, Nithish Kannen, Vincent Perot, Nan Ding, Shlomi Cohen-Ganor, Victor Lavrenko, Junru Wu, Georgie Evans, Cicero Nogueira dos Santos, Madhavi Sewak, Ashley Brown, Andrew Hard, Joan Puigcerver, Zeyu Zheng, Yizhong Liang, Evgeny Gladchenko, Reeve Ingle, Uri First, Pierre Sermanet, Charlotte Magister, Mihajlo Velimirović, Sashank Reddi, Susanna Ricco, Eirikur Agustsson, Hartwig Adam, Nir Levine, David Gaddy, Dan Holtmann-Rice, Xuanhui Wang, Ashutosh Sathe, Abhijit Guha Roy, Blaž Bratanič, Alen Carin, Harsh Mehta, Silvano Bonacina, Nicola De Cao, Mara Finkelstein, Verena Rieser, Xinyi Wu, Florent Altché, Dylan Scandinaro, Li Li, Nino Vieillard, Nikhil Sethi, Garrett Tanzer, Zhi Xing, Shibo Wang, Parul Bhatia, Gui Citovsky, Thomas Anthony, Sharon Lin, Tianze Shi, Shoshana Jakobovits, Gena Gibson, Raj Apte, Lisa Lee, Mingqing Chen, Arunkumar Byravan, Petros Maniatis, Kellie Webster, Andrew Dai, Pu-Chin Chen, Jiaqi Pan, Asya Fadeeva, Zach Gleicher, Thang Luong, and Niket Kumar Bhumihar. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. 4

[7] Shaurya Dewan, Rushikesh Zawar, Prakanshul Saxena, Yingshan Chang, Andrew Luo, and Yonatan Bisk. Diffusionpid: Interpreting diffusion via partial information decomposition. *CoRR*, abs/2406.05191, 2024. 20

[8] Joseph Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–, 1971. 5, 7

[9] Shaz Furniturewala, Kokil Jaidka, and Yashvardhan Sharma. Impact of decoding methods on human alignment of conversational llms. In *Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, WASSA 2024, Bangkok, Thailand, August 15, 2024*, pages 273–279. Association for Computational Linguistics, 2024. 2

[10] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 2

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1

[12] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR*

2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. 2

[13] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. 5, 6

[14] Chao-Chun Hsu, Szu-Min Chen, Ming-Hsun Hsieh, and Lun-Wei Ku. Using inter-sentence diverse beam search to reduce redundancy in visual storytelling. *arXiv preprint arXiv:1805.11867*, 2018. 2

[15] Ting-Hao K. Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Aishwarya Agrawal, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. Visual storytelling. In *15th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, 2016. 5

[16] Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Dragomir Radev, Yejin Choi, and Noah A Smith. A call for clarity in beam search: How it works and when it stops. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 77–90, 2024. 1

[17] Lei Ke, Wenjie Pei, Ruiyu Li, Xiaoyong Shen, and Yu-Wing Tai. Reflective decoding network for image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8888–8897, 2019. 2

[18] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15954–15964, 2023. 2

[19] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39. Association for Computational Linguistics, 2017. 1

[20] Jing Yu Koh, Daniel Fried, and Russ Salakhutdinov. Generating images with multimodal language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 2, 5, 6, 7

[21] Hezheng Lin, Xing Cheng, Xiangyu Wu, and Dong Shen. Cat: Cross attention in vision transformer. In *2022 IEEE international conference on multimedia and expo (ICME)*, pages 1–6. IEEE, 2022. 1

[22] Shaobo Lin, Kun Wang, Xingyu Zeng, and Rui Zhao. Explore the power of synthetic data on few-shot object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*, pages 638–647. IEEE, 2023. 23

[23] Chang Liu, Haoning Wu, Yujie Zhong, Xiaoyun Zhang, Yanfeng Wang, and Weidi Xie. Intelligent grimm - open-ended visual storytelling via latent diffusion models. In *CVPR*, 2024. 2, 3, 5

[24] Tao Liu, Kai Wang, Senmao Li, Joost van de Weijer, Fahad Shahbaz Khan, Shiqi Yang, Yaxing Wang, Jian Yang, and Ming-Ming Cheng. One-prompt-one-story: Free-lunch consistent text-to-image generation using a single prompt. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 5, 6, 7

[25] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. 2024. 2

[26] Yujie Lu, Pan Lu, Zhiyu Chen, Wanrong Zhu, Xin Wang, and William Yang Wang. Multimodal procedural planning via dual text-image prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 10931–10954. Association for Computational Linguistics, 2024. 27

[27] Richard E. Mayer. Multimedia learning. pages 85–139. Academic Press, 2002. 27

[28] Clara Meister, Gian Wiher, and Ryan Cotterell. On decoding strategies for neural text generators. *Trans. Assoc. Comput. Linguistics*, 10:997–1012, 2022. 2

[29] Sachit Menon, Ishan Misra, and Rohit Girdhar. Generating illustrated instructions. In *CVPR*, 2024. 2, 5, 6, 7

[30] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024. 5

[31] Xichen Pan, Pengda Qin, Yuhong Li, Hui Xue, and Wenhu Chen. Synthesizing coherent story with auto-regressive latent diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2920–2930, 2024. 2

[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 4172–4182. IEEE, 2023. 3

[33] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. 2024. 1

[34] Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with localized narratives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 647–664. Springer, 2020. 20

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 5

[36] Tanzila Rahman, Hsin-Ying Lee, Jian Ren, Sergey Tulyakov, Shweta Mahajan, and Leonid Sigal. Make-a-story: Visual memory conditioned consistent story generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2493–2502, 2023. 2, 3

[37] Vasco Ramos, Yonatan Bitton, Michal Yarom, Idan Szpektor, and Joao Magalhaes. Contrastive sequential-diffusion learning: Non-linear and multi-scene instructional video synthesis. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4645–4654, 2025. 1, 2, 3, 4, 5, 6, 20

[38] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, and et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530, 2024. 20

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 3, 6, 26

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 2, 3

[41] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 379–389. The Association for Computational Linguistics, 2015. 1

[42] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 1

[43] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. 2017. 2

[44] Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. A thorough examination of decoding methods in the era of llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 8601–8629. Association for Computational Linguistics, 2024. 2, 8

[45] Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *CoRR*, abs/2501.06848, 2025. 2

[46] Marko Smilevski, Ilija Lalkovski, and Gjorgji Madjarov. Stories for images-in-sequence by using visual and narrative components. In *ICT Innovations 2018. Engineering and Life Sciences: 10th International Conference, ICT Innovations 2018, Ohrid, Macedonia, September 17–19, 2018, Proceedings 10*, pages 148–159. Springer, 2018. 3

[47] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 1

[48] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 1

[49] Tomás Soucek, Dima Damen, Michael Wray, Ivan Laptev, and Josef Sivic. Genhowto: Learning to generate actions and state transformations from instructional videos. In *CVPR*, 2024. 2, 3

[50] Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the DAAM: interpreting stable diffusion using cross attention. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5644–5659. Association for Computational Linguistics, 2023. 20

[51] Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search for improved description of complex scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 2

[52] Zhongqi Wang, Jie Zhang, Shiguang Shan, and Xilin Chen. T2ishield: Defending against backdoors on text-to-image diffusion models. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXV*, pages 107–124. Springer, 2024. 23

[53] Sam Wiseman and Alexander M. Rush. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1296–1306. The Association for Computational Linguistics, 2016. 4

[54] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith

Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. 1

[55] Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. Storydiffusion: Consistent self-attention for long-range image and video generation. *CoRR*, abs/2405.01434, 2024. 2, 5, 6, 7

A. Contextual Scene Descriptions

In many cases, step descriptions alone may lack context as precise visual prompts. These step descriptions are often interdependent, relying on prior steps for contextual understanding, or may miss visual details, e.g. *"Put the ice cream in freezer for 1 hour"*, or actions that involve multiple tasks, such as *"Slice the tomatoes, grate the cheese, and stir the sauce"*. To address this challenge, we leverage Gemini [38], specifically the Gemini 1.5 Flash model, to refine step descriptions into visually detailed prompts while considering sequence context. Specifically, Gemini produces a contextualized caption c_j based on the current step description s_j and all preceding steps as follows:

$$c_j = \phi(s_j | \{s_1, s_2, \ldots, s_{j-1}\}) \quad (6)$$

Following this approach we can improve image generation prompts by maintaining consistency and preserving the dependencies between sequential steps.

B. Impact of Latent Selection on Beam Search Performance

Latent selection plays a crucial role in determining the quality and coherence of image sequences generated through beam search. By analyzing how different latent groups contribute to generation performance, we can better understand their impact on semantic alignment and visual coherence.

This aligns with recent work on interpreting diffusion models. For example, Pont-Tuset et al. [34] ground textual descriptions in image regions, while Tang et al. [50] and Dewan et al. [7] explore how textual prompts influence generation through attention analysis and partial information decomposition. In contrast, our method contributes to interpretability by examining how latent space traversal, guided by beam search, affects the generation of coherent, semantically aligned image sequences.

To systematically investigate this, we evaluate several sequences, each consisting of four steps, allowing us to assess the effects of latent selection across a diverse set of sequences. Using both automatic evaluation and human annotations, we analyze different latent configurations to determine which groups contribute most to generating coherent and semantically meaningful sequences.

B.1. Impact of Latent Groups on Performance

Annotations. Following the initial analysis of beam search configurations (Sec. 5.4), where we used latent indexes 0, 1, 2, and 3, we further investigated the impact of using different latent configurations on the generated sequences. While latents 0, 1, 2, and 3 show promising results, latents 1, 3, 5, and 7 exhibit significantly poorer performance. In particular, beam search configurations for these underperforming latents result in very limited selection rates, indicating they are rarely chosen in both human and Gemini annotations. For human annotations across 6 tasks, the only configuration that yielded any selection was the one with 4 steps back and a beam width of 3, achieving a selection rate of 8.33%. All other configurations resulted in 0% selection. In contrast, none of the configurations showed any non-zero selection rate under Gemini annotations, reinforcing the lack of preference for these latents.

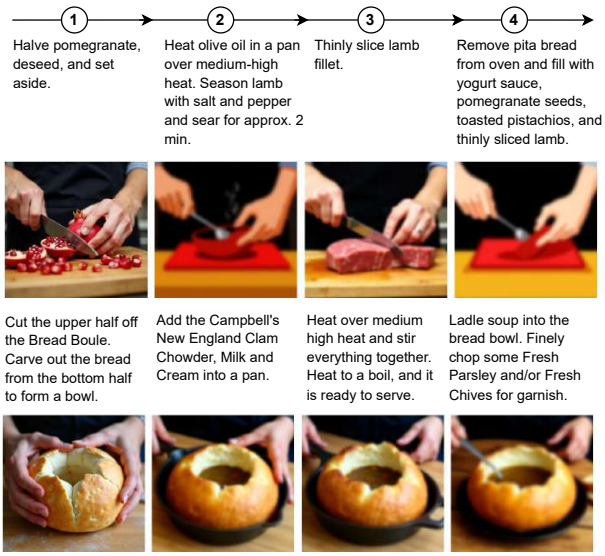


Figure 4. Impact of using late-index latents configurations on sequences generation.

Similarly, for Gemini annotations evaluated on the same 6 sequences, the configurations do not yield any notable results, with no selection rates. When the evaluation is extended to 35 sequences, the performance for latents 1, 3, 5, and 7 remains sparse, with a single configuration (2 steps back, beam width of 3) being selected, with a selection rate of 2.86%. Figure 4 visually demonstrates the poor results, where the images appear nearly identical, further supporting the findings of limited diversity and low performance, which aligns with the findings of [3, 37].

B.2. Latent Selection on Image Sequence Generation

Following our evaluation of the impact of latent groups on performance, we now examine the role of individual latents within the higher-performing group (Latents 0-3). As shown in Figure 5, Latent 3 is selected most frequently (44.4%), followed by Latent 0 (20%), while Latents 1 and 2 are used less often (11.1% and 8.9%, respectively). This uneven distribution suggests that Latents 0 and 3 contribute disproportionately to the generation of sequences with better text alignment. One possible explanation is that these latents encode more semantically relevant or visually discriminative

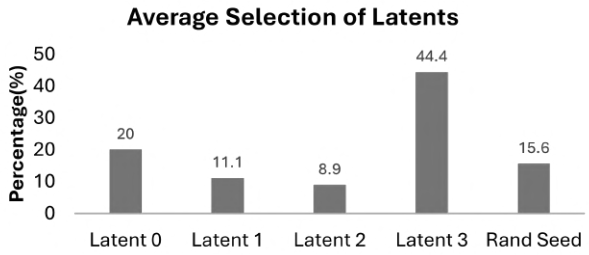


Figure 5. The average number of times that the model selected a given **latent** to generate the next image.

features that align well with the text prompts. Their frequent selection by beam search indicates that these latents are particularly effective at resolving ambiguities or reinforcing important visual cues during sequence generation. The lower usage of Latents 1 and 2 may reflect their tendency to encode less salient or redundant information, which contributes less to text-conditioned coherence. Interestingly, the Rand seed, which injects stochasticity into the generation process, accounts for 15.6% of selections. This suggests that while beam search primarily exploits strong latents like 0 and 3, it also values diversity. The inclusion of randomized elements allows the search to escape local optima and sample from alternative yet plausible paths, enhancing the robustness of the final sequence.

B.3. Effect of Cumulative Latent Access

To investigate how access to latents from previous steps influences the quality of generation, we conducted an ablation study in which we constrain the set of available latents throughout the generation of the sequence. Specifically, we evaluate performance when the model is limited to latents from only the first step, then from the first two steps, and so on, up to including latents from all previous steps (i.e., full cumulative access).

As shown in Figure 6, including latents from only the first step makes step 4 appear very closed up. Limiting the model to latents from the first two steps makes it more difficult to follow the step text while maintaining coherence, which is particularly evident in step 5. Gradually expanding the latent set to include more steps improves sequence quality, especially when moving from two to three steps. However, including latents from all previous steps does not yield further improvements in coherence, suggesting diminishing returns with increasing temporal access.

In contrast, BeamDiffusion uses latents from only the two most recent steps and achieves results that are comparable, or even superior, to the full cumulative case. Each latent carries information from its own step as well as the previous steps it incorporated: during generation, a latent from a previous step is updated with the current step, effectively propagating

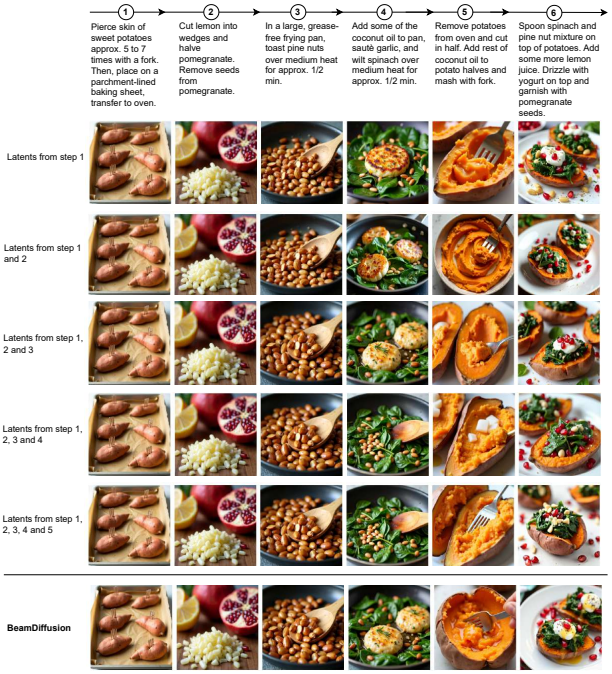


Figure 6. Visual comparison of sequences generated with cumulative latent access. Each row shows a sequence generated using latents from only the first step, the first two steps, and so on, up to all previous steps. The bottom row shows the result from BeamDiffusion, which uses latents from only the two preceding steps.

all past information forward (see Figure 1b for the latent chain generation). By always using the last two steps latents, BeamDiffusion maintains a summary of the entire sequence without needing full access to all prior steps, preserving strong alignment and visual consistency while significantly reducing computational overhead.

Overall, this experiment highlights that cumulative latent access enhances generation quality up to a point, but full historical memory is unnecessary. Strategies such as BeamDiffusion, which leverage a limited yet informative temporal window, can produce high-quality sequences efficiently. These findings also underscore the importance of the non-linear sequence denoising strategy introduced in Section 5.5, which enables more targeted and effective reuse of past latents.

B.4. Automatic and Qualitative Evaluation of Decoding Strategies

Automatic evaluation. Table 6 reports automatic comparisons across similarity metrics and faithfulness/consistency measures. The combined scores CLIP* and DINO* offer a more holistic signal than their individual components. Here, BeamDiffusion achieves the best CLIP* score (40.00) and matches the highest DINO* (30.00), indicating strong performance across both text–image and image–image similar-

Table 6. Comparison of different decoding methods across embedding-based sequence quality metrics. CLIP-I, DINO-I, CLIP-T, CLIP*, and DINO* are reported as win percentages, while Goal Faithfulness, Step Faithfulness, and Cross-Image Consistency are reported as absolute scores.

Method	CLIP-I	DINO-I	CLIP-T	CLIP*	DINO*	Goal Faith.↑	Step Faith.↑	Cross-Image Cons.↑
Greedy/CoSeD$_{len=1}$	30.00	30.00	35.00	25.00	30.00	0.76	0.90	**0.62**
Nucleus Sampling	25.00	**40.00**	**40.00**	35.00	**35.00**	0.72	0.80	0.60
BeamDiffusion	**45.00**	30.00	25.00	**40.00**	**35.00**	**0.82**	**0.92**	0.61

Figure 7. Illustrative comparison of decoding approaches. BeamDiffusion prevents cascading inconsistencies across steps while preserving semantic and visual consistency.

ity. By contrast, Greedy/CoSeD lags significantly on CLIP* (25.00) and only reaches 30.00 on DINO*, while Nucleus Sampling improves slightly on CLIP* (35.00) and DINO* (35.00) but at the expense of weaker instruction alignment and coherence.

BeamDiffusion also obtains the strongest scores on goal faithfulness (0.82) and step faithfulness (0.92), showing that its generations remain closely aligned with both the overall task and individual step instructions. Greedy/CoSeD achieves reasonably high step faithfulness (0.90) and the best cross-image consistency (0.62), but falls behind in goal alignment. Nucleus Sampling, in contrast, is weaker across all three measures (0.72, 0.80, 0.60). Taken together, the similarity metrics and the faithfulness/consistency measures highlight that BeamDiffusion avoids the sharp trade-offs of the other decoding methods, achieving a favorable balance between multimodal similarity, instruction following, and visual coherence.

Qualitative evaluation. Figures 7 and 8 illustrate characteristic error modes of baseline strategies. Nucleus Sampling often suffers from blur propagation, where low-frequency artifacts introduced in one step persist across subsequent generations, degrading the sequence. Greedy decoding avoids blur but instead produces visual inconsistencies, with objects abruptly shifting style or geometry across steps (e.g., the changing appearance of the pan between step 1 and step 3, in Figure 7).

In contrast, BeamDiffusion avoids both pitfalls. By dynamically selecting informative latent trajectories, it prevents error accumulation while preserving smooth semantic and visual transitions. The generated sequences remain sharper, semantically faithful, and more coherent over time. This demonstrates that decoding is not merely a technical detail but a decisive factor in determining the fidelity and consistency of long-horizon generations.

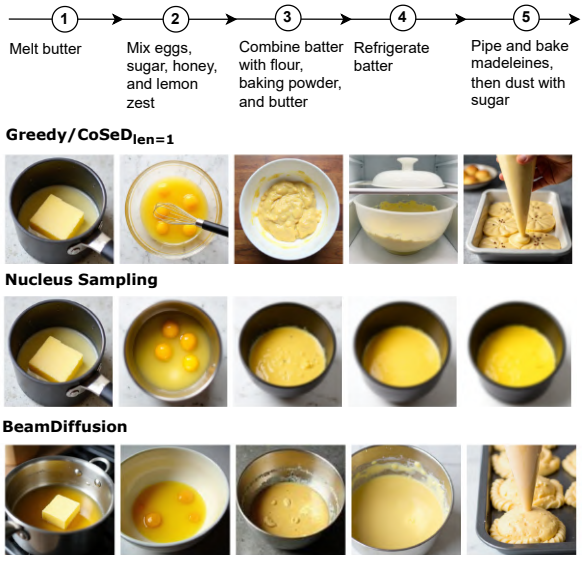


Figure 8. Illustrative comparison of decoding approaches. BeamDiffusion prevents cascading inconsistencies across steps while preserving semantic and visual consistency.

Summary. Overall, both automatic and qualitative results show that BeamDiffusion delivers a principled improvement

over standard strategies. By explicitly managing latent trajectories, it achieves robustness against error propagation and inconsistency while preserving semantic faithfulness and cross-step coherence.

C. Minimizing Decoding Risk

The generation process using diffusion models begins with a seed that determines the initial conditions of the model. The choice of this seed significantly affects the outcome, as it influences the path the model takes during generation. Using multiple random seeds in the first step ensures a diverse exploration of possible outputs, avoiding premature narrowing of options into a single path based on that seed. This limits the diversity of the results and increases the chance of missing better options.

Greedy Decoding and Nucleus Sampling. In both greedy decoding and nucleus sampling, we use the same approach in the first step. Multiple random seeds are employed to generate a set of candidate images. Each candidate is then evaluated using the CLIP score mechanism, which compares the generated images with the step text. The image that best aligns with the text, according to the CLIP score, is selected to continue the process. This method ensures that both greedy decoding and nucleus sampling benefit from multiple diverse starting points, increasing the quality and variety of the generated outputs.

BeamDiffusion. In BeamDiffusion, relying on a single random seed in the first step effectively makes the process a greedy selection, as it locks the generation into a specific path. This limits diversity and reduces the chances of discovering higher-quality outputs. By using multiple random seeds, each seed generates a different decoding trajectory, helping to broaden the search space and increasing the likelihood of finding better solutions. Using multiple random seeds in the initial step enhances the exploration of the latent space, ensuring better and more diverse results across different decoding strategies.

Random Seeds for mid-sequence Generation. The sequence of scenes $S = \{s_1, s_2, \ldots, s_L\}$ may have one scene in the middle of the sequence that should be independent. To address this, for steps $s_{j>1}$, we use a combination of random seeds and past latent vectors to maintain a balance between diversity and coherence, which is particularly important for such scenarios.

D. Metrics and Annotations

Automatic Metrics. In image sequence generation, CLIP similarity scores between visually similar frames often exceed 0.85, even for near-duplicate content. Wang et al. [52]

Figure 9. Illustration of the limitation of CLIP-I: despite a high CLIP-I score, the result is not semantically correct or visually faithful. The example compares several automatic metrics (CLIP-T, CLIP-I, DINO-I, CLIP*, DINO*), highlighting discrepancies and failure cases in sequence evaluation.

highlight that such high CLIP-I scores can indicate semantic duplication, which poses a challenge when diversity across frames is desired. While CLIP-I and DINO-I are common proxies for semantic similarity, rewarding near-duplicates undermines meaningful visual progression. Conversely, poor alignment between images and their textual prompts is reflected in low CLIP-T scores. Lin et al. [22] address this by clipping CLIP-T values below 0.1, effectively filtering out weak text-image matches.

To balance these factors, we set CLIP-I scores above 0.9 and DINO-I scores above 0.85 to zero, penalizing visual redundancy. Similarly, we zero out CLIP-T scores below 0.1 to discourage poor alignment with the prompt. This clipping strategy encourages the generation of sequences that are both semantically relevant and visually diverse. In Figure 9, we present the automatic metric scores for a sequence across all methods. The results illustrate that very high CLIP-I values can lead to sequences that fail to follow the textual

steps accurately, as observed in the case of GILL. Despite achieving strong visual similarity, the sequence diverges from the intended narrative, highlighting a key limitation of relying solely on CLIP-I . This example reinforces the importance of our clipping strategy: without penalizing excessively high CLIP-I values, the generation process may favor near-duplicates over meaningful visual progression, ultimately undermining semantic fidelity to the prompt.

Even after applying this clipping, we observe that automatic metrics remain imperfectly aligned with human and Gemini evaluations, underscoring the ongoing challenge of developing reliable metrics for evaluating sequences of images.

Human annotations. To facilitate the selection of the best beam search configuration, we developed a custom website specifically for human annotation. We found that existing tools lacked the flexibility necessary for our specific use case, especially when evaluating multiple configurations simultaneously. To select the best beam search configuration, 5 annotators are presented with 24 different sequences, each corresponding to a unique beam search configuration.

In total, the 24 configurations were generated by combining three key factors:

- **Steps Back** (m in Eq. 2): This refers to how many previous steps the model uses from its latent representations to explore new candidate outputs. We evaluated three different settings for the number of steps back: 2, 3, and 4, which correspond to using latents from the last two, three, or four images, respectively.
- **Latents Indexes**: We tested two sets of latent configurations, 1, 3, 5, 7 and 0, 1, 2, 3, by selecting specific indexes from the latent representations generated in previous stages. This was done to explore how varying levels of latent detail impact the quality of the generated sequences.
- **Beam Width** (w in Eq. 5): We explored four different beam widths, 2, 3, 4 and 6, which determine how many beams are retained at each step during the search process.

Evaluating all of these at once proved to be exhausting and difficult to manage. To simplify the evaluation process, we adopted a elimination annotation method. In this approach, 5 annotators compare two configurations at a time and select the one with better overall coherence. If neither configuration stands out, annotators can select *"both good"* or *"both bad"* (Figure 10), in which case the computationally cheaper option advances to the next round. This process is repeated until we identify the best configuration. This method focuses on direct comparisons, making it easier to pinpoint the most effective beam search settings while also considering computational efficiency.

To evaluate all models performance, annotators are presented with five generated sequences side by side. Figure 11 illustrates an example of this comparison process, where the annotators assess the sequences to determine the most effective approach.

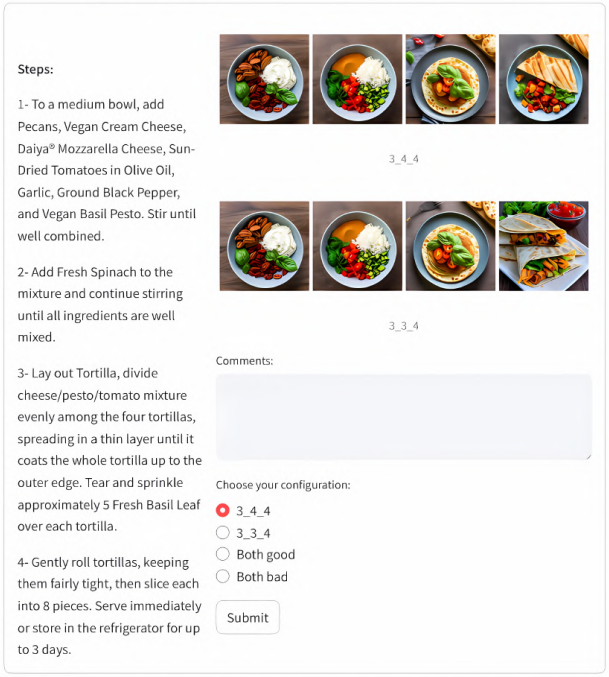


Figure 10. Example of the human annotation elimination round for selecting the best beam search configuration.

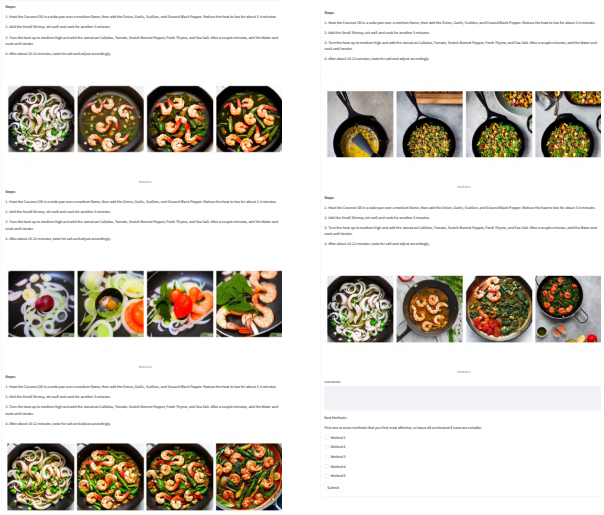


Figure 11. Example of a human annotation to choose the best method.

Gemini evaluation. Gemini's selection process for annotating the best beam search configuration and the best generation method closely mirrored that of human annotators,

Gemini Prompt:
Evaluate two image sequences using the following criteria:
- **Visual Consistency**: Objects, ingredients, backgrounds, colors, and styles should remain consistent across frames. Frames should transition smoothly, resembling a video-like progression with no sudden changes or missing elements.
- **Semantic Consistency**: Images must logically follow the described steps, clearly and accurately.

Steps of the process: {steps}
After evaluating both sequences, select the one with the strongest overall consistency (visual and semantic). Pay close attention to even minimal differences, as the sequences are very similar. If both sequences are equally good or bad, choose the one that is slightly better. Choose exactly one of the following options:
- "First sequence"
- "Second sequence"

Figure 12. Gemini prompt used to select the best beam search configuration and best method.

with one key adjustment: Gemini tends to default to *"both good"* or *"both bad"* when such options are available. To ensure more decisive and accurate results, we removed these options from the prompt. We also refined our comparison methodology. Initially, presenting all five sequences at once led to biased evaluations, as the model disproportionately favored the last two sequences. To mitigate this, we adopted a pairwise evaluation strategy, comparing two sequences at a time. This approach resulted in more balanced and reliable assessments across all comparisons.

Figure 12 shows the prompt used in Gemini to guide the evaluation of image sequences based on visual and semantic consistency.

E. Datasets

Recipes. Each manual task in the Recipes datasets includes a title, a description, a list of ingredients, resources and tools, and a sequence of step-by-step instructions, which may or may not be illustrated. The Recipes dataset comprises approximately 1,400 tasks, with an average of 4.9 steps per task, totaling 6,860 individual steps. Most tasks feature an image for each step.

VIST. We used the Descriptions of Images-in-Isolation (DII) annotations, as they allow for the generation of more precise and informative visual prompts. These annotations provide standalone descriptions of images, making it easier to capture their key visual elements without relying on surrounding context. The dataset includes 50 000 stories, offering a diverse range of visual and textual information. To ensure the most accurate and relevant annotation for each image, we leveraged CLIP. By using CLIP, we were able to select the annotation that best represented the image, improving the quality of our generated prompts and enhancing the overall alignment between text and visuals.

F. Challenges

In the development and evaluation of the BeamDiffusion method, we encountered several challenges that required careful consideration and innovation. These challenges spanned multiple aspects of the model, ranging from the difficulty of generating accurate and concise captions, to identifying suitable domains for testing, and addressing the computational complexity of the generation process. While we were able to propose solutions for some of these challenges, others remain open and present ongoing areas of research. In this section, we outline the key challenges we faced and the strategies we employed, while acknowledging that not all of them have been fully addressed.

Captions. One of the challenges encountered in this process is generating captions that accurately describe the steps involved in the sequence. Often, a single step may consist of multiple sub-steps, each contributing to the overall generation. This makes it difficult to write a single concise caption that captures the full scope of each step. The challenge lies in ensuring that the caption not only reflects the specific action or transformation occurring at each step but also conveys the progression and complexity of the entire sequence. Additionally, if we divide the steps into smaller sub-steps to better describe the process, the sequence becomes very extensive, making it even harder to maintain clarity and readability while ensuring comprehensive coverage of the entire generation process.

Domains. Identifying a domain that effectively highlights the advantages of the BeamDiffusion method, particularly the benefits of beam search, is still a challenge. While BeamDiffusion provides improvements in consistency and coherence, not all domains clearly showcase these advantages. Certain domains may lack the complexity or diversity needed to demonstrate the method's strengths, such as its adaptive latent selection or its ability to maintain alignment with contextual instructions. Finding a domain that not only aligns with the method's capabilities but also provides a clear and compelling demonstration of its improvements remains an ongoing challenge.

Complexity of the generation. The BeamDiffusion model generates multiple candidate images at each step by exploring a range of latents. We analyze the complexity of the model for a steps back of m steps and N latents explored. In the initial steps, the complexity is high because pruning begins only after the second step. For each beam, $N \cdot m$ latents are explored, and the search space grows exponentially, resulting in significant computational cost in the early stages. During the first two steps, the complexity grows with the number of candidates and latents, proportional to

$\mathcal{O}(N \cdot m)^j)$ for $j \leqslant 2$. Specifically, for $j = 1$, the complexity is proportional to the number of explored random seeds. After the second step, pruning reduces the number of candidates to the most promising w beams, decreasing the complexity to $\mathcal{O}(N \cdot m \cdot w)$.

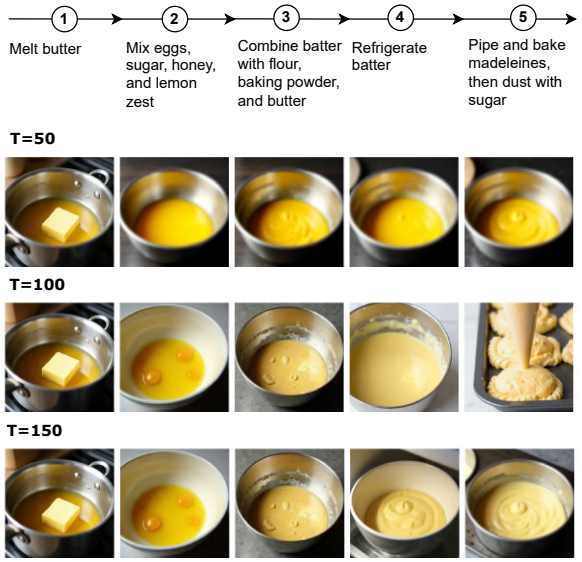
To further reduce complexity, a heuristic based on text relevance can be used to predict the top P most relevant past images. By restricting the search space to the latents of these top P candidates, the model improves efficiency by prioritizing contextually relevant images, particularly in the early steps. While the initial three steps remain computationally demanding due to the expansive search space, pruning combined with this heuristic can significantly lower the computational cost by refining the candidate set.

G. Implementation Details

Base Models. Our method is primarily implemented on top of FLUX.1-dev[2], a diffusion transformer–based model released by Black Forest Labs. We extend its open-source implementation to support beam-based generation, latent trajectory tracking, and contrastive scoring mechanisms as described in the main text. Unless otherwise specified, we follow the default configuration of FLUX.1-dev with 100 denoising steps, chosen to balance generation quality and computational efficiency. While the default 50 steps generally produce globally coherent images, we observe that finer structures often remain underdeveloped, resulting in blur and reduced visual consistency across sequences. Increasing to 100 steps substantially sharpens details and improves sequence coherence, as illustrated in Figures 13a and 13b. Further increasing to 150 steps provides no additional gains, confirming that 100 steps is a reasonable trade-off. To ensure that the choice of decoding method is not dependent on a specific backbone, we also implement BeamDiffusion on Stable Diffusion v2.1[39], a widely used U-Net–based latent diffusion model, using its standard parameters. This setup verifies that our conclusions regarding decoding strategies generalize across different generative architectures.

Latent Trajectory Reuse. In our approach, maintaining coherence across sequential generations requires access to the intermediate latent representations produced during the generative process. Fortunately, the model pipeline inherently provides the full sequence of latent representations at each generation step. This enables us to easily extract and reuse specific latents as seed for new generations.

Classifier Training. To train the contextual classifier, we employ a sequence-aware training approach that enables the model to capture dependencies between steps in the generation process. The classifier φ is primarily trained on the Recipes dataset, which provides rich sequential image-text pairs of cooking steps, allowing it to learn meaningful context transitions in a structured procedural domain. Additionally, a small amount of data from the VIST dataset is included during training to introduce limited variability in narrative structure. The classifier takes as input a candidate image I_l, its textual description s_l, and the previous sequence images and respective step text ($B_{l-1}^{(b)}$). It then



(a) Example 1: 50-step generations exhibit blur propagation, while 100 steps reduce blur and sharpen frames. 150 steps show no substantial gain over 100.

(b) Example 2: 50 steps produce blurry, inconsistent frames, while 100 steps improve fidelity and coherence. 150 steps do not yield further noticeable improvements.

Figure 13. Qualitative comparison of 50 vs. 100 denoising steps across two examples. Increasing steps to 150 does not produce substantial gains over 100, indicating diminishing returns.

scores I_l based on how well it fits within the evolving trajectory.

During training, we use labels $l_{d,l}$ indicating whether I_l is the correct continuation in the sequence for task d and step l. The model is trained to minimize the cross-entropy loss:

$$\text{argmin} \sum_{d=1}^{D} \sum_{l=1}^{L} l_{d,l} \cdot \log(\varphi(I_l, s_l, B_{l-1}^{(b)})), \quad (7)$$

This contrastive training encourages the model to assign high scores to contextually appropriate continuations while penalizing incoherent or semantically unrelated ones. As a result, the classifier learns to distinguish subtle differences in step-to-step coherence, both visually and textually.

Hardware. All experiments were conducted on a NVIDIA A100 GPU with 40GB of memory.

H. Broader Impacts

This work proposes a method to improve generative pretraining by incorporating subject-aware mutual information maximization, aiming to produce more contextually aligned and coherent sequences of images. The ability to generate visually consistent image sequences has positive applications in areas such as animation, visual storytelling, educational media, and accessibility tools (e.g., for generating illustrative content from text). However, as with many generative models, there are potential negative societal impacts. Enhanced coherence and subject fidelity could be misused to create misleading or deceptive visual content, such as more convincing visual disinformation, propaganda, or synthetic media (e.g., deepfakes).

I. Application Examples

This section presents additional visual examples and qualitative results that complement the main paper. Images and other visual aids are essential tools that significantly improve comprehension across a variety of domains. Whether breaking down complex tasks, illustrating a sequence of events, or clarifying intricate concepts, visual representations provide clear, step-by-step guidance that enhances understanding [27]. From technical procedures and instructional design to storytelling and educational content, multi-scene image sequences help readers grasp information more intuitively, transforming even the most complex or dense topics into something more accessible. However, generating a sequence of images that not only aligns with each textual instruction but also maintains overall coherence remains a major challenge [26]. In this appendix, we include expanded examples to further illustrate how our model addresses this issue. These include additional visualizations of the beam search process and extended qualitative comparisons across domains such as recipes and visual storytelling. Together, these examples highlight the strengths and limitations of current approaches in achieving textual alignment and visual consistency in sequential image generation.

I.1. Sequence Examples

To provide a comprehensive comparison between all models, we present several examples across different domains in Figures 14-17, highlighting how each method maintains sequence coherence in sequential image generation.

For the Recipes domain (*"Buttermilk chicken wings"*, Figure 14), most models incorrectly depict the chicken as already cooked in step 2, instead of starting raw and gradually appearing cooked. BeamDiffusion correctly captures this progression, faithfully reflecting the cooking process across all steps. Similarly, in the Recipes task *"Sweet potato and orange soup"* (Figure 15), GILL, StackedDiffusion, 1P1S, and StoryDiffusion misrepresent the ingredient being cut, while BeamDiffusion consistently depicts a person cutting a sweet potato and maintains stepwise visual consistency without hallucinations.

For the VIST stories, BeamDiffusion demonstrates strong identity and style preservation. In *"4th of July 2006"* (Figure 16), it is the only method to maintain the same child between step 2 and step 4. In *"Yosemite Weekend"* (Figure 17), both BeamDiffusion variants maintain coherent tree styles across steps, whereas other models show a noticeable break between step 1 and step 2.

Overall, while baseline methods may partially adhere to textual instructions or maintain visual similarity, BeamDiffusion consistently balances stepwise textual alignment and visual coherence, producing sequences that are both contextually accurate and visually stabl

Figure 14. Example of the Recipes domain task *"Buttermilk chicken wings"*, with a sequence of 4 steps.

Figure 15. Example of the Recipes domain task *"Sweet potato and orange soup"*, with a sequence of 4 steps.

Figure 16. Example of the out-of-domain VIST story *"4th of July 2006"*, with a sequence of 5 steps.

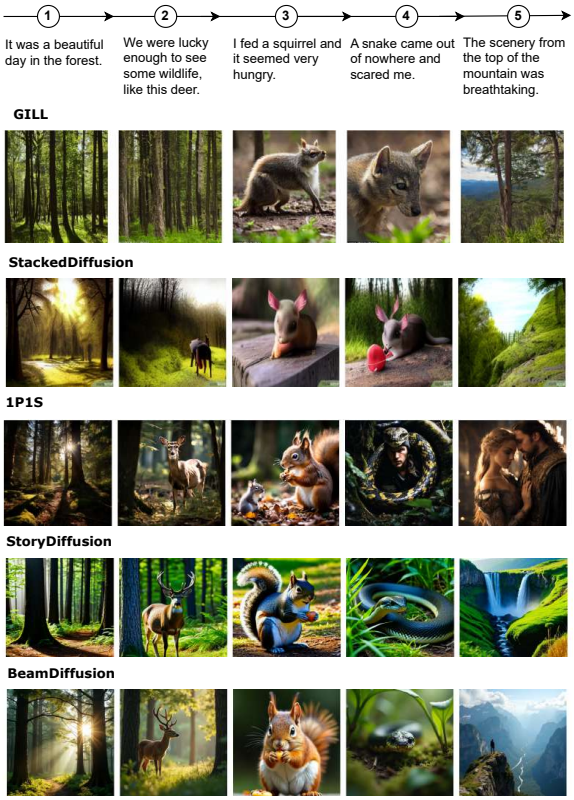


Figure 17. Example of the out-of-domain VIST story *"Yosemite Weekend"*, with a sequence of 5 steps.