Discovering Latent Graphs with GFlowNets for Diverse Conditional Image Generation

Bailey Trang[1], **Parham Saremi**[4,5], **Alan Q. Wang**[2],
Fangrui Huang[1], **Zahra TehraniNasab**[4,5], **Amar Kumar**[4,5],
Tal Arbel[4,5], **Li Fei-Fei**[1], **Ehsan Adeli**[1,2,3 *]
[1]Dept. of Computer Science, Stanford University, Stanford, CA, USA
[2]Dept. of Psychiatry and Behavioral Sciences, Stanford University, Stanford, CA, USA
[3]Dept. of Biomedical Data Science, Stanford University, Stanford, CA, USA
[4]Center for Intelligent Machines, McGill University, Montreal, QC, Canada
[5]MILA - Quebec AI institute, Montreal, QC, Canada
{trangn, alanqw, fangruih, feifeili, eadeli}@stanford.edu
{parham.saremi, zahra.tehraninasab, amar.kumar}@mail.mcgill.ca
tal.arbel@mcgill.ca

Abstract

Capturing diversity is crucial in conditional and prompt-based image generation, particularly when conditions contain uncertainty that can lead to multiple plausible outputs. To generate diverse images reflecting this diversity, traditional methods often modify random seeds, making it difficult to discern meaningful differences between samples, or diversify the input prompt, which is limited in verbally interpretable diversity. We propose Rainbow, a novel conditional image generation framework, applicable to any pretrained conditional generative model, that addresses inherent condition/prompt uncertainty and generates diverse plausible images. Rainbow is based on a simple yet effective idea: decomposing the input condition into diverse latent representations, each capturing an aspect of the uncertainty and generating a distinct image. First, we integrate a latent graph, parameterized by Generative Flow Networks (GFlowNets), into the prompt representation computation. Second, leveraging GFlowNets' advanced graph sampling capabilities to capture uncertainty and output diverse trajectories over the graph, we produce multiple trajectories that collectively represent the input condition, leading to diverse condition representations and corresponding output images. Evaluations on natural image and medical image datasets demonstrate Rainbow's improvement in both diversity and fidelity across image synthesis, image generation, and counterfactual generation tasks.

1 Introduction

Conditional image generation produces novel images that adhere to given input prompts or conditions[2] like text [32, 75]. In real-world scenarios, an input prompt has inherent ambiguity, which may correspond to multiple plausible output images [7, 40, 76, 85], especially when prompts are abstract, high-level information. For example, an input text prompt describing a "sunset scene" could map to many valid images, differing in factors such as season, light control, and overall ambiance. Similarly, in medical imaging, brain magnetic resonance images (MRIs) of two patients of the same age and gender may nevertheless display variability in the structure of brain regions and patterns of intensity

*Corresponding author.
[2]We refer to "prompts" and "conditions" interchangeably.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

despite having identical conditions due to subject-specific and medical scanner-specific details. In both cases, failing to address inherent uncertainty and capturing diversity in generative models can lead to suboptimal decision-making, misinterpretations, and generation collapse, where limited and uniform outputs fail to represent the necessary variability [13, 18, 38, 39, 58, 73].

Previous attempts at generating diverse images in conditional image generation models, such as using GANs [19], diffusion models [10, 24], and latent diffusion models [62] (LDMs), can be categorized into two main approaches: (1) Traditional methods typically rely on randomness; for example, repeating the generation process with different random seeds or varying the random noise on the same seeds in diffusion models [25, 34, 49, 51] to create multiple outputs. While these methods can produce non-identical images, they often fail to capture true diversity of choices and may exhibit inherent biases; (2) Another line of work involves diversifying and adding details to the input prompt verbally using a pretrained Large Language Model (LLM) such as ChatGPT [22, 60, 82]. Although this approach can enhance the richness of the generated content, it is confined to text-based conditions and relies on external LLM models and their own biases. Consequently, these strategies may not adequately address the inherent uncertainty of conditional image generation tasks. In addition, a more versatile approach is needed to handle multiple condition types. For instance, generating medical images conditioned on age, sex, diseases, or other medical details can enrich datasets in fields where data collection is costly and time-consuming, such as in 3D brain MRI or chest X-ray datasets.

Addressing these limitations, we introduce Rainbow, a novel conditional image generation framework designed to produce diverse and plausible images. Rainbow can be integrated into any pretrained conditional image generative model. The primary idea is to create multiple images simultaneously that capture uncertainty by collectively reflecting the input condition. To achieve this, we aim to generate diverse condition representations that encapsulate various aspects of the uncertainty inherent in the input condition within the latent space. Each representation produces a distinct output image while the pretrained generative models remain frozen or minimally modified. As a result, Rainbow delivers a range of high-quality images that comprehensively interpret the input prompt.

To achieve diverse condition latent representations that collectively reflect the input condition, we first construct a graph structure, called the *latent graph*, within the latent representation computation. Next, we utilize Generative Flow Networks (GFlowNets) [4, 5] to sample diverse trajectories over the graph collectively representing the input condition. Specifically, GFlowNets are designed to capture uncertainty in tasks with multiple possible outputs (multiple modes) by sampling diverse high-quality intermediate representations (e.g., trajectories over a graph) that lead to varied outputs, each representing one possible optimal outcome (one mode of the solution space). GFlowNets have been applied to many contexts, including molecule generation [4], gene regulatory networks [3, 48], and dropout masks [42]. In Rainbow, trajectories generated by GFlowNets collectively capture diverse interpretations of the input condition, leading to diverse condition latent representations, which are subsequently used to produce diverse output images.

Our contributions include:

- First, we introduce Rainbow, a novel conditional image generation framework that captures uncertainty and produces diverse images.
- Second, by discovering the diversity in condition latent representations, Rainbow is applicable to any pretrained conditional generative model, regardless of the condition type, and addresses the limitation of relying on randomness during generation.
- Third, our experiments on text- and non-text-based conditions across natural images and medical images (brain MRIs and chest X-rays) demonstrate Rainbow's improved capacity to capture uncertainty, generate diverse and plausible images, and benefit downstream tasks.

2 Preliminary

2.1 Generative Flow Networks (GFlowNets)

GFlowNets is a probabilistic model that samples diverse high-quality objects *i.e.* diverse trajectories of node/edge through a graph, where the likelihood of generating a trajectory x is proportional to an unnormalized probability or reward $R(x) = e^{-\mathbb{E}(x)}$, with $\mathbb{E}$ denoting the expectation of some quantity of interest associated with x [4, 5, 44]. GFlowNets samples a sequence of actions that modify

a compositional trajectory (*i.e.*, adding one edge to a trajectory), starting from a universal initial state and continues through successive modifications dictated by a trainable policy until it reaches a terminal state or achieves a specific graph sparsity. This policy is trained so that the probability of terminating the trajectory x at a particular final state is proportional to the reward $R(x)$.

Specifically, GFlowNets operate on a graph $G = (S, A)$, where S is the set of states and A is the set of actions (transitions). The objective is to model a nonnegative flow $F : A \to \mathbb{R}_{\geq 0}$, which defines the unnormalized likelihood of taking action to transform from state s to s' [44]. To ensure correct sampling, the flow F needs to satisfy certain constraints, such as the flow matching constraints [5].

Flow Matching Constraints is the core principle of GFlowNets, which enforces that for any intermediate states, the incoming flow equals the outgoing flow. For any state s, the state flow $F(s)$ is defined as the total flow through state s, and the edge flow $F(s' \to s)$ is the flow along transitions $s' \to s$; subsequently, the flow matching constraints is formulated as $F(s) = \sum_{(s' \to s) \in A} F(s' \to s) = \sum_{(s \to s'') \in A} F(s \to s'')$. The goal is to train the GFlowNets model so that the state flow at any terminal state s_T that obtains object x is proportional to the reward $F(s_T) \propto R(x)$.

2.1.1 Detailed Balance Objective

Detailed Balance Objective [5] (DBO) is one of the training objectives for GFlowNets, along with other approaches such as *flow matching* [5] and *trajectory balance objectives* [44]. DBO enforces consistency between forward and backward transitions while aligning terminal states with a reward function $R(x)$. Let s_i denote the state at step i, where s_0 is the initial state and s_n is the terminal state that yields object x after n steps; DBO defines the *forward policy* $P_F(s_i|s_{i-1};\theta)$ parameterized by θ is the probability of transitioning to state s_i from s_{i-1}, while the *backward policy* $P_B(s_{i-1}|s_i;\theta)$ models the reverse transition; The *state flow* $F_\theta(s)$, a scalar function, estimates the unnormalized likelihood of passing through state s. Subsequently, the DB loss combines two critical terms:

$$\mathcal{L}_{\text{DB}}(x, R(x)) = \sum_{i=1}^{n-1} \left(\log \frac{F_\theta(s_{i-1}) P_F(s_i|s_{i-1};\theta)}{F_\theta(s_i) P_B(s_{i-1}|s_i;\theta)} \right)^2 + \left(\log \frac{F_\theta(s_{n-1}) P_F(s_n|s_{n-1};\theta)}{R(x)} \right)^2. \quad (1)$$

Specifically, the first term ensures conservation of flow between consecutive states s_{i-1} and s_i. By minimizing the squared log-ratio of forward and backward transitions, the loss enforces that the forward probability to transform s_{i-1} to s_i equals the backward probability to revert s_i back to s_{i-1}. In addition, by weighting with their respective flows, this term guarantees that the net flow through any state transition is balanced. The second term aligns the final state s_n's flow with reward $R(x)$.

2.1.2 Capturing Diversity with GFlowNets

GFlowNets promote diversity through three interconnected mechanisms grounded in their flow-matching constraint foundation. First, the terminal state flow alignment $F(s_T) \propto R(x)$ enforces proportional sampling where candidates x (associated with terminal states s_T) are generated with probability $p(x) = \frac{F(s_T)}{Z} \propto R(x)$, preserving all reward modes unlike reinforcement learning's $\arg\max R(x)$ objective [4]. Second, the flow conservation constraint ensures global balance: at every non-terminal state $s \in S$, incoming flows from predecessors equal outgoing flows to successors, preventing preferential routing to dominant modes while maintaining non-zero probability for all viable paths [5]. Finally, the stochastic forward policy $P_F(s''|s;\theta) = \frac{F(s \to s'')}{F(s)}$, derived from normalized edge flows $F(s \to s'')$, enables amortized trajectory generation. Unlike Markov Chain Monte Carlo's (MCMC) local random walks, this allows direct jumps between distant modes (e.g., structurally distinct molecular graphs with comparable $R(x)$) through single-pass sampling via P_F [4], bypassing MCMC's iterative transitions [44]. Together, these mechanisms ensure diverse high-reward candidates are sampled proportionally to their rewards while preserving exploration capacity across disconnected regions of the solution space.

2.2 Latent Diffusion Models

Training an LDM consists of two stages. In the first stage, an autoencoder is trained which learns to map each image $\mathbf{X}$ to a lower-dimensional latent embedding $\mathbf{z}$. Let $\mathcal{E}_I$ and $\mathcal{D}_I$ denote the image encoder and decoder making up the autoencoder, respectively. In the second stage, a (conditional) diffusion model is trained on the optimized latent embeddings $\mathbf{z} = \mathcal{E}_I(\mathbf{X})$. The generative process of

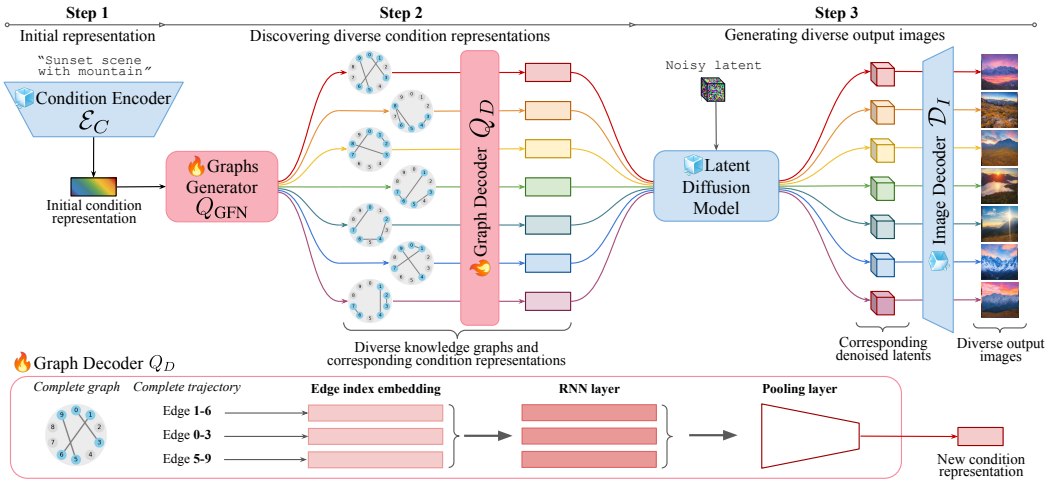


Figure 1: Rainbow operates by transforming an input condition into diverse images. Initially, it employs the pretrained condition encoder to derive an initial representation of the input condition (which contains uncertainty about locations or objects with the given prompt in this example). Then, a *graphs generator* produces multiple trajectories over a graph that reflect the input condition. These graphs are encoded into new latent condition representations. New condition representations and a latent noisy image are processed through the Latent Diffusion Model to acquire denoised image latents, which are subsequently decoded into diverse output images.

an LDM takes in a noisy latent $\mathbf{z}$ sampled from some prior distribution $p(\mathbf{z})$ and iteratively denoises it to produce a generated sample $\hat{\mathbf{z}}_0$ by $\hat{\mathbf{z}}_0 = \text{LDM}(\mathbf{z}, \mathbf{c})$, where $\mathbf{c}$ is the condition. Finally, the denoised latent is passed through the image decoder $\mathcal{D}_I$ by $\hat{\mathbf{X}} = \mathcal{D}_I(\hat{\mathbf{z}}_0)$ to obtain the synthesized image.

During training, noise is added to the latent representation to create a noisy latent image $\mathbf{z}_t$, where t denotes the diffusion timestep. The model predicts the noise ϵ added to the latent image, minimizing the difference between the predicted noise $\hat{\epsilon}$ and the actual noise ϵ at every timestep t, as described in Equation 2, where ϵ_ω is the neural backbone that performs time-conditioned denoising of the latent embedding. Typically, ϵ_ω is implemented as a time-conditional UNet [63, 65].

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{\mathcal{E}(\mathbf{X}),\epsilon\sim\mathcal{N}(0,1),t}||\epsilon - \epsilon_\omega(\mathbf{z}_t, t, \mathbf{c})||_2^2. \tag{2}$$

3 Rainbow

Rainbow captures the inherent uncertainty in conditional image generation and produces diverse yet realistic images that reflect the input condition. The core objective is to decompose the input condition into diverse latent representations, distinct yet jointly interpret the same condition. Each new latent representation is then processed to generate an output image, enabling us to produce various images corresponding to the input condition. To achieve diversity in latent representations, Rainbow conducts a latent graph in latent representation computation and utilizes the GFlowNets [4, 5] to sample diverse high-quality trajectories over the graph, which are then decoded into condition latent representations to produce diverse output images.

3.1 Rainbow's Inference Pipeline

Let M denote the number of images to be generated, and we assume that all models are fully trained. Figure 1 visualizes the inference process of Rainbow in three main steps.

First, the input condition (e.g. input prompt) C is encoded into a condition initial latent representation, denoted as $\mathbf{c} \in \mathbb{R}^{S_c}$, (*e.g.* $\mathbb{R}^{77\times1024}$), via a learned condition encoder $\mathcal{E}_C$. **Next**, the graph generator Q_{GFN} takes as input the initial condition embedding $\mathbf{c}$ and outputs a set of M distinct trajectories over the graph. Each generated graph is then transformed into a new condition representation by the graph decoder model Q_D, as described in Equation 3. **Finally**, a latent diffusion model generates images

from the set of condition embeddings $\hat{\mathbf{c}}^{1:M}$ and a set of noisy latents $\mathbf{z}^{1:M}$ sampled independently from the prior to generate M images $\hat{\mathbf{X}}^{1:M}$, described in Equation 4.

$$\hat{\mathbf{c}}^{1:M} = Q_D(Q_{\text{GFN}}(\mathbf{c})). \tag{3}$$

$$\left\{\hat{\mathbf{X}}^i = \mathcal{D}_I(\text{LDM}(\mathbf{z}^i, \hat{\mathbf{c}}^i)),\ i = 1, ..., M\right\} \tag{4}$$

3.2 Rainbow's Training Details

This section describes the training strategy to obtain diverse condition latent representations $\hat{c}^{1:M} \in \mathbb{R}^{M \times S_c}$ from the initial $c \in \mathbb{R}^{S_c}$. A detailed algorithm is provided in Appendix C. We assume the existence of a pretrained LDM. During Rainbow training, we freeze condition encoders $\mathcal{E}_C$, image encoder $\mathcal{E}_I$, image decoder $\mathcal{D}_I$, and Unet model; the graph generator Q_{GFN} and graph decoder Q_D are trained from scratch. We present Rainbow training progress into three stages as detailed below.

Stage 1: Discovering Diverse Graph Representations. We construct an undirected graph, $\mathcal{G}^*$, with N nodes, which yields $N(N-1)/2$ non-self-loop edges. Inspired by previous works [47, 80] that learn the underlying connections of variables in the latent space for greater interpretable context exploration, edge embeddings in Rainbow are randomly initialized. During training, Rainbow assigns interpretable meaning to edges without being constrained by pre-defined edge semantics.

We design Q_{GFN} as a GFlowNets model that iteratively predicts edges to be added to each of the set of M trajectories over $\mathcal{G}^*$, while maintaining a fixed per-graph sparsity, ρ. At each step, the GFlowNets predicts M edges, adding each edge to one of the M trajectories. This process is repeated for S steps. Specifically, the GFlowNets terminate the edge sampling process once ρ is reached, and no explicit terminal states are defined (as in [48]). The total number of edges S is calculated as $S = (1-\rho) \cdot \frac{N(N-1)}{2}$. With this design, the number of states in the GFlowNets equals the number of edges plus 1, $S+1$, which includes one initial state and S states for adding S edges.

In the initialization, we create a set of M trajectories $\mathcal{T}_{s=0}^{1:M} = \{\tau_{s=0}^1, \tau_{s=0}^2, \ldots, \tau_{s=0}^M\}$, where s represents the state index within the GFlowNets framework. We initialize each trajectory with a special starting element, the edge index 0. At state $s = S$, each trajectory in $\mathcal{T}_{s=S}^{1:M}$ is expected to be filled with S *edge indices* in the range 1 to $\frac{N(N-1)}{2}$.

At states $s = i$ with $1 \leq i \leq S$, we input the previous trajectory $\mathcal{T}_{s=i-1}^{1:M}$ and the condition $\mathbf{c}$ to predict probability of edges to add to the current M trajectories, one edge for each, as described in Equation 5. In Rainbow, the edge prediction strategy is derived from DBO, which is introduced in Section 2.1.1 with further details in Appendix C. After reaching the final state $s = S$, we obtain M trajectories $\mathcal{T}_{s=S}^{1:M}$, each filled with S edges.

$$\mathcal{T}_{s=i\in 1..S}^{1:M} = Q_{\text{GFN}}(\mathcal{T}_{s=i-1}^{1:M}, \mathbf{c}). \tag{5}$$

Stage 2: Decoding Graphs into Condition Representations. A graph decoder model Q_D is utilized to decode each trajectory into a condition representation of shape S_c, denoted as $\hat{\mathbf{c}}^{1:M} \in \mathbb{R}^{S_c}$. As visualized in Figure 1, Q_D is designed with three key steps.

First, for each trajectory $\mathcal{T}_{s=S}^i$, Q_D encodes the sequence of edge indices (of shape $\mathbb{R}^{S\times 1}$) into a sequence of edge embeddings of shape $\mathbb{R}^{S \times d_{\text{dim}}}$. Subsequently, these edge embeddings are passed through an RNN to ensure the order correlation of edges. Finally, the output of the RNN is processed by a projection pooling layer to map the sequence from $\mathbb{R}^{S \times d_{\text{dim}}}$ to the desired shape $\mathbb{R}^{S_c}$.

The final condition representations are computed as a convex combination between the diverse representations and the original condition representation $\mathbf{c}$ by a blending factor γ:

$$\hat{\mathbf{c}}^{1:M} = \gamma Q_D(\mathcal{T}_{\text{s=S}}^{1:M}) + (1-\gamma)\mathbf{c}. \tag{6}$$

Stage 3: Getting reward and computing losses. After obtaining $\hat{\mathbf{c}}^{1:M}$, we perform the diffusion process as introduced in Section 2.2 with the added noise ϵ_w and get M predicted noise $\hat{\epsilon}^{1:M}$. Subsequently, to evaluate how *good* the sampled M trajectories, corresponding to M predicted noise, the reward function $R(\mathcal{T}_{\text{s=S}}^{1:M})$, defined as the exponential of the negative MSE, as in Equation 7.

Our training objective combines two loss terms. First, the *GFlowNets loss*, $\mathcal{L}_{\text{GFN}}$, follows the DBO introduced in Section 2.1.1) with reward function $R(\mathcal{T}_{\text{s=S}}^{1:M})$. Second, the *diffusion denoising loss*,

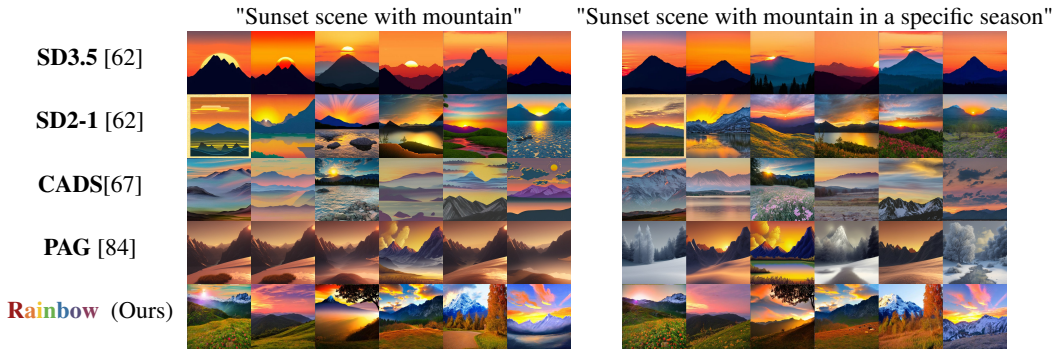


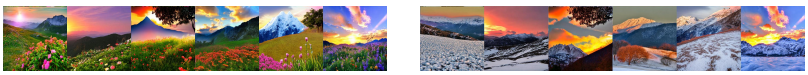
Figure 2: **Comparison of multiple images generated by baselines and Rainbow.** The baseline methods tend to produce images with repetitive layouts and primarily drawing art styles, failing to capture the uncertainty of "season". In contrast, Rainbow generates a variety of sunset scenes, showcasing diverse light levels, grass colors, and effectively capturing different seasons.

$\mathcal{L}_{\text{LDM}}$, (mentioned in Section 2.2) computes the mean squared error (MSE) between the added noise ϵ and the predicted noise $\hat{\epsilon}^{1:M}$. More specifically, $\mathcal{L}_{\text{GFN}}$ is to train the graphs generator Q_{GFN}, ensuring the diversity of sampled trajectories as well as the alignment to reward; meanwhile, $\mathcal{L}_{\text{LDM}}$ is to optimize the graph decoder model Q_D. The total loss $\mathcal{L}_{\text{total}}$ is a weighted combination:

$$R(\mathcal{T}_{s=S}^{1:M}) = e^{-\text{MSE}(\epsilon, \hat{\epsilon}^{1:M})}, \tag{7}$$

$$\mathcal{L}_{\text{GFN}} = \mathcal{L}_{\text{DB}}(\hat{\epsilon}^{1:M}, R(\hat{\epsilon}^{1:M})), \tag{8}$$

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{GFN}} + \beta\mathcal{L}_{\text{LDM}}. \tag{9}$$



(a) Base prompt + **Spring** edges (b) Base prompt + **Winter** edges

Figure 3: **Images generated by Rainbow seasonal edges** with the base prompt "Sunset scene with mountain". Most objects and layouts are consistent between the images, with noticeable season-specific details in the second image, such as spring flowers and winter snow.

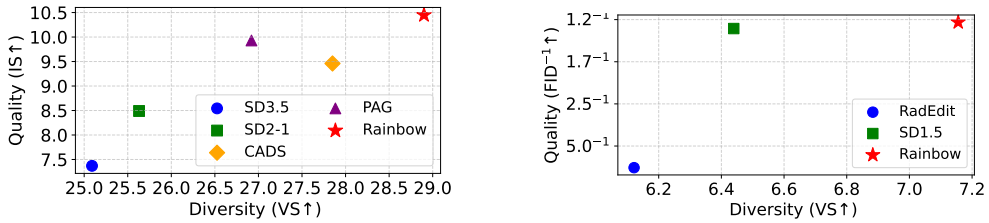
4 Experiment

We conduct experiments to investigate the following hypotheses. $\mathcal{H}_1$: Utilizing diverse graphs facilitates generating diverse images; $\mathcal{H}_2$: Latent graphs can be extracted into meaningful and interpretable patterns; $\mathcal{H}_3$: Improved ability to capture diversity enhances the performance of downstream tasks. Reproducibility details are in Appendix D.

4.1 Experiment Setup

Natural Images We use the Flickr30k dataset [83], which includes about 30k images paired with captions describing daily-life scenes, which contain uncertainty on object choices or styles. We build our Rainbow on top of the pretrained Stable Diffusion v2-1-base (SD2-1) with frozen pretrained VAE [35] image encoder/decoder, CLIP [56] text encoder, and Unet model. We evaluate the results against SD2-1, Stable Diffusion v3-medium (SD3.5), CADS [67] - a recent sampling strategy enhances diversity in the image-generation task, and pretrained checkpoint of PAG [84]- a recent work that improves diversity in the text-to-image task by prompt diversifying with GFlowNets. In our comparisons, both Rainbow and CADS utilize SD2-1's pretrained encoder-decoder and diffusion models. Rainbow's graph generator module includes $M = 40$, $N = 20$, and $S = 32$.

3D Brain MRIs We curate a dataset of about 27k datapoints for training with no diagnosed disease from the following datasets: ADNI [53], ABCD Study [33, 78], HCP [77], PPMI [55], and AIBL [14]. This task contains uncertainty in anatomical details such as ventricle sizes. Our setting employs



(a) Quantitative comparison on Natural Images (b) Quantitative comparison on Chest X-rays

Figure 4: **Quantitative analysis on diversity and image quality of SD-based Rainbow**. Rainbow consistently outperforms SD baselines in diversity with higher Vendi Score (VS) across domains and image quality with higher Inception Score (IS) in natural images and FID^{-1} in chest X-rays.

demographic input conditions on age and binary sex (0: male, 1: female) and fine-tunes the *Medical Open Network for Artificial Intelligence* (MONAI) [8]'s optimized 3D LDM along with Rainbow training. We benchmark Rainbow against LDM and a GAN-based [37] baselines. Rainbow's graph generator module includes $M = 8$, $N = 8$, and $S = 8$.

Chest X-rays We use the CheXpert dataset [30], which contains 170k training images. This dataset contains diversity in medical devices (such as chest tubes and wires), diseases (such as pneumonia and pleural-effusion) and anatomical details. We implement Rainbow on top of frozen parameters of a finetuned Stable Diffusion v1.5 (SD1.5) by previous work [36] for chest X-ray data. We generate 2D chest X-ray images based on text prompt conditions, *e.g., "Chest X-ray showing Support Devices"*. In addition to the finetuned SD1.5, we include RadEdit [52], a model trained from scratch on multiple chest radiology data such as CheXpert [29], MIMIC-CXR [31], and NIH-CXR [79] data for image editing tasks (more details at Appendix D.3), in the result comparison. Rainbow's graph generator module includes $M = 10$, $N = 20$, and $S = 33$.

4.2 Experiment Results

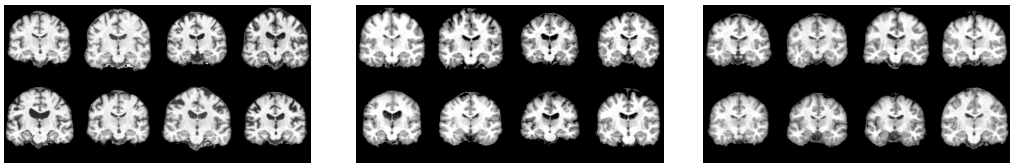
4.2.1 Diverse Images Generation Results

Investigating $\mathcal{H}_1$, we analyzed the generated images by baseline models and the proposed Rainbow in scenarios where the input condition contains uncertainty.

Natural Images Qualitatively, Figure 2 compares the generated images using two prompts, with 5 over 40 generations displayed. SD3.5 consistently generates repetitive layouts with backlit mountains, an orange sky, and ambiguous seasons. Additionally, SD2-1 offers a broader variety of objects and layouts but adheres to a drawing art style for the first prompt and predominantly generates spring scenes with green grass, lacking seasonal diversity for the second prompt. CADS marks diverse objects, yet produces unclear or winter-dominated images for the second prompt. PAG with one base image and diversified prompts does not introduce significant edits in this case and produces repeated layouts and many unclear season. Conversely, Rainbow produces images with diverse objects and light tones and effectively captures seasonal elements such as lush spring greenery and golden autumn foliage, even in the first prompt. For the second prompt, Rainbow demonstrates balanced seasons and clear seasonal features. We quantify the diversity and quality of generations in 60 prompts from the

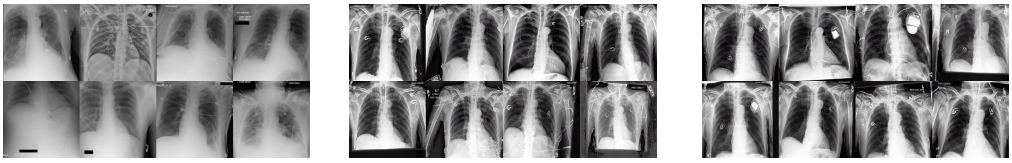
	FID score ↓			**Sex Classification Accuracy ↑**			**Age MAE ↓**		
Real data	1e-5			96%			3.55		
	Synthesis	Conv. age	Conv. sex	Synthesis	Conv. age	Conv. sex	Synthesis	Conv. age	Conv. sex
Random	-	-	-		48%			32.17	
GAN [19]	2.3329	-	-	63%	-	-	29.23	-	-
LDM [8]	0.3288	0.3570	0.3590	68%	68%	67%	21.52	20.35	19.43
Rainbow (Ours)	**0.3149**	**0.3375**	**0.3285**	**79%**	**80%**	**73%**	**13.73**	**18.59**	**16.40**

Table 1: **Quantitative evaluation on 3D Brain MRIs.** "Conv." stands for "Converted". The lower FID score, the higher accuracy, and the lower mean absolute error (MAE in years) indicate better performance. Rainbow outperforms baselines across tasks and metrics.



(a) Actual samples from dataset (b) Generations by Rainbow (c) Generations by baseline LDM

Figure 5: **Comparison of MRI image generations for *65-year-old male* individual.** Compared to actual samples from males aged 63-65, Rainbow captures greater diversity in details like ventricle sizes, while the baseline LDM generates images with less variation.



(a) Generations by RadEdit (b) Generations by SD1.5 (c) Generations by Rainbow

Figure 6: Comparison of chest X-ray image generations for the prompt *Chest X-ray with support devices*. Rainbow is able to provide high-quality images while generating a more diverse set of medical devices compared to the baselines - (a) and (b).

COCO Validation set [41], with 40 images per prompt. We use the Inception Vendi score (VS) [16] to evaluate diversity and the Inception score (IS) [70] to assess image quality. As shown in Figure 4a, Rainbow outperforms the baseline in both diversity (higher VS) and image quality (higher IS). In addition, we observe that all methods produce images relevant to the prompts with a similar CLIP score [57], which is approximately 30.3. Numeric results are in Table 3, classifier-free-guidance scale affects are discussed in Figure 18, full 40 generations in Figures 12−16, all in Appendix E.

Brain MRIs Figure 5 showcases the generated brain MRI images conditioned on a 65-year-old male individual. Unlike younger age groups (*e.g.* 10-20 years old) with characteristic small ventricles, or the older age groups (*e.g.* 70+ years old) with large ventricles [1, 17, 71], the 60s age range includes a wide variety of ventricle patterns [61], as visualized in Figure 5a with actual samples. While all models generated high-quality images, Rainbow effectively captures the diverse ventricle patterns with varying ventricle sizes, whereas the baseline LDM tends to produce similar ventricle regions across different samples. We provide full axial, coronal, and sagittal views for this experiment in Figure 28 and visibility of structures and details in generated 3D MRIs in Figure 27 in Appendix E.

In addition, we quantify brain MRI generations obtained from multiple tasks: image synthesis (with 200 conditions balanced on age and sex) and counterfactuals on age (three age shifts at 10, 40, and 80 years) and sex (flipping binary sex). For each task, we report FID score [72] and performance on age and sex prediction by pretrained classifier models (details at Appendix D.2). As shown in Table 1, Rainbow outperforms LDM and GAN baselines across all metrics and tasks with lower FID, higher sex accuracy, and lower age MAE. To justify age and sex prediction models, we report "Real" results that were tested on real data and "Random" results that were evaluated on random outputs. Figure 29 in Appendix E provides counterfactual generations.

Chest X-rays Figure 4b quantifies generations by Rainbow and baselines using FID and VS. Rainbow achieves a higher VS, indicating greater diversity than the finetuned SD model, while also improving image quality with a lower FID score. Both Rainbow and SD outperform the RadEdit. Figure 6 provides a qualitative comparison, images are generated using the prompt *"Chest X-ray showing support devices"*, where Rainbow generates a more diverse set of medical devices, such as pacemakers, in all generations, while baselines do not show any devices in some images. All models achieve similar CLIP scores of 33.5. Additional results including generations, Figure 22 and numeric results, Table 4, are outlined in Appendix E.

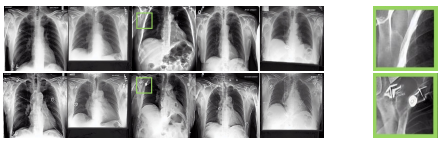


Figure 7: Images by Rainbow with the base prompt "Chest X-ray with no significant findings" (top) and being appended "supporting devices" edges (bottom) with devices added.

Figure 8: **Diversity in image editing**. Using the DAC [74] editing pipeline, given the cat (left) and "A cat wearing a wool cap", Rainbow captures diverse cap colors, while SD2-1 generates white caps.

4.2.2 Latent Graph Interpretation

Investigating $\mathcal{H}_2$, we explore sets of edges that present for 4 seasons in Figure 2 Section 4.2.1. From images generated by Rainbow with the two prompts, we first cluster images based on observable seasonal features, *e.g.* snow for winter. We then extract the 10 most frequently added edges for each season when having "in a specific season" in the prompt. Subsequently, we append these 10 extra edges to the original trajectories of the first prompt and generate new images. Figure 3 presents the effect of manipulated graphs on the newly generated images. We can observe the addition patterns of colorful flowers in Figure 3a and snow in Figure 3b. Although edges in the latent graph are not predefined, Rainbow can implicitly learn to capture specific context and group edges into meaningful features. For images with four seasons' edges, see Figure 17 in Appendix E.

We apply the same approach to chest X-rays to explore the set of "support devices" edges by extracting the 10 most frequently added edges when changing the prompt from "Chest X-ray with no significant finding" to "Chest X-ray showing support devices" (visualized in Figure 25 in Appendix E). Figure 7 shows the transformation with added "support devices" edges into trajectories with the appearance of medical devices in the generated images.

4.2.3 Performance on Downstream Tasks

Exploring $\mathcal{H}_3$, we conduct two downstream tasks: image editing and counterfactual generation.

We conduct an image-editing task with the natural-image domain that modifies the input image based on a prompt. We compare Rainbow to SD2-1 using the DAC image-editing pipeline [74]. Figure 8 shows the results of editing an image of a cat to add a wool cap. DAC with SD2-1 consistently produces images with a white cap, while Rainbow generates caps in multiple colors. This demonstrates Rainbow's ability to capture uncertainty (the cap color) and generate diverse samples. For the full 40 generations, see Figure 19 in Appendix E.

For 3D Brain MRIs illustrated in Figure 9, we perform the age prediction task using training data that includes synthesized data from Rainbow and the LDM baseline. We include 1600 generations arranged by age from 0 to 100 for both sexes, with each condition generating 8 samples. Real data is incorporated in specific proportions. Figure 9 visualizes that models trained with synthesized data generated by Rainbow achieved better performance. Both models performed best and better than models trained with fully real data at 50%. Specifically, at 50%, both models outperformed those trained with only 1600 real data points (100% dashed line).

4.3 Ablation studies

We assess the effect of varying the number of trajectories M. As shown in Figure 10 (left) for 3D brain MRI, a decrease in M leads to a drop in performance, with a significant performance decrease when $M = 1$. However, even with $M = 1$, Rainbow still performs better than the baseline LDM. A similar pattern is observed in Figure 10 (right) for chest X-ray data; the lowest performance corresponds to the lowest number of graphs, given the same sparsity, with $M = 10$ yielding the best performance on diversity assessment. Models for synthesizing chest X-rays are partially trained for 16,500 steps (out of 24,000 steps for fully trained models) with a fixed $N = 20$. Additional ablation results varying the number of nodes are provided in Appendix E.2.

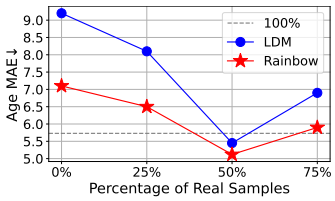


Figure 9: **Age Prediction Task trained on synthesized data.** Rainbow achieves the lowest MAE (Mean Absolute Error in years).

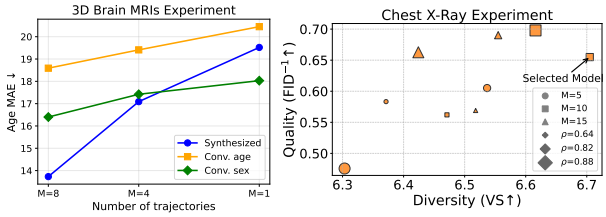


Figure 10: **Rainbow over number of trajectories** (M) showing that high values of M yield better performance, which are MAE (Mean Absolute Error in years) on the 3D brain MRI and high diversity (VS score) in the Chest X-ray experiments. ρ is the sparsity in Rainbow. "Conv." stands for "Converted".

5 Related Work

Conditional Image Generation with Diffusion Models have been driven by diffusion models, which iteratively denoise random inputs into coherent samples while outperforming GANs in stability and fidelity [26]. Key innovations include classifier-guided sampling [12], which steers generation using gradient signals from pretrained classifiers, and latent diffusion models (LDMs) [63], which operate in compressed latent spaces to enable high-resolution synthesis. Text-to-image models like DALL·E 2 [59] and Stable Diffusion [63] leverage large-scale multimodal pretraining to align textual prompts with visual concepts, while extensions like Palette [68] enable fine-grained control through spatial conditioning. These frameworks highlight the versatility of diffusion processes, with applications spanning artistic creation [69], medical imaging, and beyond.

Diversity in Conditional Generation Balancing diversity and fidelity remains a core challenge in conditional generation. GAN-based approaches address mode collapse via mode-seeking regularization [46] or self-conditioned clustering [43], while diffusion models inherently trade off diversity and quality through their noise schedules [2]. Methods like ControlNet [86] enhance controllability by injecting spatial constraints (e.g., edges, depth maps) into diffusion processes, whereas mutual information regularization [87] improves statistical dependency between latent codes and outputs. Adversarial training with semantic-guided negative sampling [9] further refines diversity in GANs, while category-consistent objectives [27] optimize photorealism and variation simultaneously. These advances collectively enable richer, more varied outputs without sacrificing semantic alignment.

Generative Flow Networks (GFlowNets) [4] offer a paradigm for sampling compositional objects (e.g., molecules, graphs) with probabilities proportional to a reward function, prioritizing diversity over the single-mode convergence of RL. Recent work extends this to sequential domains via recurrent architectures [50], demonstrating their capacity to model temporal dependencies in tasks like program synthesis. Our work adapts GFlowNets to *knowledge graph generation*, using the Trajectory Balance objective [45] to align forward edge-addition policies with backward inference while preserving order-dependent semantics through RNNs. To our knowledge, this is the first integration of GFlowNets with latent diffusion models.

6 Conclusion

We introduced Rainbow, a novel conditional image generation framework that captures uncertainty and produces diverse, plausible images. Rainbow constructs a latent graph in latent representation computation and leverages Generative Flow Networks to sample diverse trajectories over the graph, thereby enhancing the diversity of the condition latent representations and outputting diverse images that collectively interpret the input condition. Our experiments across natural and medical images not only demonstrate that Rainbow outperforms existing baselines in generating diverse, plausible images, but also highlight Rainbow's flexibility in adapting to any condition type. We discuss limitations and directions for future research in Appendix B.

Acknowledgement

This work was supported in part by National Institutes of Health grant AG089169. This study was also supported by the Stanford School of Medicine Department of Psychiatry and Behavioral Sciences Jaswa Innovator Award, and the Stanford HAI Hoffman-Yee Award. Part of the data used in the preparation of this article was obtained from the Australian Imaging, Biomarker & Lifestyle Flagship Study of Ageing (AIBL) database (aibl.csiro.aug), with preprocessing and harmonization performed by Mind Data Hub at Stanford University. We thank Mohammad Abbasi from Stanford Translational AI (STAI) in Medicine and Mental Health Lab, Stanford University, for his contribution to preprocessing all the 3D brain MRI datasets used in this work. Additionally, we would like to thank Natural Sciences and Engineering Research Council of Canada, the Canadian Institute for Advanced Research (CIFAR) Artificial Intelligence Chairs program for grants; Mila - Quebec AI Institute, Google Research, Calcul Quebec, and the Digital Research Alliance of Canada for providing grants and computing resources.

References

[1] Elwaleed Mustafa Alasar, Mohamet Awad Omer, and Ghada Abdel El Erahman Sakin. Morphometric assessment of aging impact in cranial/ventricles' volumes and ct/mri imaging systems parameters. *American Journal of Public Health Research*, 7(4):157–160, 2019.

[2] Pietro Astolfi, Marlene Careil, Melissa Hall, Oscar Mañas, Matthew Muckley, Jakob Verbeek, Adriana Romero-Soriano, and Michal Drozdzal. Consistency-diversity-realism pareto fronts of conditional image generative models. *ArXiv*, abs/2406.10429, 2024.

[3] Lazar Atanackovic, Alexander Tong, Bo Wang, Leo J. Lee, Yoshua Bengio, and Jason Hartford. Dyngfn: towards bayesian inference of gene regulatory networks with gflownets. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[4] Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Abu-Hussein. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:27381–27394, 2021.

[5] Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *J. Mach. Learn. Res.*, 24(1), March 2024.

[6] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria De La Iglesia-Vaya. Padchest: A large chest x-ray image dataset with multi-label annotated reports. *Medical image analysis*, 66:101797, 2020.

[7] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1209–1218, 2016.

[8] Manuel Jorge Cardoso, Wenqi Li, Richard Brown, Nic Ma, Eric Kerfoot, Yiheng Wang, Benjamin Murrey, Andriy Myronenko, Can Zhao, Dong Yang, Vishwesh Nath, Yufan He, Ziyue Xu, Ali Hatamizadeh, Wentao Zhu, Yun Liu, Mingxin Zheng, Yucheng Tang, Isaac Yang, and Andrew Feng. Monai: An open-source framework for deep learning in healthcare, 11 2022.

[9] Miriam Cha, Youngjune Gwon, and H. T. Kung. Adversarial learning of semantic relevance in text-to-image synthesis. *ArXiv*, abs/1812.05083, 2018.

[10] Minshuo Chen, Song Mei, Jianqing Fan, and Mengdi Wang. An overview of diffusion models: Applications, guided generation, statistical rates and optimization. *ArXiv*, abs/2404.07771, 2024.

[11] Joseph Paul Cohen, Joseph D. Viviano, Paul Bertin, Paul Morrison, Parsa Torabian, Matteo Guarrera, Matthew P Lungren, Akshay Chaudhari, Rupert Brooks, Mohammad Hashir, and Hadrien Bertrand. TorchXRayVision: A library of chest X-ray datasets and models. In *Medical Imaging with Deep Learning*, 2022.

[12] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[13] Mischa Dombrowski, Weitong Zhang, Sarah Cechnicka, Hadrien Reynaud, and Bernhard Kainz. Image generation diversity issues and how to tame them, 2024.

[14] Kathryn A Ellis, Ashley I Bush, David Darby, Daniela De Fazio, Jonathan Foster, Peter Hudson, Nicola T. Lautenschlager, Nat Lenzo, Ralph N. Martins, Paul Maruff, Colin Masters, Andrew Milner, Kerryn Pike, Christopher Rowe, Greg Savage, Cassandra Szoeke, Kevin Taddei, Victor Villemagne, Michael Woodward, and David Ames. The australian imaging, biomarkers and lifestyle (aibl) study of aging: methodology and baseline characteristics of 1112 individuals recruited for a longitudinal study of alzheimer's disease. *International Psychogeriatrics*, 21(4):672–687, 2009.

[15] Anders M. Fjell, Lars T. Westlye, Inge Amlien, Thomas Espeseth, Ivar Reinvang, Naftali Raz, Ingrid Agartz, David H. Salat, Douglas N. Greve, Bruce Fischl, Anders M. Dale, and Kristine B. Walhovd. Minute effects of sex on the aging brain: A multisample magnetic resonance imaging study of healthy aging and alzheimer's disease. *Journal of Neuroscience*, 29(27):8774–8783, 2009.

[16] Dan Friedman and Adji B. Dieng. Vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2306.13177*, 2023.

[17] Shohei Fujita, Susumu Mori, Kengo Onda, Shouhei Hanaoka, Yukihiro Nomura, Takahiro Nakao, Takeharu Yoshikawa, Hidemasa Takao, Naoto Hayashi, and Osamu Abe. Characterization of brain volume changes in aging individuals with normal cognition using serial magnetic resonance imaging. *JAMA Network Open*, 6(6):e2318153–e2318153, 06 2023.

[18] M. Gerstgrasser, R. Schaeffer, A. Dey, R. Rafailov, H. Sleight, J. Hughes, T. Korbak, R. Agrawal, D. Pai, A. Gromov, D. A. Roberts, D. L. Yang, and D. L. Donoho. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*, 2024.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 06 2014.

[20] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017.

[21] Ruben C Gur, Raquel E Gur, Warren Obrist, Beryl Skolnick, Martin Reivich, and Thomas E Schlaepfer. Sex differences in brain aging: A quantitative magnetic resonance imaging study. *Psychiatry Research*, 61(1):129–135, 1995.

[22] Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. In *Neural Information Processing Systems*, 2023.

[23] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6626–6637, 2017.

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[27] Tao Hu, Chengjiang Long, and Chunxia Xiao. Crd-cgan: Category-consistent and relativistic constraints for diverse text-to-image generation. *ArXiv*, abs/2107.13516, 2021.

[28] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[29] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

[30] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz,

Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: a large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[31] Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019.

[32] Minguk Kang and Jaesik Park. Contragan: contrastive learning for conditional image generation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[33] Nicole R. Karcher and Deanna M. Barch. The abcd study: understanding the development of risk for mental and physical health outcomes. *Neuropsychopharmacology*, 46(1):131–142, 2021.

[34] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *ArXiv*, abs/2206.00364, 2022.

[35] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[36] Amar Kumar, Anita Kriz, Mohammad Havaei, and Tal Arbel. Prism: High-resolution & precise counterfactual medical image generation using language-guided stable diffusion. *https://arxiv.org/abs/2503.00196*, 2025.

[37] Gihyun Kwon, Chihye Han, and Dae-shik Kim. Generation of 3d brain mri using auto-encoding generative adversarial networks. In Dinggang Shen, Tianming Liu, Terry M. Peters, Lawrence H. Staib, Caroline Essert, Sean Zhou, Pew-Thian Yap, and Ali Khan, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 118–126, Cham, 2019. Springer International Publishing.

[38] Generative AI Lab. Understanding model collapse: A hidden threat in generative ai, 2024.

[39] Adrien LeCoz, Stéphane Herbin, and Faouzi Adjed. Explaining an image classifier with a generative model conditioned by uncertainty, 2024.

[40] Minhyeok Lee and Junhee Seok. Estimation with uncertainty via conditional generative adversarial networks. *Sensors*, 21(18), 2021.

[41] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[42] Dianbo Liu, Moksh Jain, Bonaventure Dossou, Qianli Shen, Salem Lahlou, Anirudh Goyal, Nikolay Malkin, Chris Emezue, Dinghuai Zhang, Nadhir Hassen, Xu Ji, Kenji Kawaguchi, and Yoshua Bengio. Gflowout: Dropout with generative flow networks, 2023.

[43] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14274–14283, 2020.

[44] Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 5955–5967. Curran Associates, Inc., 2022.

[45] Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:5955–5967, 2022.

[46] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, S. Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1429–1437, 2019.

[47] Trang Nguyen, Amin Mansouri, Kanika Madan, Nguyen Duy Khuong, Kartik Ahuja, Dianbo Liu, and Yoshua Bengio. Reusable slotwise mechanisms. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[48] Trang Nguyen, Alexander Tong, Kanika Madan, Yoshua Bengio, and Dianbo Liu. Causal inference in gene regulatory networks with gflownet: Towards scalability in large systems, 2023.

[49] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 18–24 Jul 2021.

[50] Ling Pan, Nikolay Malkin, Dinghuai Zhang, and Yoshua Bengio. Gflownet-em for sequential structured prediction. In *International Conference on Machine Learning (ICML)*, 2023.

[51] Wei Peng, Tian Xia, Fabio De Sousa Ribeiro, Tomas Bosschieter, Ehsan Adeli, Qingyu Zhao, Ben Glocker, and Kilian M. Pohl. Latent 3d brain mri counterfactual, 2024.

[52] Fernando Pérez-García, Sam Bond-Taylor, Pedro P Sanchez, Boris van Breugel, Daniel C Castro, Harshita Sharma, Valentina Salvatelli, Maria TA Wetscherek, Hannah Richardson, Matthew P Lungren, et al. Radedit: stress-testing biomedical vision models via diffusion image editing. In *European Conference on Computer Vision*, pages 358–376. Springer, 2024.

[53] Ronald C. Petersen, Paul S. Aisen, Laurel A. Beckett, Michael C. Donohue, Anthony C. Gamst, Danielle J. Harvey, Jr. Jack, Clifford R., William J. Jagust, Leslie M. Shaw, Arthur W. Toga, John Q. Trojanowski, and Michael W. Weiner. Alzheimer's disease neuroimaging initiative (adni): clinical characterization. *Neurology*, 74(3):201–209, 2010.

[54] Walter H. L. Pinaya, Petru-Daniel Tudosiu, Jessica Dafflon, Pedro F da Costa, Virginia Fernandez, Parashkev Nachev, Sebastien Ourselin, and M. Jorge Cardoso. Brain imaging generation with latent diffusion models, 2022.

[55] E. Pulliam and A. B. Singleton. The parkinson's progression markers initiative (ppmi): Study design and protocol. *Movement Disorders*, 26(9):1453–1460, 2011.

[56] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Phil Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *ICML*, 2021.

[58] Marianne Rakic, Hallee E. Wong, Jose Javier Gonzalez Ortiz, Beth Cimini, John Guttag, and Adrian V. Dalca. Tyche: Stochastic in-context learning for medical image segmentation, 2024.

[59] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[60] Royi Rassin, Aviv Slobodkin, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg. Grade: Quantifying sample diversity in text-to-image models, 2025.

[61] Naftali Raz, Ulman Lindenberger, Karen M Rodrigue, Kristen M Kennedy, Denise Head, Adrienne Williamson, Cheryl Dahle, Denis Gerstorf, and James D Acker. Aging, sexual dimorphism, and hemispheric asymmetry of the cerebral cortex: Replicability of regional differences in volume. *Neurobiology of Aging*, 25(3):377–396, 2004.

[62] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021.

[63] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[64] 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 (CVPR)*, pages 10684–10695, June 2022.

[65] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[66] Mihaela Rosca, Balaji Lakshminarayanan, David Warde-Farley, and Shakir Mohamed. Variational approaches for auto-encoding generative adversarial networks, 2017.

[67] Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M. Weber. CADS: Unleashing the diversity of diffusion models through condition-annealed sampling. In *The Twelfth International Conference on Learning Representations*, 2024.

[68] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Raphael Gontijo Lopes, et al. Palette: Image-to-image diffusion models. *arXiv preprint arXiv:2111.05826*, 2022.

[69] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022.

[70] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 2016.

[71] Rachael I. Scahill, Chris Frost, Rhian Jenkins, Jennifer L. Whitwell, Martin N. Rossor, and Nick C. Fox. A longitudinal study of brain volume changes in normal aging using serial registered magnetic resonance imaging. *Archives of Neurology*, 60(7):989–994, 07 2003.

[72] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.3.0.

[73] I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, and Y. Gal. Ai models collapse when trained on recursively generated data. *Nature*, 625:1–5, 2024.

[74] Xue Song, Jiequan Cui, Hanwang Zhang, Jingjing Chen, Richang Hong, and Yugang Jiang. Doubly abductive counterfactual inference for text-based image editing. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9162–9171, 2024.

[75] David Stap, Maurits J. R. Bleeker, Sarah Ibrahimi, and Maartje ter Hoeve. Conditional image generation and manipulation for user-specified content. *ArXiv*, abs/2005.04909, 2020.

[76] G. Turinici. Diversity in deep generative models and generative ai. *arXiv preprint arXiv:2202.09573*, 2023.

[77] David C. Van Essen, Stephen M. Smith, Deanna M. Barch, Timothy E.J. Behrens, Essa Yacoub, and Kamil Ugurbil. The wu-minn human connectome project: An overview. *NeuroImage*, 80:62–79, 2013. Mapping the Connectome.

[78] N. D. Volkow, G. F. Koob, R. T. Croyle, R. L. Balster, A. R. Childress, and C. R. Schuster. The abcd study: Enhancing the understanding of adolescent brain and cognitive development. *NeuroImage*, 163:1–5, 2019.

[79] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3462–3471, 2017.

[80] Elizabeth Wilson, Mika Satomi, Alex McLean, Deva Schubert, and Juan Felipe Amaya Gonzalez. Embodied exploration of latent spaces and explainable ai, 2024.

[81] Jiang Xu, Shotai Kobayashi, Shuhei Yamaguchi, Ken-ichi Iijima, Kazunori Okada, and Kazuya Yamashita. Gender effects on age-related changes in brain structure. *American Journal of Neuroradiology*, 21(1):112–118, 2000.

[82] Mengping Yang and Zhe Wang. Image synthesis under limited data: A survey and taxonomy, 2023.

[83] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78, 2014.

[84] Taeyoung Yun, Dinghuai Zhang, Jinkyoo Park, and Ling Pan. Learning to sample effective and diverse prompts for text-to-image generation, 2025.

[85] Guiyu Zhang, Huan ang Gao, Zijian Jiang, Hao Zhao, and Zhedong Zheng. Ctrl-u: Robust conditional image generation via uncertainty-aware reward modeling, 2024.

[86] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *ArXiv*, abs/2302.05543, 2023.

[87] Zhiwen Zuo, Ailin Li, Zhizhong Wang, Lei Zhao, Jianfeng Dong, Xun Wang, and Meng Wang. Statistics enhancement generative adversarial networks for diverse conditional image synthesis. *IEEE Transactions on Circuits and Systems for Video Technology*, 34:6167–6180, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope, as we have included baselines and experiments that address all claims made.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, the paper discusses the limitations of the work performed by the authors, as we have addressed it in B.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Yes, the paper provides mathematical formulation for GFlowNet and demonstrating how the algorithm can help Rainbow to enhance diversity.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, the paper fully discloses all the information needed to reproduce the main experimental results, including a parameter table and figures for hyperparameter tuning in the Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Yes, the paper provides open access to the data and specifies that the code will be provided upon acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, the paper specifies all the training and test details, including data splits, hyperparameters, selection criteria, and type of optimizer, clearly in both the main text and the Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes, the paper reports appropriate quantitative and qualitative results in the experiments to substantiate all claims.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, the paper provides sufficient information on the computer resources needed to reproduce the experiments, including details about the GPU used and the training duration.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, the research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes, we discussed in Appendix A.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly credited the creators and original owners of the code, pretrained models, and checkpoints used in our paper, and have explicitly mentioned and respected the license and terms of use associated with these assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will contribute the source code and pretrained model from this work to the research community, and will provide detailed documentation of the training process and usage instructions alongside the assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not include any crowdsourcing experiments or research with human subjects. Instead, we have used the Flickr30k dataset, which contains images from human daily life, and we have provided relevant details in the dataset introduction.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not include any crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Our method development does not involve LLM.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Broader Impacts

In this work, we presented Rainbow, a method that generates diverse images. These types of methods can play a crucial role in advancing AI models by enhancing robustness, reducing biases, and improving performance in various downstream tasks. By ensuring greater diversity in generated data, these models help mitigate biases that arise from underrepresented groups, building more equitable and generalizable AI systems. Additionally, generative models can be leveraged for counterfactual image generation, enabling the exploration of alternative scenarios for scientific and medical applications. However, while diversity is valuable, the realism of generated natural images raises concerns about misinformation and the misuse of synthetic content. In medical applications, counterfactual images must be rigorously validated by domain experts before being used to train autonomous systems, as incorrect or misleading data could lead to severe clinical consequences. Expert validation ensures that these images maintain diagnostic fidelity, preserving patient safety and the integrity of AI-driven medical decision-making.

B Limitations and Future Work

Although we introduced significant advancements in Rainbow and demonstrated experimental improvements, some limitations suggest potential areas for future work. One limitation is the higher computational resources required for training Rainbow, as we update M trajectories in parallel. Future research could focus on optimizing the training process to alleviate these computational demands. Although working on latent graph reveals a high level of flexibility for Rainbow to be applicable to any kind of condition, another area for improvement is the interpretation of latent graphs in Rainbow. This would aid in enhancing the latent graphs' interpretability more automatically.

In terms of directions for future exploration, one direction is to extend Rainbow to other domains that require diversity and the ability to manage uncertainty, such as text generation, recommendation systems, and decision-making tasks. Another promising direction is to scale Rainbow to even larger latent graphs, which could capture uncertainties across multiple tasks. This expansion could lead to the creation of a foundational world model capable of addressing uncertainties in a variety of applications. One crucial analysis to add is anatomical plausibility tests and checks before making use of these images for any clinical application.

C Further Computation Details

C.1 Conditional Image Generation with Latent Diffusion

During Training The goal is to generate an output image $\mathbf{X}$ given an input image $\mathbf{I} \in \mathbb{R}^{\mathcal{S}_I}$ with shape $\mathcal{S}_I$ and a condition $\mathbf{C}$. Initially, the input image $\mathbf{I}$ is encoded into a latent representation $\mathbf{z}_0^I = \mathcal{E}_I(\mathbf{I})$ using an encoder $\mathcal{E}_I$. The latent code $\mathbf{z}_0^I \in \mathbb{R}^{\mathcal{L}_I}$ represents the underlying structure of the image in a lower-dimensional latent space, where $\mathcal{L}_I$ denotes the dimensions of this latent space.

The latent code $\mathbf{z}_0^I$ is then subjected to a forward diffusion process, which iteratively adds noise over T steps:

$$q(\mathbf{z}_t^I \mid \mathbf{z}_{t-1}^I) = \mathcal{N}(\mathbf{z}_t^I \mid \sqrt{\alpha_t}\mathbf{z}_{t-1}^I, (1-\alpha_t)\mathbf{I}),$$

where α_t is a variance scheduling parameter that controls the amount of noise added at each step.

Concurrently, the condition $\mathbf{C}$ is encoded into its own latent representation $\mathbf{c} = \mathcal{E}_C(\mathbf{C})$ using an encoder $\mathcal{E}_C$. This latent representation $\mathbf{c} \in \mathbb{R}^{\mathcal{L}_C}$ encapsulates the conditioning information needed for the generation process, where $\mathcal{L}_C$ denotes the dimensions of the conditional latent space.

During the reverse diffusion process, the objective is to reconstruct the original latent representation $\mathbf{z}_0^I$ from the noisy latent code $\mathbf{z}_T^I$, guided by the condition latent representation $\mathbf{c}$. This reverse diffusion is generally modeled using a neural network ϵ_θ, which predicts the noise added at each timestep of the diffusion process in a commonly used formulation:

$$\mathbf{z}_{t-1}^I = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{z}_t^I - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(\mathbf{z}_t^I, t, \mathbf{c})\right) + \sigma_t \mathbf{n},$$

where $\mathbf{n} \sim \mathcal{N}(0, \mathbf{I})$ and σ_t are scaling factors for the noise at step t. Note that there are variations in diffusion modeling where different parameter schedules or noise prediction methods might be employed.

Finally, the reconstructed latent code $\mathbf{z}_0^I$ is decoded back into the image space to produce the final output image $\hat{\mathbf{X}} = \mathcal{D}(\mathbf{z}_0^I)$, where $\mathcal{D}$ is the decoder function that maps the latent representation back to the high-dimensional image space, yielding the generated image $\hat{\mathbf{X}}$.

Image Generation from Input Condition Along with the encoded latent representation $\mathbf{c}$ from input condition C, a latent image $\mathbf{z}_T$ is sampled from a Gaussian prior distribution, typically $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The reverse diffusion process, conditioned on $\mathbf{c}$, iteratively refines $\mathbf{z}_T$ until obtaining $\mathbf{z}_0$. This final latent code $\mathbf{z}_0$ is then decoded using $\mathcal{D}$ to produce the output image $\hat{\mathbf{X}}$.

Counterfactual Generation from Input Image and Condition The input image $\mathbf{I}$ is encoded into its latent representation $\mathbf{z}_0^I = \mathcal{E}_I(\mathbf{I})$. The counterfactual condition $\mathbf{C}'$ is encoded into $\mathbf{c}' = \mathcal{E}_C(\mathbf{C}')$. The latent code $\mathbf{z}_0^I$ is perturbed with noise and reverse diffusion, guided by $\mathbf{c}'$, is applied to generate a new latent code $\mathbf{z}_0$. Finally, this modified latent code $\mathbf{z}_0$ is decoded using $\mathcal{D}$ to obtain the counterfactual image $\hat{\mathbf{X}}$.

C.2 GFlowNets Training

The Algorithm 1 describes the training process of Rainbow to iteratively construct diverse trajectories over the latent graph using GFlowNets and the Detailed Balance (DB) objective. The process is divided into three phases: initialization, iterative edge sampling, and loss computation.

C.2.1 Initialization Phase

The algorithm initializes with the input as the initial condition representation $c \in \mathbb{R}^{S_c}$. Configuration includes the number of parallel graphs M, the number of nodes N, and the sparsity ρ. The total number of edges S is calculated as $S = (1 - \rho) \cdot \frac{N(N-1)}{2}$. A set of M trajectories $\mathcal{T}_{s=0}^{1:M} \in \mathbb{R}^{M \times 1}$ is initialized with a special starting edge index 0. Two masks—`forward_mask` $\in \mathbb{R}^{M \times n}$ and `backward_mask` $\in \mathbb{R}^{M \times n}$—are created to enforce valid transitions during sampling. These masks control which edges are available to be reached in the forward and backward paths. The log-likelihood difference tensor $ll\_diff \in \mathbb{R}^{(S+1) \times M}$ is initialized to track state transitions for the DB loss.

C.2.2 Iterative Edge Sampling Phase

For each step $i \in \{1, \ldots, S\}$, the algorithm performs the following operations:

Graph Encoding and Forward/Backward Probabilities: The current trajectories $\mathcal{T}_{s=i-1}^{1:M}$ are encoded into latent representations $rep_g \in \mathbb{R}^{M \times h_g}$ using the graph decoder $\mathbf{Q_d^g}$. These are concatenated with the repeated condition c to form a combined representation $rep \in \mathbb{R}^{M \times h}$. The forward predictor $\mathbf{MLP_{FW}}$ computes log-forward probabilities $log\_forward \in \mathbb{R}^{M \times n}$ and log-flow values $log\_flow \in \mathbb{R}^{M \times 1}$, masked to exclude already added edges. The backward predictor $\mathbf{MLP_{BW}}$ computes log-backward probabilities $log\_backward \in \mathbb{R}^{M \times n}$, masked to restrict invalid backward transitions.

Edge Sampling and Mask Updates: Edges are sampled from $log\_forward$ using a multinomial distribution, and the trajectories $\mathcal{T}_{s=i}^{1:M}$ are updated with the new edges. The log-likelihood differences $ll\_diff[i]$ accumulate the log-flow and log-forward probabilities of the sampled edges. For $i > 1$, $ll\_diff[i-1]$ is updated with backward probabilities to ensure transition consistency. The masks `forward_mask` and `backward_mask` are dynamically adjusted to reflect added edges.

C.2.3 Finalization and Loss Computation Phase

After S steps, the completed trajectories $\mathcal{T}_{s=S}^{1:M}$ are decoded into condition representations $\hat{c}^{1:M}$ using the graph decoder Q_D, blended with the original condition c via the factor γ as in Equation 6. The diffusion denoising process generates rewards $log\_reward \in \mathbb{R}^{M \times 1}$, which are incorporated into $ll\_diff[S]$. The DB loss $\mathcal{L}_{DB}$ is computed as the mean squared error of ll_diff.

$$\mathcal{L}_{DB} = \text{mean}(ll\_diff^2).$$

Algorithm 1 Rainbow Training Pipeline

Input: Intial condition representation $c \in \mathbb{R}^{S_c}$
M: Number of graphs that are computed parallelly
N: Number of nodes in graph
ρ: Sparsity
$n = N(N-1) * 0.5 * (1-\rho)$: Total number of edges in undirected graphs with N nodes
S: Number of edges in the final graph

GFlowNets Architecture
h_g, h_c: dimension of the encoded graph g_i and the encoded condition c, respectively
$h = h_g + h_c$: GFlowNets hidden size
$\mathbf{Q_E^c} : \mathbb{R}^{S_c} \to \mathbb{R}^{h_c}$ ▷ Condition encoder model
$\mathbf{Q_d^g} : \mathbb{R}^{n} \to \mathbb{R}^{h_g}$ ▷ Instant graph decoder model (Used during edges sampling)
$\mathbf{MLP_{FW}} : \mathbb{R}^{h} \to \mathbb{R}^{n+1}$ ▷ Forward probability and flow predictor, n dimensions for forward probability, and the last dimension for state flow
$\mathbf{MLP_{BW}} : \mathbb{R}^{h} \to \mathbb{R}^{n}$ ▷ Backward probability predictor

GFlowNets computation flow for transforming state $i-1$ to state i
Inputs: $\mathcal{T}_{s=i-1}^{1:M} \in \mathbb{R}^{M \times i}, \quad c' \in \mathbb{R}^{h_c}$
$rep_g = \mathbf{Q_E^g}(\mathcal{T}_{s=i-1}^{1:M}) \in \mathbb{R}^{M \times h_g}$ ▷ Encoding graphs from state i-1
$rep = \text{concatenate}(rep_g,\ c'.repeat(M))$ ▷ Concatenate encoded graphs and the repeated encoded condition
$pred = \mathbf{MLP_{FW}}(rep)$ ▷ Preparing for forward probability and flow prediction
$log\_forward = \text{log\_softmax}(pred[:, :-1] - forward\_mask * \inf)$ ▷ Getting forward probability that with added edges excluded
$log\_flow = pred[:, -1:]$ ▷ Getting log flow
$log\_backward = \text{log\_softmax}(\mathbf{MLP_{BW}}(rep) - backward\_mask * \inf))$ ▷ Getting backward probability among added edges
Outputs: $log\_forward \in \mathbb{R}^{M \times n}$, $log\_backward \in \mathbb{R}^{M \times n}$, $flow \in \mathbb{R}^{M \times 1}$

Step 0. Initialization
$\mathcal{T}_{s=0}^{1:M} \leftarrow \mathbf{0}_{M,1}$ ▷ Empty M undirected graphs with a starting special edge index 0
$forward\_mask \leftarrow \mathbf{0}_{M \times n}, \quad backward\_mask \leftarrow \mathbf{1}_{M \times n}$ ▷ Initialize masks
$ll\_diff \leftarrow \mathbf{0}_{(S+1) \times M}$ ▷ First state for the initial state and S states for adding S edges, $ll\_diff[0]$ is not touched

Graphs generator training pipeline using GFlowNets and Detail-balance loss
$c' = \mathbf{Q_E^c}(c)$
for i in $1 \dots S$ **do** ▷ Loops of execution
    $log\_forward,\ log\_backward,\ log\_flow = \mathbf{Q_{GFN}}(\mathcal{T}_{s=i-1}^{1:M}, c')$
    $edges \leftarrow \text{multinomial}(log\_forward) \in \mathbb{R}^{M \times n}$ ▷ Sampling edges as actions
    $\mathcal{T}_{s=i}^{1:M} \leftarrow edges$ ▷ Action as edges are added into trajectories

    # Updating flows
    $ll\_diff[i] \mathrel{+}= log\_flow + log\_forward.\text{gather}(actions)$
    if i > 1 **then**
        $ll\_diff[i-1] \mathrel{-}= log\_flow - log\_backward.\text{gather}(actions)$
    end if
    $forward\_mask \mathrel{+}= actions$ ▷ Updating the forward mask
    $backward\_mask \mathrel{-}= actions$ ▷ Updating the backward mask
    if i == S **then** ▷ Completing the last turn
        $\mathcal{T}_{s=S}^{1:M} = \mathcal{T}_{s=S}^{1:M}$
        $\hat{c}^{1:M} = Q_D(\mathcal{T}_{s=S}^{1:M}) * \gamma + c * (1-\gamma)$ ▷ Decode *done* graphs into latent condition shape
        Performing Diffusion denoising conditioned on $\hat{c}^{1:M}$ and get $log\_reward \in \mathbb{R}^{M \times 1}$ as in Equation 8
        $ll\_diff[S] \mathrel{-}= log\_rewards$
    end if
end for
$\mathcal{L}_{DB} = ll\_diff^2.\text{mean}()$ ▷ Optimization step

D Experiment Setup

Table 2 indicates hyperparameters used in this work for all experiments.

D.1 Hyper-parameter

Parameter Name	Natural Images	3D Brain MRIs	Chest X-rays
Learning Rate	1e-5	25e-7	1e-5
Pretrained-LDM epochs	-	80	-
Rainbow epochs	3	20	5
Batch Size	1	1	8
α	1 (Freeze Unet)	0.2	1 (Freeze Unet)
β	1	0.8	1
Training image shape	$3 \times 256 \times 256$	$1 \times 160 \times 192 \times 176$	512×512
Training condition type	Text prompt	Age and binary sexes	Text prompt
Training condition shape	Dynamic length	2 dimensions	Dynamic length
Latent image shape	$4 \times 64 \times 64$	$1 \times 32 \times 40 \times 48 \times 44$	$4 \times 64 \times 64$
Latent condition shape S_c	77×1024	256	77×1024
Inference image shape	$3 \times 512 \times 512$	$1 \times 160 \times 192 \times 176$	512×512
Encoder - Decoder	VAE	VAE	VAE
Graph Size N	20 nodes	8 nodes	20 nodes
Number of Graphs M	40	8	10
Sparsity ρ	0.83	0.70	0.82
Num. Edges S	32	8	33
Use RNN	Yes	Yes	Yes
Edge Embedding dim	512	128	512
Latent $dimh_g = h_c$	1024	1024	1024
Blending factor γ	0.5	0.5	0.5

Table 2: Model Architecture and Parameter Indications

D.2 Evaluation Metrics

Natural Images To evaluate Rainbow on natural images, we use Inception Score (IS) and Inception Vendi Score (VS). For both metrics, we use the feature extraction model from pre-trained Pytorch Inception-v3.

3D Brain MRIs We use Fréchet Inception Distance [23] (FID) to evaluate the feature quality of synthetic images. FID is a widely adopted metric for assessing the similarity between the feature distributions of real and generated images. It is based on the premise that high-quality synthetic images should exhibit feature distributions similar to those of real images when passed through a pre-trained neural network. FID is computed by calculating the Fréchet distance between two multivariate Gaussian distributions fitted to the feature vectors of real and generated images.

For feature extraction, we use a 3D ResNet50, which is particularly well-suited for capturing the complex 3D structures and patterns inherent in volumetric data. The model is trained on a diverse set of 23 medical imaging datasets, enabling it to generalize effectively across various medical image types, such as MRI and CT scans. The feature vectors are extracted from the final convolutional layer (conv seg), with a dimensionality of 2048. This layer captures high-level semantic features of the images, making it ideal for evaluating perceptual similarity between the real and synthetic images.

Lower FID values indicate that the distributions of real and generated images are more closely aligned, suggesting that the synthetic images are of higher quality.

To evaluate the faithfulness of capturing these conditions in our generated samples, we train a CNN-based age regressor and sex classifier on the real data (i.e., the same data that is used to train our proposed model and all baselines). The architectures for these CNNs can essentially be seen as the encoder half of a typical UNet [65], consisting of 4 downsampling levels and 2 convolutional blocks per level, with each block consisting of a convolution layer, batchnorm layer, and ReLU layer. The

age regressor is trained to minimize MSE loss, and the sex classifier is trained to minimize binary cross-entropy.

Chest X-rays To evaluate Rainbow on chest X-rays, we use FID and Vendi Score similar to the previous modalities. For feature extraction, we use a pre-trained DenseNet-121 [28] model from the TorchXrayVision library [11], which is trained on multiple chest X-ray datasets such as CheXpert [29], NIH-CXR [79], PadChest [6], and MIMIC-CXR [31]. The feature vectors used for calculating the metrics are extracted from the last layer (before the classifier head) with a dimensionality of 1024.

D.3 Baselines

Natural Images We use 4 baselines: Stable Diffusion v2-1-base (SD2-1) [64][3], Stable Diffusion v3-medium [4], CADS [67], which is a sampling strategy that anneals the conditioning signal by adding scheduled, monotonically decreasing Gaussian noise to the conditioning vector during inference to balance diversity and condition alignment, and PAG [84], a novel approach that frames prompt adaptation as a probabilistic inference problem utilizing GFlowNet for the generation of diverse, high-quality prompts.

Brain MRIs The GAN based model is from [37]. This 3D GAN model addresses both image blurriness and mode collapse problems by leveraging α-GAN [66] that combines the advantages of Variational Auto-Encoder (VAE) and GAN with an additional code discriminator network. The model also uses the Wasserstein GAN with Gradient Penalty (WGAN-GP) loss [20] to lower the training instability.

The standard LDM is based on [54, 8, 63]. It can be viewed as identical to Rainbow, except that it does not leverage any latent graphs.

Chest X-rays We consider two chest X-ray baseline models. The first model is RadEdit [52], a latent diffusion model developed by Microsoft Health. This model is trained on 487,680 frontal view chest X-rays of multiple datasets such as MIMIC-CXR [31], NIH-CXR [79], and CheXpert [29]. The second baseline is a fine-tuned Stable Diffusion v1.5 model [36] on the CheXpert [29] dataset.

D.4 Training Time and Computation Sources

For general-domain experiments, training was conducted on a single NVIDIA H100-80GB GPU, completing in 12 hours.

The brain MRI experiment utilized 4 NVIDIA H100-80GB GPUs paired with 32 CPU cores over 3 days pretraining, followed by continued Rainbow training on the same hardware configuration for an additional 24 hours.

The chest X-ray experiments utilized 4 NVIDIA A100-80GB GPUS paired with 24 CPU cores. This model was finetuned on 512×512 chest X-rays and prompt pairs for 15 hours with an overall batch-size of 8.

E Addition Experiment Results

E.1 Numeric Evaluation Results

Table 3 presents numeric results for Figure 4a.

Table 4 presents numeric results for Figure 4b.

E.2 Chest X-Ray Ablation Studies

Figure 11 shows the results for the full ablation study across three parameters N, M, and ρ for the chest X-ray model. The ablation setup is same as discussed in the main text, where each model is trained on a single GPU for 16500 steps. It can be observed that the models with more diversity (higher VS) have higher sparsity values, and the models with lower FID have lower sparsity. However,

[3] `https://huggingface.co/stabilityai/stable-diffusion-2-1-base`
[4] `https://huggingface.co/stabilityai/stable-diffusion-3-medium`

Model name	IS ↑	CLIP ↑	Pixcel VS ↑	Inception VS ↑
SD3.5	7.37	30.29	3.75	25.09
SD2-1	8.49	30.27	3.74	25.63
CADS	9.46	30.27	3.88	27.85
PAG	9.93	30.21	3.90	26.92
DDIM	8.44	30.28	3.91	26.62
Rainbow	**10.45**	**30.32**	**3.94**	**28.90**

Table 3: **Quantitative comparison of natural-image experiment**. An upward arrow "↑" indicates that a higher value corresponds to better performance. Rainbow consistently outperforms SD baselines in diversity (higher Vendi scores), image quality (higher IS score), and prompt context delivery (higher or comparable CLIP score)

Model name	FID ↓	CLIP ↑	VS ↑
RadEdit	10.28	**33.58**	6.12
SD1.5	1.32	33.51	6.44
Rainbow	**1.27**	33.45	**7.16**

Table 4: **Quantitative comparison of Chest X-Ray experiment** . An upward arrow "↑" indicates that a higher value corresponds to better performance. Rainbow is increasing the VS and achieves higher diversity while improving the generation quality by lowering FID. All models have close CLIP similarity scores.

this does not imply that sparsity parameters are the single factor affecting quality and diversity. In general, we see more squares and triangles in the upper right section of the plot, showing that $M = 10$ and $M = 15$ are overall better than the $M = 5$ models. Furthermore, blue and orange are also more prevalent in the upper-right section of the plot indicating that models with $N = 25$ do not perform as good as models with fewer nodes.

E.3 Full of 40 Generations for General-domain Experiments

Figure 19 provides the 40 generations that support Figure 8 in the main text.

Supporting Figure 2 in the main text, Figure 12 present full 40 generations with two prompts by SD3.5, Figure 13 present whole 40 generation with two prompts by SD2-1, Figure 14 present full 40

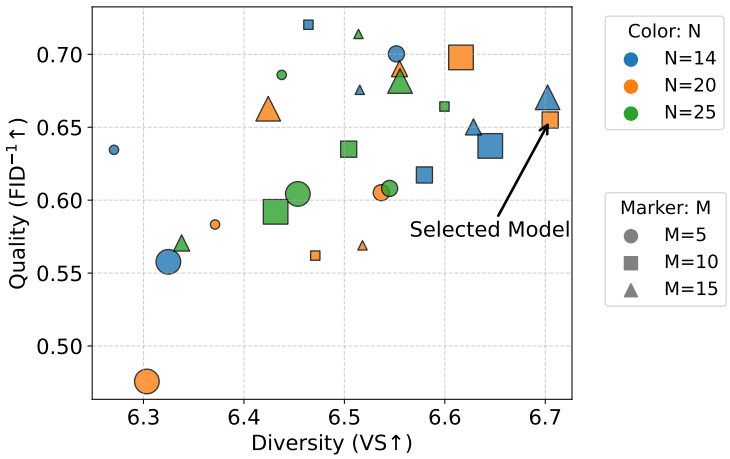


Figure 11: Ablation studies on the Chest X-Ray model by varying the number of nodes N, the number of graphs M, and the sparsity parameters ρ. Different colors show the N values (blue=15, orange=20, green=25). Different shapes show the M values (circle=5, square=10, triangle=15), and the size of the shapes shows the sparsity (biggest=0.88, middle=0.82, smallest=0.64). The final selected model is shown using the arrow labeled as 'Selected Model'.

generation with two prompts by CADS, Figure 15 present full 40 generation with two prompts by PAG, and Figure 16 present whole 40 generation with two prompts by Rainbow.

Supporting Figure 3 in the main text, Figure 17 presents full 40 generations with 4 sets of seasonal edges.

Figure 20 provides a qualitative and quantitative comparison of generated images between models with additional prompts.

Supporting Figure 6 in the main text, Figure 22 presents full 40 generations with 4 sets of seasonal edges.

(a) "Sunset scene with mountain" by SD3.5

(b) "Sunset scene with mountain in a specific season" by SD3.5

Figure 12: SD3.5 generates repeated layouts and objects in both prompts, producing unclear seasons or dominated late-autumn/early-winter aesthetics in the second prompt.

E.4 Addition prompts for Text-to-image task

Figure 20 provides a comparison of diverse natural-image generation between models in 4 additional prompts.

E.5 3D Brain MRI Fidelity Analysis

As visualized in Figure 27, Rainbow maintains sharper anatomical details across all age groups while avoiding artifacts. **LDM** introduces subtle distortions, particularly in age-sensitive regions:

- **Young (14 vs.16):** Rainbow preserves fine textures like developing white matter tracts. LDM's output shows mildly blurred cortical layers.

(a) "Sunset scene with mountain" by SD2-1

(b) "Sunset scene with mountain in a specific season" by SD2-1

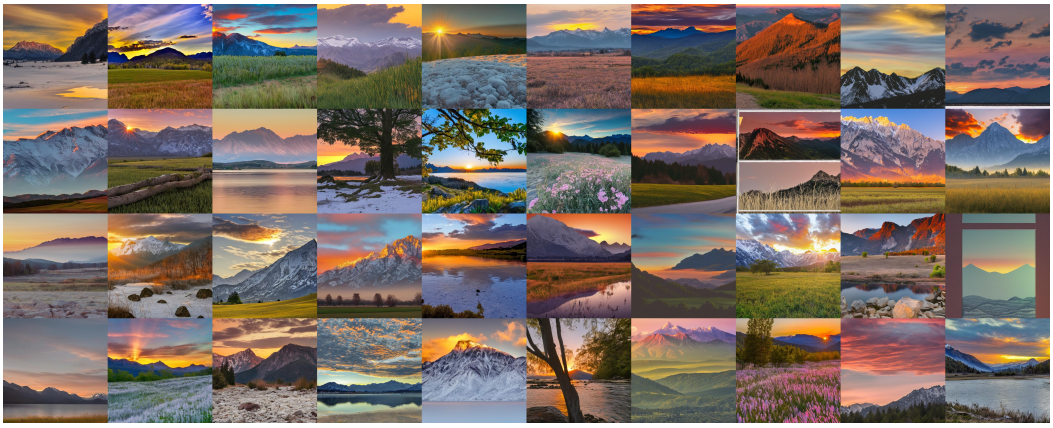
Figure 13: SD2-1 generates diverse layouts and objects but heavily prioritizes a specific art style in the first prompt, while producing ambiguous or spring-dominated seasons in the second prompt.

- **Middle-aged (45 vs.44):** Rainbow retains small vessels and tissue gradients. LDM exhibits "smeared" edges around ventricles.
- **Elderly (75 vs.72):** Rainbow realistically renders age-related atrophy (e.g., widened sulci). LDM generates unnaturally smooth brain surfaces, masking thinning cortex.

E.6 Qualitative Results for 3D MRI Counterfactual Generation Task

For brain MRI, we perform a counterfactual generation task. Given the factual image and condition, the task is to generate a counterfactual MRI based on a counterfactual condition on age or sex. Table 1 presents the numerical evaluation of the classification of sex and age based on the generated counterfactuals. Rainbow achieves higher performance, which underscores its effectiveness. Figure 29 compares the counterfactual generations. Both Rainbow and LDM generate correct patterns, such as smaller ventricles at a younger age (do(age=10)), evident from the red regions in the difference plot, and larger ventricles and cortex at an older age (do(age=80)), shown by the green regions. Additionally, sex conversion from male to female shows smaller ventricles and partially larger cortex regions. However, the baseline LDM exhibits some artifacts. These findings are consistent with previous studies on the effects of age and sex on MRI characteristics [15, 21, 81].

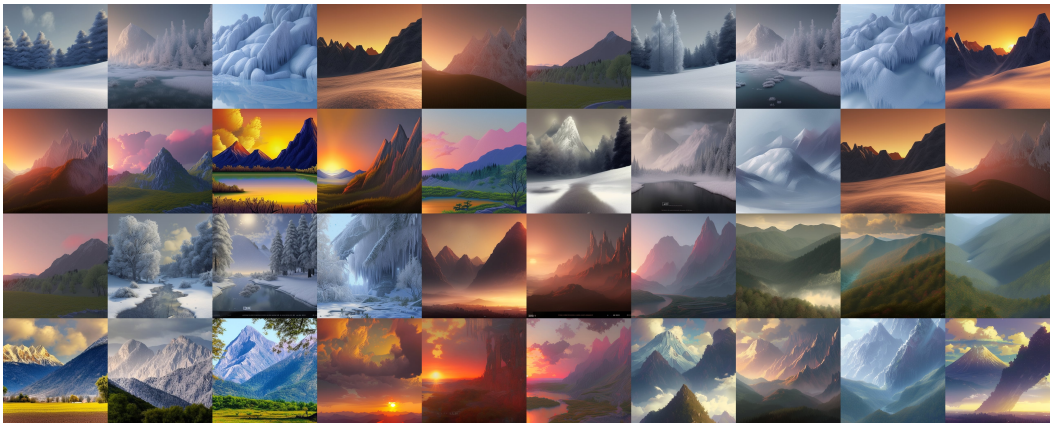
(a) "Sunset scene with mountain" by CADS



(b) "Sunset scene with mountain in a specific season" by CADS

Figure 14: CADS generates diverse layouts and objects, but prioritizes a specific art style in the first prompt. While CADS produces more defined seasonal characteristics in the second prompt, including generations with clear spring environments, some outputs retain ambiguity or disproportionately favor winter season.

(a) "Sunset scene with mountain" by PAG



(b) "Sunset scene with mountain in a specific season" by PAG

Figure 15: Generations by PAG

(a) "Sunset scene with mountain" by Rainbow

(b) "Sunset scene with mountain in a specific season" by Rainbow

Figure 16: Rainbow generates diverse layouts, objects, and seasonal environments with high compositional flexibility, achieving balanced variation across spring, summer, autumn, and winter visual details.

(a) "Sunset scene with mountain" + **Spring** edges

(b) "Sunset scene with mountain" + **Summer** edges

(c) "Sunset scene with mountain" + **Autumn** edges

(d) "Sunset scene with mountain" + **Winter** edges

Figure 17: Comparison of images generated by Rainbow with manipulated trajectories with 10 extra edges specific to each season. We observe that objects and layouts are consistent between the images, with clear season-specific details such as colorful spring flowers and green grass, hot-toned sky for summer, autumn yellow leaves, and snow for winter.

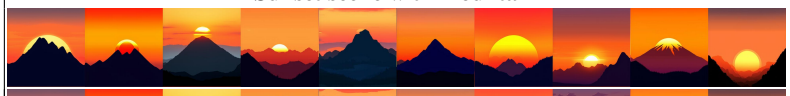
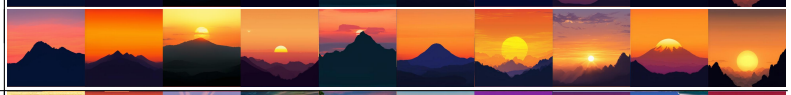
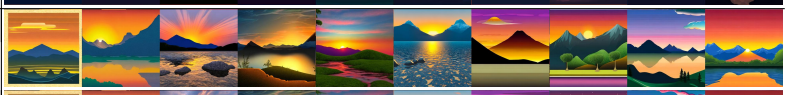
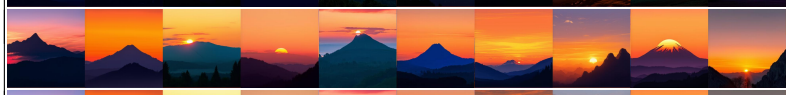
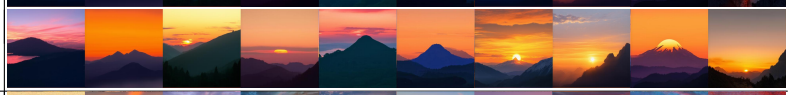
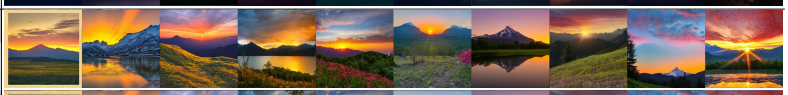
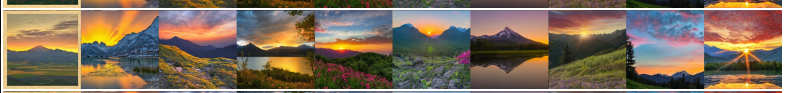
	CFG	IS ↑	VS ↑	
				"Sunset scene with mountain"
SD3.5	7.5	1.44	3.80	
	5.0	1.42	3.69	
	3.0	1.43	3.78	
SD2-1	7.5	1.96	4.29	
	5.0	1.92	4.53	
	3.0	2.07	5.24	
				"Sunset scene with mountain in a specific season"
SD3.5	7.5	1.20	3.06	
	5.0	1.40	3.54	
	3.0	1.46	3.85	
SD2-1	7.5	1.88	4.50	
	5.0	1.85	4.69	
	3.0	1.69	5.32	

Figure 18: Comparison of the effect of classifier-free guidance (CFG) scale on SD baselines across different prompts. For each prompt, 40 images are generated to compute metrics, and 10 are displayed. Metrics include the Inception Score (IS) for image quality and the Inception Vendi Score (Vendi) for diversity assessment. **Across models, CFG scales, and prompts, there is no consistent pattern suggesting that reducing CFG yields more diverse images.** Specifically, we can observe that decreasing the CFG only affects the detail level of the objects (more realistic, sharper) without influencing the context or object choices. Therefore, reducing CFG does not improve diversity in context, which is the major strength of the proposed Rainbow.

(a) DAC + SD2-1

(b) DAC + Rainbow

Figure 19: Diversity in image-editing task. Given the original image of a cat (left) and the editing prompt "A cat wearing a wool cap", the baseline mostly produces white caps, while Rainbow captures diverse color choices of caps.

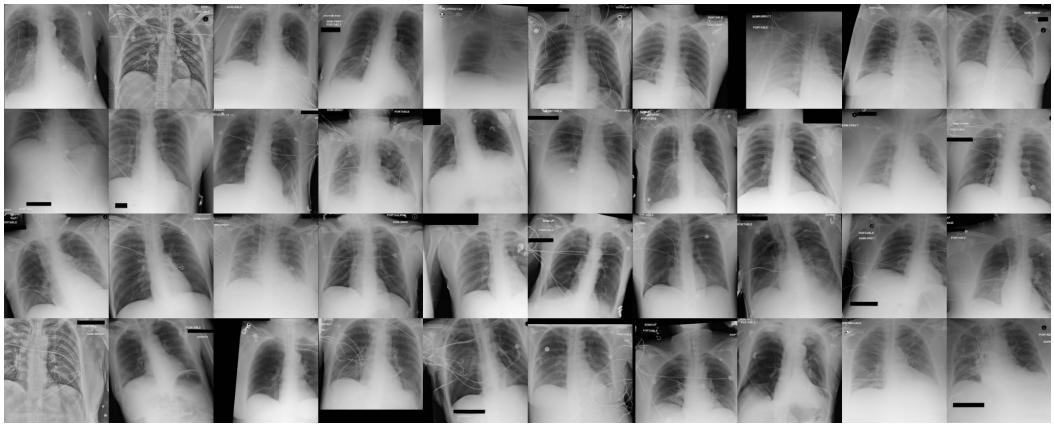
	Vendi ↑	CLIP ↑	
			"A forest with animals"
SD3.5	3.60	33.33	
SD2-1	4.77	33.59	
CADS	5.05	33.42	
PAG	4.84	31.76	
Rainbow	**6.09**	**33.66**	
			"Photo of San Francisco"
SD3.5	3.37	30.08	
SD2-1	4.77	**30.37**	
CADS	4.90	29.57	
PAG	4.64	29.56	
Rainbow	**5.46**	29.49	
			"Photo of brownies and lemonade"
SD3.5	3.96	34.88	
SD2-1	4.95	**35.27**	
CADS	5.50	34.90	
PAG	4.63	30.82	
Rainbow	**5.79**	33.39	
			"A grayscale tiger"
SD3.5	1.89	32.57	
SD2-1	1.94	32.31	
CADS	2.23	32.63	
PAG	**2.65**	**34.87**	
Rainbow	2.50	32.76	

Figure 20: Diversity and quality across Rainbow and SD baselines. Each prompt generates 40 images per model. Notably, Rainbow captures seasonal variations in the "forest with animals" prompt and multiple perspectives of San Francisco beyond seas and buildings in SD baselines. For "brownies and lemonade", Rainbow generates diverse relevant objects on the table, and for "tiger", it provides multiple views, showcasing enhanced versatility and realism. The SD baselines tend to generate similar layouts and contexts across prompts, demonstrating less diversity.

(a) Prompt: "A glass mug dropped to the ground"

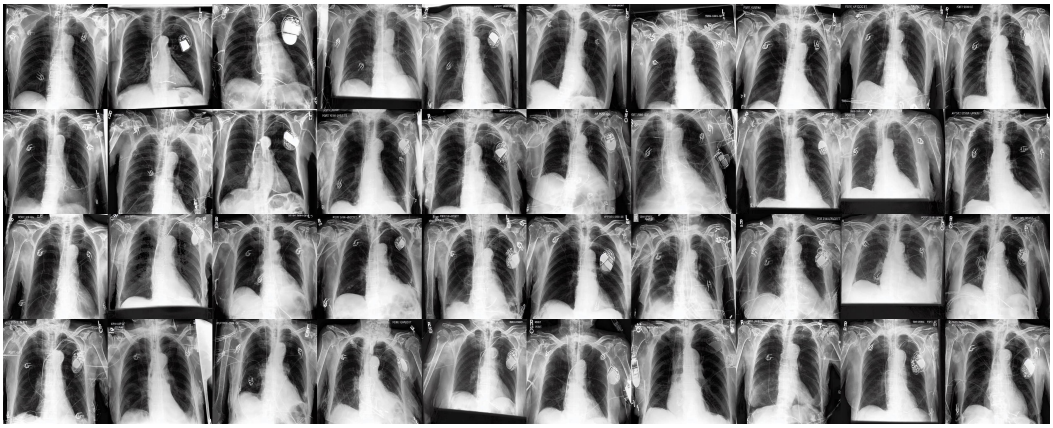
(b) Prompt: "A cow in a desert during a vibrant sunset"

Figure 21: Examples of cases that Rainbow perform not good.



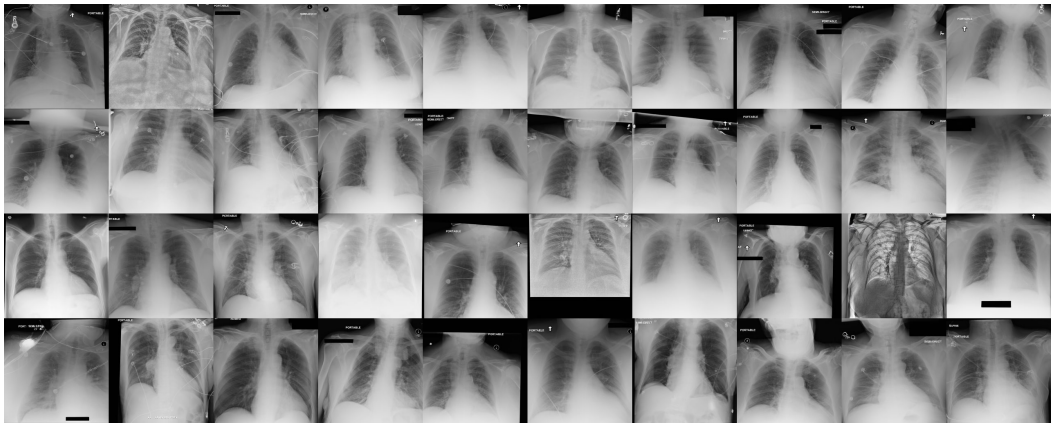
(a) Generations by RadEdit

(b) Generations by SD1.5

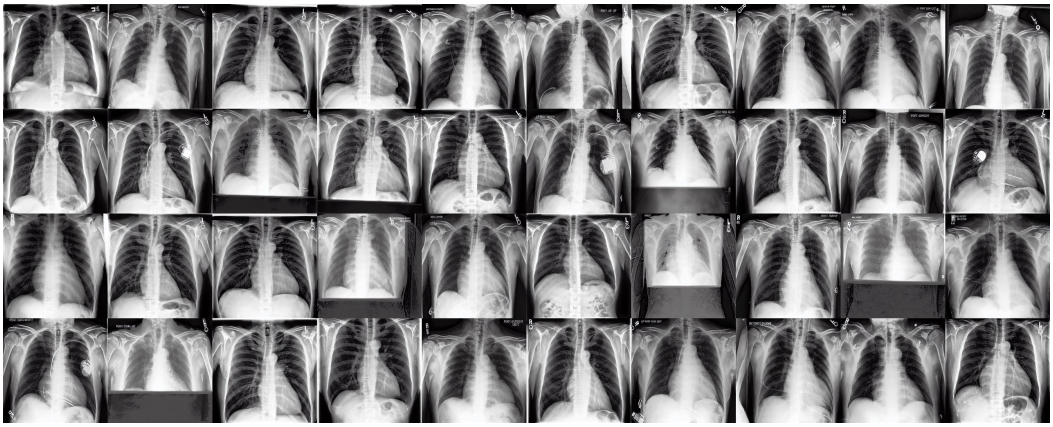


(c) Generations by Rainbow

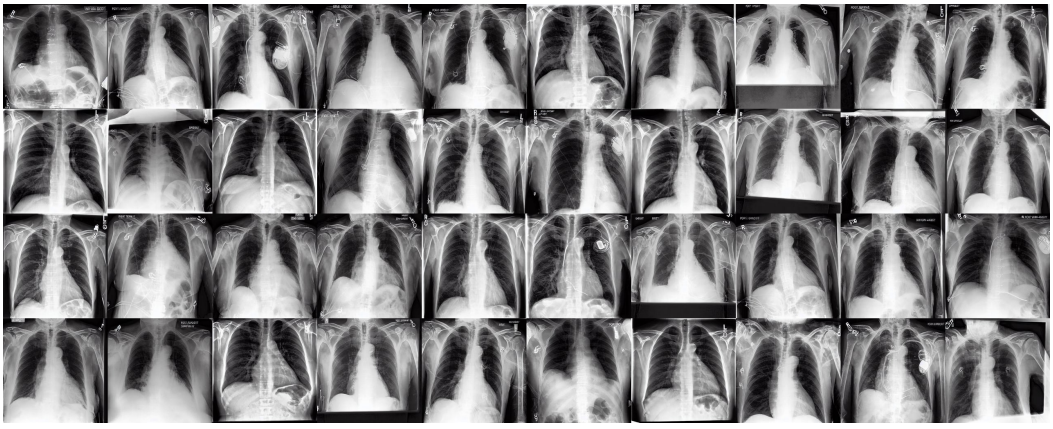
Figure 22: Generations with the prompt "Chest X-ray showing support devices." Rainbow generates a more diverse set of medical devices compared to the SD model and RadEdit, while maintaining image quality comparable to the SD model.



(a) Generations by RadEdit

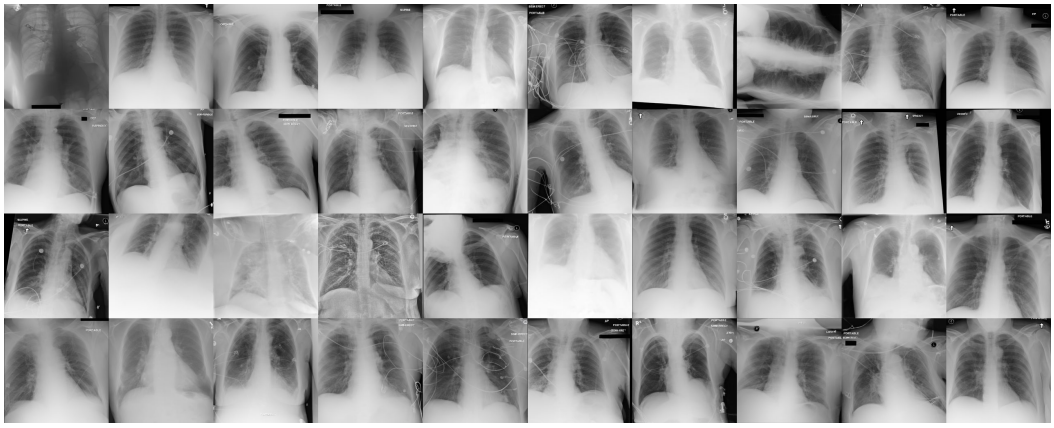


(b) Generations by SD1.5

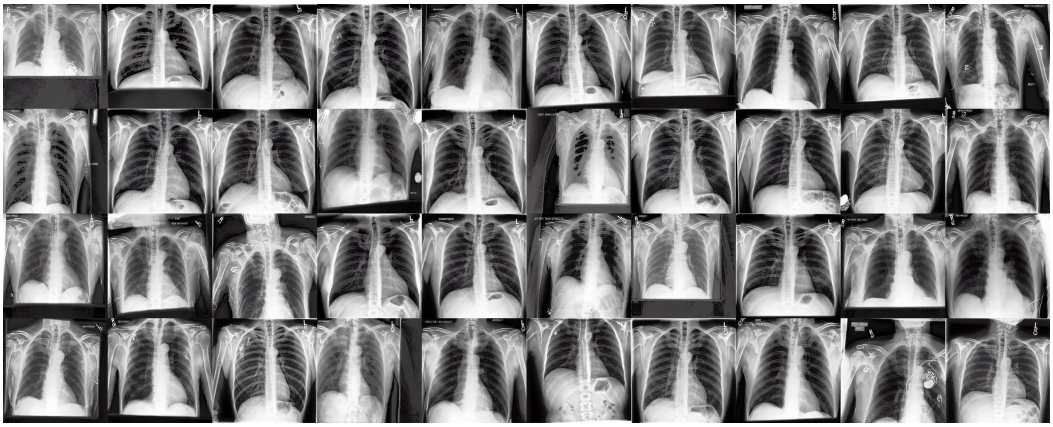


(c) Generations by Rainbow

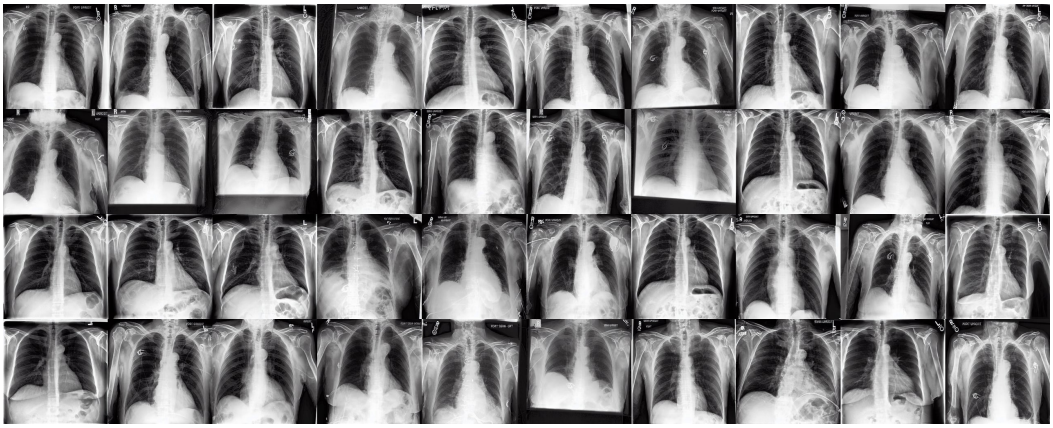
Figure 23: Generations with the prompt "Chest X-ray showing Cardiomegaly". Rainbow shows more diversity in the anatomy of the generated chest X-ray, while SD mostly generates left and right lungs that are similar to each other. Furthermore, Rainbow provides diversity in the location of the generated support devices.



(a) Generations by RadEdit



(b) Generations by SD1.5



(c) Generations by Rainbow

Figure 24: Generations with the prompt "Chest X-ray with no significant findings". Rainbow shows more diversity in the anatomy of the generated chest X-ray, while SD has less variation in the anatomical structure of the lungs.

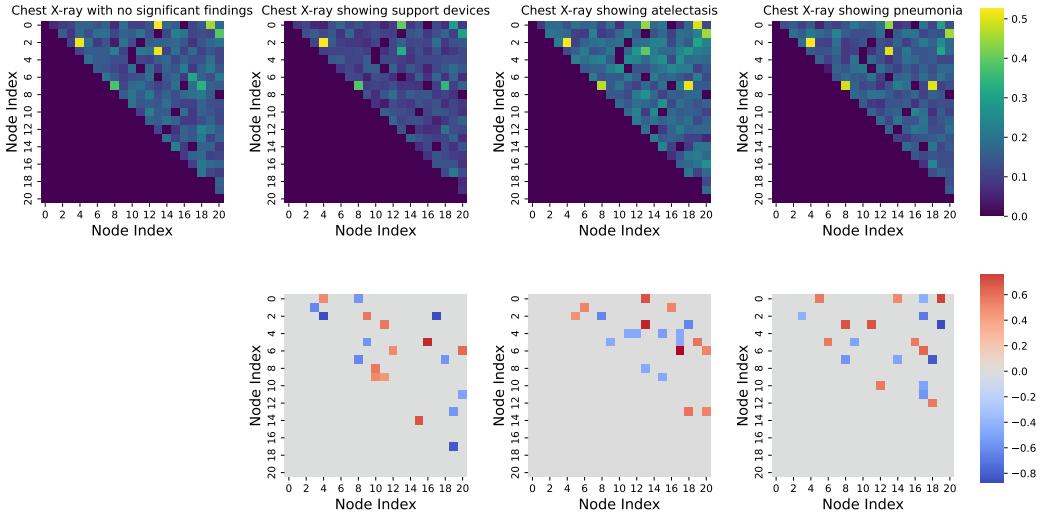


Figure 25: Heatmaps of edges across different prompts. In the first row, we present heatmaps of edges across 40 trajectories per prompt, with corresponding prompts labeled. In the bottom row, we illustrate the difference in edge distribution compared to the baseline prompt "Chest X-ray with no significant findings". We analyzed and extracted the 10 most frequently added edges and the 10 most frequently removed edges for the *difference heatmap*. Our observations reveal that (1) some edges consistently appear in most prompts with a significant proportion, and (2) it is evident that certain edges are representative of specific contexts. Particularly, in the difference heatmap, we can see certain edges are added with a high proportion, approximately 60%.

(a) Base prompt

(b) Base prompt + **suport devices** edges

Figure 26: Comparison of images generated by Rainbow with the base prompt "Chest X-ray with no specific findings" and with "support device" edges. By adding the edges corresponding to the entity "support devices" to the latent graph, we're able to modify the images with support devices. This demonstrates that Rainbow 's latent graphs encode structured and interpretable knowledge, and that manipulating these graphs enables fine-grained, concept-specific image editing.

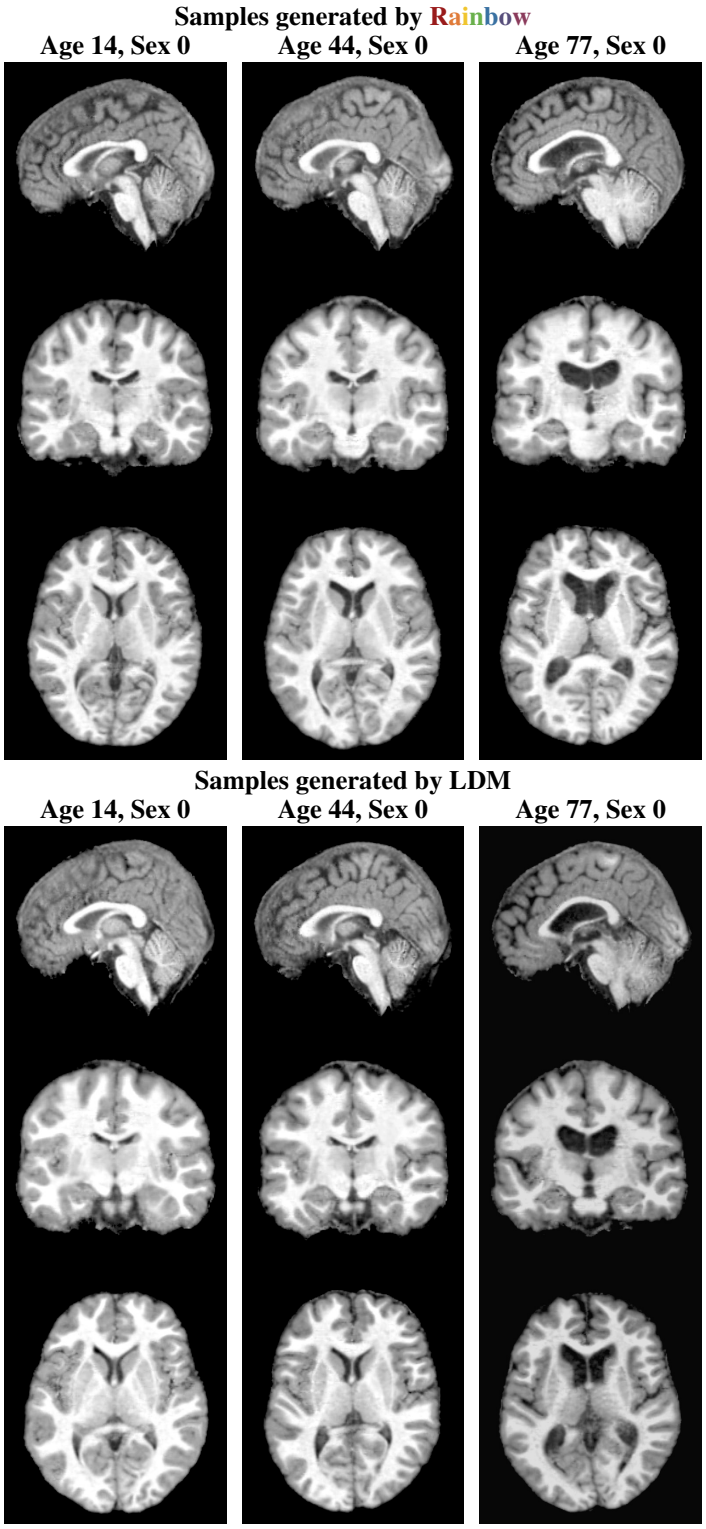
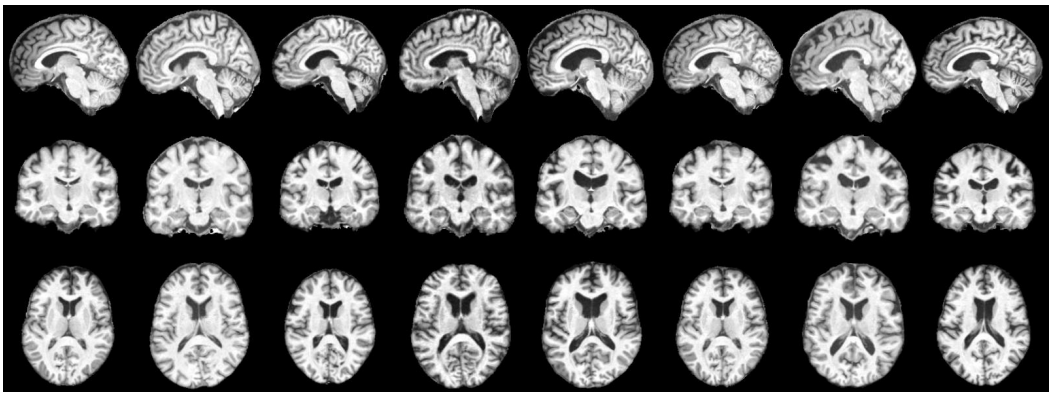
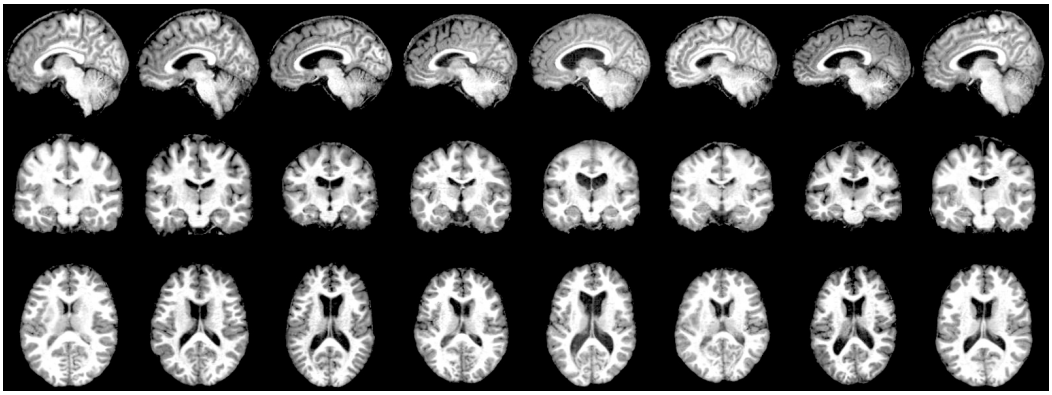


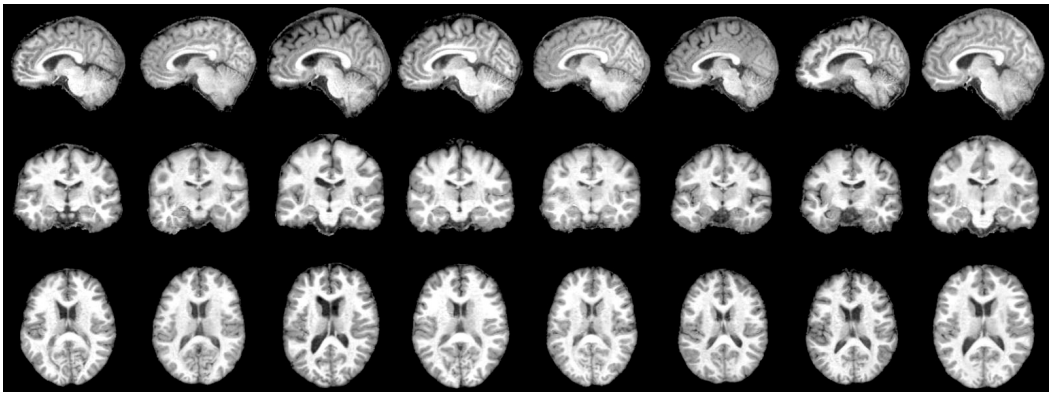
Figure 27: Brain MRI fidelity comparison



(a) Actual samples of male patients in age range from 60 to 65



(b) Generated MRI by Rainbow



(c) Generated MRI by baseline LDM

Figure 28: Multiple generations per input condition comparison. The figure displays MRI images showcasing 8 samples generated from a single input condition. Given the input condition of a 65-year-old male, both Rainbow and the baseline LDM can generate plausible MRI images. However, compared to actual samples from males aged 60 to 65, it is evident that Rainbow captures a greater diversity in details, such as varying ventricle sizes. In contrast, the baseline LDM tends to generate images with consistently similar ventricle sizes, demonstrating less diversity.

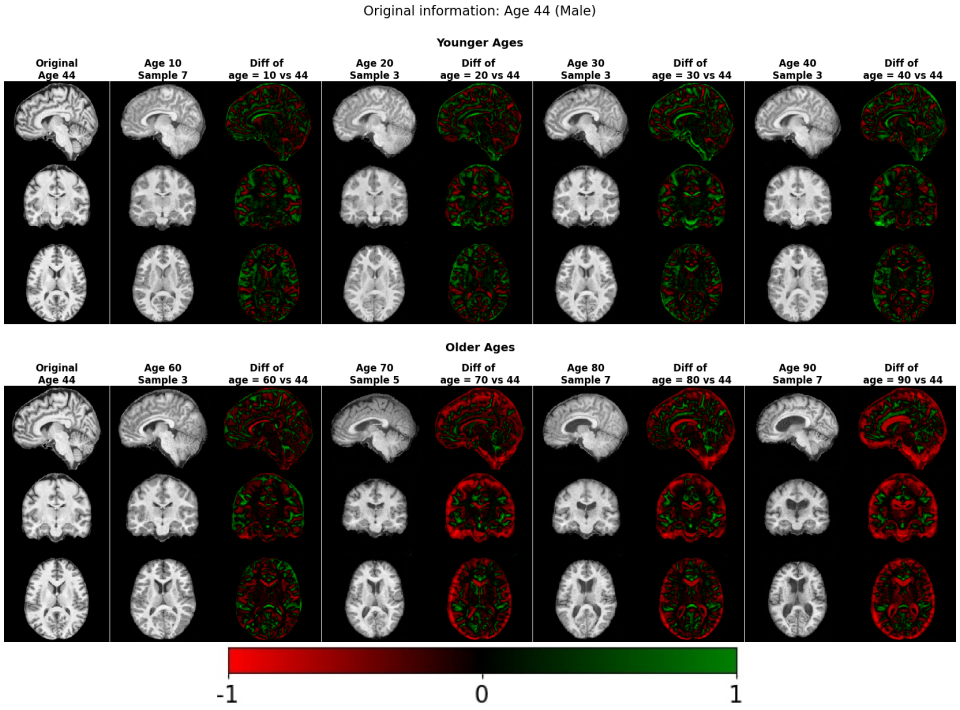


Figure 29: Counterfactual 3D Brain MRIs by Rainbow and the difference between the original image and generated counterfactuals

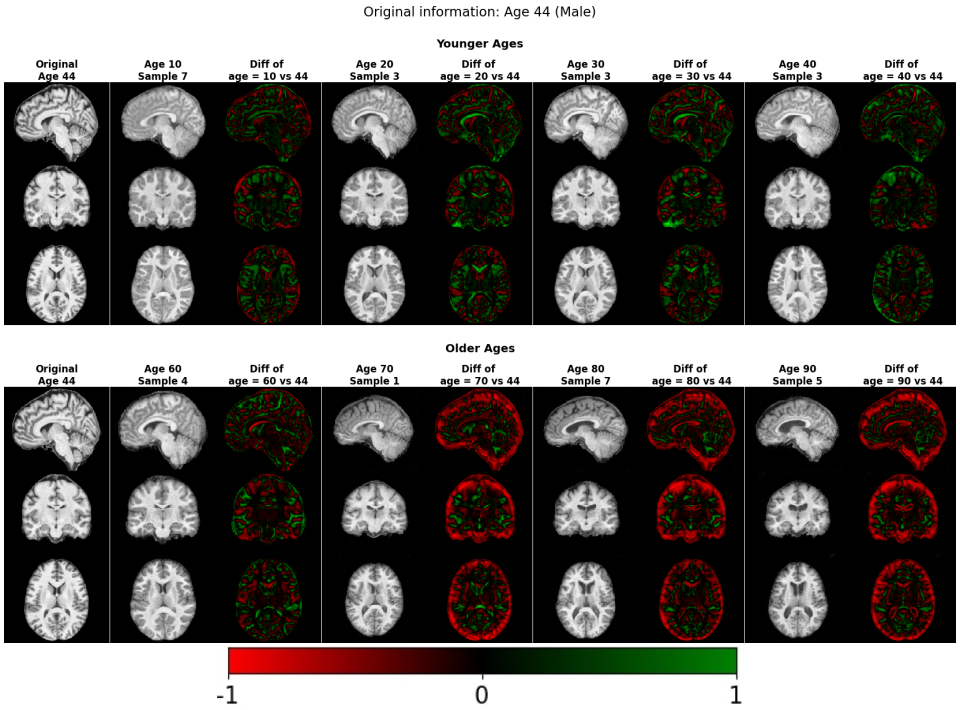


Figure 30: Counterfactual 3D Brain MRIs by baseline LDM and the difference between the original image and generated counterfactuals