PRODUCT OF EXPERTS FOR VISUAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern neural models capture rich priors and have complementary knowledge over shared data domains, e.g., images and videos. Integrating diverse knowledge from multiple sources—including visual generative models, visual language models, and sources with human-crafted knowledge such as graphics engines and physics simulators remains under-explored. We propose a probabilistic framework that combines information from these heterogeneous models, where expert models jointly shape a product distribution over outputs. To sample from this product distribution for controllable image/video synthesis tasks, we introduce an annealed MCMC sampler in combination with SMC-style resampling to enable efficient inference-time model composition. Our framework empirically yields better controllability than monolithic methods and additionally provides flexible user interfaces for specifying visual generation goals.

1 INTRODUCTION

Recent image and video generative models (Saharia et al., 2022; Rombach et al., 2022; Ho et al., 2022; Brooks et al., 2024) have achieved remarkable success in realistic appearance modeling, yet still have limitations in following complex text instructions and adhering to real-world constraints such as physical laws. The former would benefit from semantic priors in visual language models (VLMs) (Radford et al., 2021; Li et al., 2023; Bai et al., 2023; Achiam et al., 2023), and the latter from rules embedded in physics simulators. However, training a single model to absorb all information sources (texts/visual corpora, simulation trajectories, etc.) can be prohibitively expensive.

We address this challenge by integrating knowledge across models at inference time (Du & Kaelbling, 2024), leveraging visual generative priors, discriminative rewards from VLMs, and rule-based knowledge from physics simulators. We focus on the task of controllable visual generation and aggregate opinions across a set of "expert" models by sampling from the product distribution defined by the models. Each expert, in our case, pretrained models such as VLMs or image and video generative models, focuses on one or more constraints in generation, and the resulting product distribution naturally assigns high probabilities only to samples that satisfy all constraints simultaneously. The use of product distributions to combine the opinions of multiple experts has been extensively used in the past (Bacharach, 1972; Genest & Zidek, 1986; Hinton, 1999), and in the visual domain in Markov Random Forests (Wang et al., 2013; Kolmogorov & Zabih, 2002; Glocker et al., 2008; Boykov & Funka-Lea, 2006) and Conditional Random Fields (Boykov & Jolly, 2001; Kumar & Hebert, 2003; He et al., 2004).

Despite its conceptual appeal, sampling from product distributions is often intractable. Straightforward approaches like rejection sampling are inefficient due to vanishing acceptance rates in high dimensions. Recent works have gotten around these difficulties by either composing model distributions in a simpler Gaussian latent space (Huang et al., 2022) or by composing through models through an annealed Langevin procedure (Du et al., 2020; 2023; Geffner et al., 2023), where the combination of annealing and gradient-based sampling enables the modes of the product probability distribution to be effectively identified. In this paper, we provide a more general framework that enables us to sample from the product of both autoregressive and diffusion-based generative models, as well as discriminative models such as VLMs. Our approach relies on a combination of Annealed Importance Sampling (Neal, 2001) and Sequential Monte Carlo (Doucet et al., 2001a) to effectively find the modes of the product distribution across all experts.

Our overall framework enables us to integrate multiple input constraints without costly retraining. We empirically validate its benefits in image generation following complex text instructions. It also naturally yields a flexible user interface for defining complex generative goals by selecting and configuring experts, e.g., allowing users to specify the pose and motion trajectory of an object of interest, insert it into an existing image, and animate the full scene.

Overall, our contributions are threefold:

- **i)** We formulate controllable visual synthesis tasks in a unified Product-of-Experts (PoE) framework that enables principled knowledge integration of heterogeneous generative and discriminative models, as well as physics simulators.
- **ii)** We propose a practical, computationally efficient sampling framework based on Annealed Importance Sampling (AIS) and Sequential Monte Carlo (SMC).
- **iii)** Our method enables flexible forms of user-instruction-following (texts, images, and low-level specifications such as object poses and trajectories) for image/video synthesis applications and achieves better user controllability and output fidelity compared to baseline methods.

2 Related Works

Compositional Generative Modeling. Prior works on compositional generative modeling typically combine multiple generative models to jointly generate data samples (Du et al., 2020; Garipov et al., 2023; Du et al., 2023; Huang et al., 2022; Mahajan et al., 2024; Du & Kaelbling, 2024; Bradley et al., 2025; Thornton et al., 2025; Gaudi et al., 2025) and has been previously applied in visual content generation (Liu et al., 2022a; Bar-Tal et al., 2023; Zhang et al., 2023; Yang et al., 2024; Su et al., 2024; Li et al., 2022), but such applications typically focus on multiple homogeneous generator experts, or, in the case of Li et al. (2022), a single generator expert with multiple discriminative experts. In contrast, our approach provides a general probabilistic framework through which many heterogeneous generative and discriminative experts can be jointly combined in generation.

Reward Steering from Discriminative Models. Our work is further related to recent work steering generative models with discriminative models and reward functions. Methods using gradient descent on a discriminative reward (Grathwohl et al., 2019; Dhariwal & Nichol, 2021; Bansal et al., 2023; Luo et al., 2025; He et al., 2023; Ye et al., 2024; Song et al., 2023; Rout et al., 2025) assume dense, differentiable gradients on the reward. Classifier-free guidance (CFG) (Ho et al., 2020) alleviates the need for training explicit reward models (classifiers) and we defer more detailed discussions to Section F.1. Another branch of work applies SMC for more exact reward steering (Wu et al., 2023; Zhao et al., 2024; Singhal et al., 2025). Both Skreta et al. (2025); He et al. (2025) and our work build on top of this formulation and consider composing both generative and discriminative experts, but with the following difference: when considering generative expert products (Eq. (2)), these works calculate *path-wise* importance weights to ensure sampling correctness, risking weight degeneracy as path length grows (Skreta et al., 2025), while ours uses *per-timestep* MCMC kernels that leave p_t (Eq. (3)) invariant without accumulating weights.

Video Generation with Physical Simulators. Our framework integrates knowledge from physical simulators to improve physical accuracy compared to end-to-end models, while alleviating the need for tedious 3D scene setups in traditional graphics rendering pipelines. Recent works have explored such direction typically convert physical simulator outputs into a specific type of conditional signals such as optical flow (Burgert et al., 2025; Montanaro et al., 2024; Yang et al., 2025) and point tracking (Gu et al., 2025), or rely on a single pre-trained video generation model (Liu et al., 2024a; Chen et al., 2025; Tan et al., 2024), while our framework allows for the integration of various signals available from physical simulation, providing more complete specifications of controls.

3 Method

We aim to generate complex scenes $x \in \mathbb{R}^d$ by combining the priors from both generative models (such as image/video models conditioned on texts or control signals converted from physics simulators) as well as discriminative models (such as VLMs), with examples detailed in Section 4. We formulate this composition of models in a probabilistic manner, where generative models are represented as probabilistic priors over data $\{p^{(i)}(x)\}_{i=1}^N$. Discriminative models are written as constraint

functions, $\{r^{(j)} : \mathbb{R}^d \to \mathbb{R}\}_{j=1}^M$, each assigning a scalar reward that represents how much a sample x matches the constraint encoded in the model. Each constraint function is converted into an unnormalized probability distribution through the Boltzmann distribution, $q^{(j)}(x) := (\exp r^{(j)})(x)$.

To combine these two classes of models, we aim to sample from the product distribution:

$$x \sim p(x) :\propto \prod_{i=1}^{N} p^{(i)}(x) \prod_{j=1}^{M} q^{(j)}(x), \tag{1}$$

where each generative expert $p^{(i)}(x)$ and discriminative expert $q^{(j)}(x)$ are defined over parts of the scene x. The product distribution $p(x)$ has high probability for a sample x when x has high probability under both $p^{(i)}(x)$ and $q^{(j)}(x)$ (Hinton, 2002; Du et al., 2020; Du & Kaelbling, 2024), allowing us to compose together the priors in both generative and discriminative models.

However, sampling from Eq. (1) is challenging, as we need to search for samples that satisfies constraints across all experts. Below, we will discuss how we can efficiently and effectively sample from the Eq. (1) for high fidelity visual generation. We first discuss how we can sample from multiple generative experts in Section 3.1 and then discuss how to sample jointly from multiple discriminative and generative experts in Section 3.2, followed by a practical implementation in Section 3.3.

3.1 Sampling from Generative Experts

We first discuss how we can effectively sample from the product of a set of generative experts,

$$x \sim p(x) :\propto \prod_{i=1}^{N} p^{(i)}(x). \tag{2}$$

To sample from the above distribution, we can use Markov chain Monte Carlo (MCMC) Robert et al. (1999), where we iteratively refine samples based on their likelihood under the product distribution. In discrete settings, a variant of Gibbs sampling may be used, while in continuous domains, Langevin sampling based on the gradient of log density may be used (Welling & Teh, 2011).

A central issue, however, is that MCMC is a local refinement procedure and can take an exponentially long time for the sampling procedure to mix and find a high likelihood scene x from the product distribution (Robert et al., 1999). To effectively sample from product of generative experts, we propose to use Annealed Importance Sampling (AIS) (Neal, 2001) and construct of a sequence of T distributions, $\{p_t(x)\}_{t=T}^1$, where $p_T(x)$ is a smooth, easy-to-sample distribution and $p_1(x)$ is the desired product distribution defined in Eq. (2). To draw a sample from $p_1(x)$, we first initialize a sample from $p_T(x)$ and iteratively run MCMC on the sample to sample from each intermediate distribution before finally reaching $p_1(x)$. Since intermediate distributions are easier to sample from, this enables us to more effectively find a high likelihood scene x from the product distribution.

We construct the intermediate probability distributions of the following form:

$$p_t(x) :\propto \prod_{i=1}^{N} p_t^{(i)}(x), \tag{3}$$

where $p_t^{(i)}(x)$ is an expert-specific distribution interpolating between an initial distribution (e.g., Gaussian or uniform) and the final expert distribution $p^{(i)}(x)$. We list concrete choices of $p_t^{(i)}(x)$ for two common classes of generative models in Section 3.3. For autoregressive models (Oord et al., 2016; Larochelle & Murray, 2011), it is the marginal distribution for a prefixed data region determined by t; for diffusion/flow models (Liu et al., 2022b; Lipman et al., 2022; Ho et al., 2020; Sohl-Dickstein et al., 2015), it is the time marginal distribution with estimated score functions.

Composing Conditional Generative Experts. To improve the efficacy of sampling from the product distribution of generative experts, we can modify each generative expert to be conditionally dependent on the outputs of other experts. Specifically, we can represent the product distribution as

$$x \sim p(x) :\propto \prod_{i=1}^{N} p^{(i)}(x_i | x_{\text{pa}(i)}), \tag{4}$$

Algorithm 1 Product Distribution Sampling

Input: Annealing length T, initial distribution $p_T(\cdot)$, particle count L, MCMC step count K, generative experts $\{p_t^{(i)}(\cdot)\}_{i=1,t=1}^{N,T}$ with kernels $\mathcal{K}_{t\leftarrow t+1}(\cdot \mid \cdot)$ for MCMC initialization and $\mathcal{K}_t(\cdot \mid \cdot)$ for MCMC steps, discriminative experts with log probabilities $\{r^{(j)}(\cdot)\}_{j=1}^{M}$.
Initialize: Sample $x^{(l)} \sim p_T(\cdot), \forall l = 1, \cdots, L$.
for $t = T-1$ to 1 **do** ▷ Transition to the next annealed distribution
  for $l = 1$ to L **do** ▷ Parallel sampling
    $x^{(l)} \leftarrow \mathcal{K}_{t\leftarrow t+1}(\cdot \mid x^{(l)})$ ▷ MCMC initialization
    for $k = 1$ to K **do**
      $x^{(l)} \leftarrow \mathcal{K}_t(\cdot \mid x^{(l)})$ ▷ MCMC step
    end for
  end for
  Resample samples with log weights $\sum_{j=1}^{M} r_t^{(j)}(x^{(l)})$ where $r_t^{(j)}(x^{(l)}) \approx r^{(j)}(\hat{x}^{(l)})$.
end for
$l^* \leftarrow \max_l \sum_{j=1}^{M} r^{(j)}(x^{(l)})$
Output: $x^{(l^*)}$

where each x_i refers to the part of a scene x represented by the generative expert $p^{(i)}$, and $x_{\text{pa}(i)}$ refers to the other parts of a scene specified by other experts the expert is conditioned on.

Making each generative expert conditionally dependent on the values of other experts reduces the multi-modality in the distribution $p^{(i)}(x)$, enabling more efficient MCMC sampling on the product distribution. Its benefit is empirically shown in Fig. 2 and Table 1.

3.2 Parallel Sampling with Discriminative Experts

Next, we discuss how we can modify the annealed sampling procedure in Section 3.1 to sample from the full product distribution from Eq. (1), including discriminative experts. Similar to the previous section, we can define a sequence of intermediate sampling distributions

$$p_t(x) :\propto \prod_{i=1}^{N} p_t^{(i)}(x) \prod_{j=1}^{M} q_t^{(j)}(x). \tag{5}$$

To implement AIS, one simple approach is importance sampling as follows. We first obtain L samples from the product of the intermediate generative expert distributions defined in Eq. (3) according to Section 3.1, and then draw a weighted random sample from these L samples, each with importance weight $\prod_{j=1}^{M} q_t^{(j)}(x)$ corresponding to the likelihood under product of discriminative experts.

An issue with this simple procedure is that each sample drawn from the MCMC may be correlated with each other, since MCMC is slow at mixing and may not cover the entire generative product distribution. To improve coverage, we use a parallel SMC sampler (Doucet et al., 2001b), where we maintain L particles over the course of annealing. At each intermediate distribution, we run MCMC on each particle to draw samples from the product of the generative distributions, and then weigh and resample particles based on their likelihood under the product of the discriminative experts.

Our overall algorithm requires only black-box access to each discriminator, requiring knowing only the likelihood the discriminator assigns to a sample x and not additional information such as log-likelihood gradients. If we do have more information about the discriminative expert's form, we can directly treat it as generative as in Section 3.1 and use approaches such as gradient-based MCMC.

3.3 Framework Instantiation

We now introduce a specific implementation of the framework as summarized in Algorithm 1, which describes the annealed distribution and expert instantiations primarily used in the paper.

Generative Experts. The framework requires specifying the annealed generative distributions factors $\{p_t^{(i)}(x)\}_{t=T}^{1}$ (Eq. (3)), a kernel $\mathcal{K}_{t\leftarrow t+1}(\cdot \mid \cdot)$ to propagate samples from the previous an-

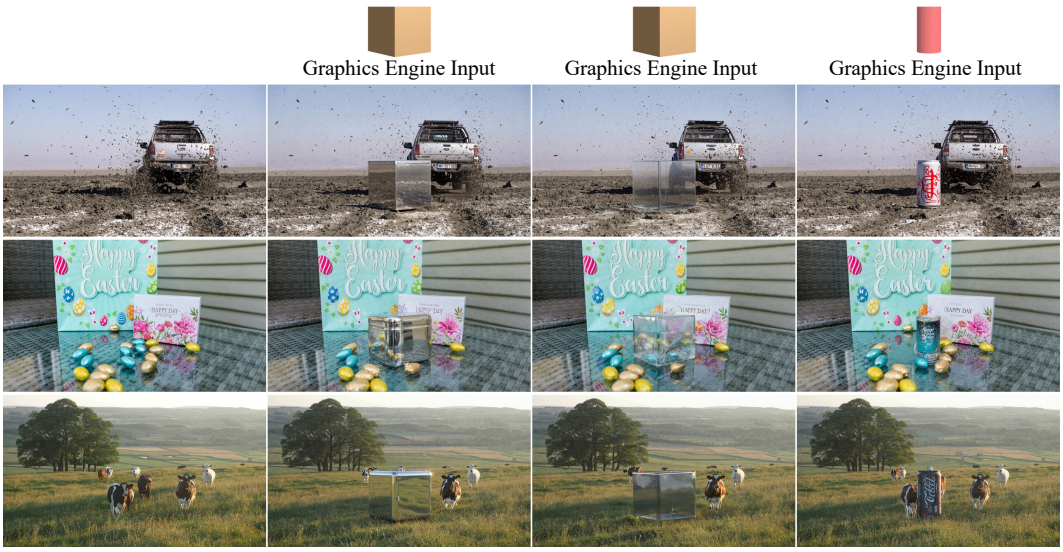


Figure 1: **Application on Image Object Insertion** where the goal is to insert assets posed in a graphics engine (top row) and described with text prompts (bottom row) into images (first column). nealed distribution $p_{t+1}(x)$ to the next distribution $p_t(x)$ (line 5, Algorithm 1), and a MCMC kernel $\mathcal{K}_t(\cdot \mid \cdot)$ that leaves p_t invariant (line 7). While $\mathcal{K}_{t \leftarrow t+1}$ does not draw a sample from $p_t(x)$ exactly, it serves as an effective initialization for MCMC sampling.

When generative expert $p^{(i)}(x)$ is a flow model, a natural choice for $\{p_t^{(i)}(x)\}_{t=T}^{1}$ is a discretization of the probability path generated by its velocity prediction $v_t^{(i)}(x)$. We set $\mathcal{K}_{t \leftarrow t+1}$ as one Euler step in the flow ODE integration, and $\mathcal{K}_t$ is the Langevin dynamics transition, both under composed score $\sum_i \nabla_x \log p_t^{(i)}(x)$ (Section C.1). When $p^{(i)}$ is an autoregressive model, we use its prefix marginal for annealing, namely $p_t^{(i)}(x) = p^{(i)}(x_{1:T+1-t})$ such that $x_{1:T} = x$, where $x_{1:T+1-t}$ is a data slice. We set $\mathcal{K}_{t \leftarrow t+1}$ such that it appends the next data slice sampled from the model-predicted conditional distribution, and $\mathcal{K}_t$ can be implemented as Gibbs sampling.

Conditional Sampling. We implement conditional generative experts $p_t^{(i)}(x_i \mid x_{\mathrm{pa}(i)})$ for flow models by modifying the original generative experts $p_t^{(i)}(x_i)$. In particular, we modify the generative flow $v_t^{(i)}(x_i)$ to take into account the predicted flow at parent regions

$$v_t^{(i)}(x_i \mid x_{i'}) \approx v_t^{(i)}(x_i) - w \sum_{i' \in \mathrm{pa}(i)} \nabla_{x_i} \| v_t^{(i)}(x_i) - \mathrm{stopgrad}(v_t^{(i')}(x_{i'})) \|_2^2, \tag{6}$$

where w is the learning rate. Justifications of this approximation are deferred to Section C.2.

Discriminative Experts. We use pre-trained VLMs as reward functions $r^{(i)}(x)$. When generative experts are flow models, the noisy samples x are out of distribution for discriminative experts. We define the intermediate counterparts in Eq. (5) as $r_t^{(j)}(x) = r^{(j)}(\hat{x})$ where $\hat{x}$ is the predicted endpoint of x (Chung et al., 2022; Efron, 2011) whose computation is specified in Eq. (8).

4 Experiments

We apply our framework to image/video synthesis tasks. Expert configurations can adapt to the desired, task-dependent input granularity, including high-level text instructions (Sections 4.1 to 4.3) and precise, low-level pose controls (Sections 4.1 and 4.2), and leverage expert knowledge ranging across natural image and video priors (Sections 4.1 to 4.3), precise physics rules from physical simulators (Section 4.2), and semantic visual understandings from VLMs (Section 4.3). To better quantify sampling quality, we evaluate the method on synthetic data with known ground truth target distribution in Section D.1.

Input Image & Rendering | Ours | RF-Solver | Ours (No Conditional)

Figure 2: **Image Object Insertion Comparisons.** Our method better adheres to input geometric conditions while faithfully preserving background details. The last column shows that conditional sampling improves visual harmonization and fidelity.

Methods	Controllability			Image Quality		Text Alignment	
	MSE (bg) (↓)	LPIPS (fg) (↓)	GPT-4o (↑)	Aesthetic (↑)	GPT-4o (↑)	ImageReward (↑)	GPT-4o (↑)
Graphics Engine Rendering Input							
RF-Solver	1.619±0.605	0.178±0.061	0.518±0.168	**0.618**±0.070	0.662±0.103	0.948±0.777	0.438±0.179
Ours No Cond	1.511±0.661	**0.065**±0.018	0.727±0.141	0.599±0.059	0.718±0.120	1.142±0.668	0.675±0.180
Ours	**1.429**±0.602	**0.065**±0.019	**0.827**±0.073	0.596±0.061	**0.783**±0.082	**1.175**±0.666	**0.817**±0.129
Character Rendering Input (Magic Insert (Ruiz et al., 2024) Dataset)							
Magic Insert	0.769±0.535	0.106±0.042	0.743±0.142	0.721±0.076	0.758±0.130	1.169±0.701	0.649±0.141
Add-it	0.710±0.473	0.128±0.048	0.763±0.126	0.715±0.066	0.778±0.097	1.078±0.708	0.684±0.155
FLUX-Fill	**0.058**±0.044	0.103±0.042	0.655±0.147	0.691±0.070	0.746±0.092	0.828±0.777	0.590±0.152
SDEdit	0.968±0.965	**0.026**±0.020	0.744±0.131	0.724±0.074	**0.871**±0.061	1.640±0.376	0.658±0.144
Ours	0.365±0.247	0.064±0.033	**0.818**±0.097	**0.760**±0.060	0.769±0.118	**1.711**±0.308	**0.763**±0.107

Table 1: **Image Object Insertion Evaluation.**

4.1 Graphics-Engine-Instructed Image Editing

Task. This task provides the following image object insertion interface for applications where users want to insert an object precisely into certain parts of the image. Inputs consist of an image to be edited, an input 3D asset posed in a graphics engine, and a text description describing higher-level information such as object materials ("metal") and semantics ("coke can"). We deploy two generative experts for this task: a depth-to-image model, FLUX.1 Depth [dev], and an inpainting model, FLUX.1 Fill [dev] (FLUX, 2024). The former ensures inserted objects follow input pose specifications, and the latter provides a natural image prior to produce realistic outputs.

Evaluation. We compare with an editing method, RF-Solver (Wang et al., 2024), which inverts the input image to a Gaussian noise and then generates the output starting from that noise, conditioned on a text prompt that additionally describes where to insert the object, e.g., "a metal box standing on the ground". The evaluation dataset consists of 10 natural images paired with 3 object assets, resulting in 30 scenes in total, with examples in Fig. 1. The metrics cover three major aspects: controllability, image quality, and text alignment. Specifically, we evaluate the MSE distance between generation outputs and input images on background pixels to evaluate background preservation, and LPIPS (Zhang et al., 2018) distance between outputs and graphics engine's RGB renderings on foreground pixels to measure the fidelity to input objects. We report aesthetic score using the LAION-Aesthetic predictor (Schuhmann & Beaumont, 2022) and measure text alignment with ImageReward (Xu et al., 2023), and query GPT-4o for automatic evaluation (details in Section E.2).

Results are in Table 1 (top half). Baseline outputs do not conform to input object pose due to the lack of precision of text prompts, and may fail to preserve input background as observed in Fig. 2.

Ablation on Conditional Sampling. Section 3.1 introduced the conditionally sampling strategy, which is crucial for efficient sampling from the product distribution. Removing the updates in Eq. (6) ("No Cond" in Table 1) decreases performance and results in less harmonized visual outputs (Fig. 2).

Character Insertion. The task spec above is closely relevant to the character insertion task studied in Magic Insert (Ruiz et al., 2024), where the goal is to insert a character from an image into a background image. We use 10 background images paired with 8 character inputs released on their official demo page to construct an evaluation dataset of 80 scenes. Since their model uses a different

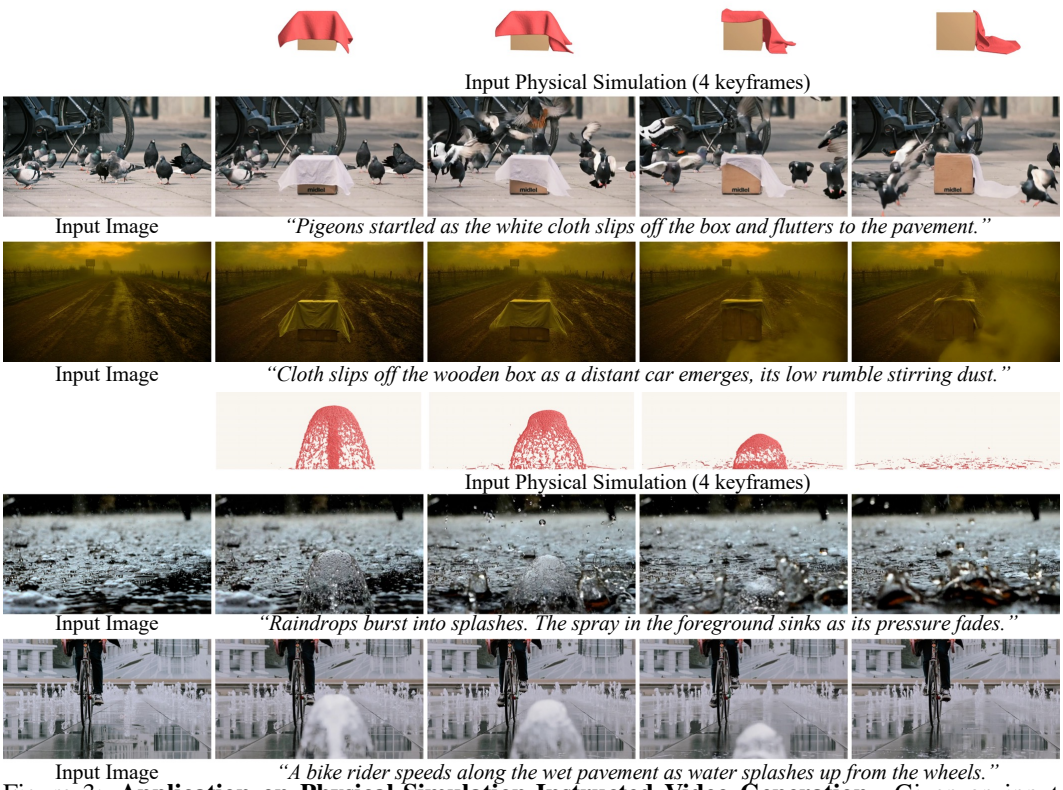


Figure 3: **Application on Physical-Simulation-Instructed Video Generation.** Given an input image and a physical simulator rendering describing precise object motions, our method generates videos aligned with input motions while synthesizing natural content for non-foreground regions.

Methods	Controllability			Video Quality				Semantic Alignment	
	IoU (fg) (↑)	LPIPS (↓)	GPT-4o (↑)	Smooth (↑)	Aesthetic (↑)	Imaging (↑)	GPT-4o (↑)	ViCLIP (↑)	GPT-4o (↑)
Object-Centric Simulation Input									
Traj2V	0.602±.193	0.109±.033	0.638±.172	0.993±.003	0.534±.071	0.570±.154	0.725±.179	0.258±.041	0.757±.141
Depth2V	**0.787**±.229	**0.103**±.031	0.650±.144	0.984±.015	0.534±.069	0.581±.155	0.750±.126	0.261±.041	0.775±.130
Image2V	0.321±.249	0.111±.021	**0.708**±.104	0.978±.025	**0.553**±.057	0.611±.101	0.775±.099	0.255±.036	0.788±.079
Ours	0.739±.221	0.104±.030	**0.708**±.104	**0.994**±.003	0.549±.068	**0.625**±.117	**0.842**±.132	**0.270**±.038	**0.817**±.080
Full-Scene Simulation Input (PhysGen3D (Chen et al., 2025) dataset)									
PhysG3D	–	–	0.438±.099	**0.995**±.002	0.571±.114	0.673±.046	0.513±.117	0.224±.026	0.588±.117
Inversion	–	0.237±.084	0.263±.122	0.993±.002	0.468±.097	0.410±.144	0.250±.206	0.202±.058	0.400±.206
Depth2V	–	0.164±.065	0.550±.173	0.993±.003	0.579±.067	0.680±.081	0.662±.216	**0.242**±.030	0.788±.108
Image2V	–	0.242±.077	0.525±.192	0.986±.010	**0.594**±.074	0.662±.052	0.638±.245	0.236±.028	0.788±.105
Ours	–	**0.136**±.047	**0.587**±.220	0.993±.003	0.575±.071	**0.688**±.045	**0.763**±.132	0.239±.038	**0.825**±.141

Table 2: **Physics-Simulator-Instructed Video Generation Evaluation.**

backbone (Podell et al., 2023), for fair comparisons, we also include three baselines using the same FLUX backbone as ours: Add-it (Tewel et al., 2024), SDEdit (Meng et al., 2021), and FLUX-Fill.

Results are in Table 1 (bottom half). Add-it receives text instructions for object insertion, but tends not to follow the specified character location in the prompts; SDEdit produces harmonized outputs but does not closely preserve the background due to the global noising operation; the inpainting method, FLUX-Fill, sometimes ignores the character descriptions in the input text instruction and fails to insert any object. These observations are also reflected qualitative samples in Section D.3.

4.2 Physical-Simulator-Instructed Video Generation

Task. The setup from Section 4.1 can be directly extended to dynamic scenes for video generation, with object poses and dynamics (e.g., a ball bouncing, Fig. 3 top row) specified by a physical simulator, an input image, and a textual scene description. Below, we consider two sets of experts: flow-based (Wang et al., 2025) and autoregressive-based (Zhang & Agrawala, 2025).

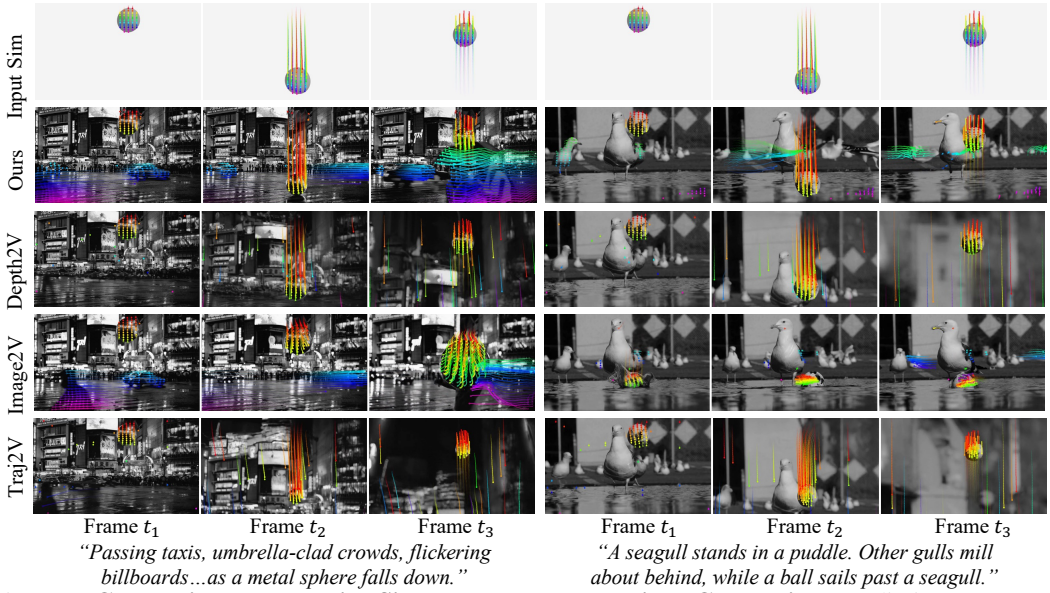


Figure 4: **Comparisons on Physics-Simulator-Instructed Video Generation.** Predictions are processed in grayscale and overlaid with estimated tracking (Xiao et al., 2024) for visualization.

Flow-Based Experts. We evaluate our method and several monolithic models, including image-to-video (Image2V), depth-to-video (Depth2V), and trajectory-to-video (Traj2V), all using the same Wan2.1 (14B) model as backbone, on a dataset consisting of 3 simulation inputs paired with 4 input images each, giving a total of 12 scenes with examples in Fig. 3. To obtain the initial frame for all methods, we edit the input image with our method described in Section 4.1 to insert the simulated object. Depth maps required as method inputs are rendered from the physical simulator, and input trajectory maps are computed using object centroids. Metrics include the ones from Section 4.1 that are relevant to this task, with additional ones adopted from a video benchmark VBench (Huang et al., 2024): motion smoothness scores introduced in VBench, MUSIQ (Ke et al., 2021) for imaging quality, and ViCLIP (Wang et al., 2023) for text prompts alignment. We further compute foreground motion trajectory accuracy with IoU on SAM2 (Ravi et al., 2024) detected from output videos.

Results are in Table 2 (top half). Depth-to-video and trajectory-to-video alone fail to capture rich non-foreground-object motions, such as cars moving and birds running in Fig. 4. In particular, they tend to compensate for the object's downward motion with upward camera motions, as visualized by the tracking trajectories in Fig. 4. Image-to-video model does not follow the input motion.

Autoregressive Experts. We implement the autoregressive construction for Eq. (3) using two generative experts: a next-frame-section video prediction model, FramePack (Zhang & Agrawala, 2025), and a Gaussian distribution $p^{\text{sim}}(x)$ centered on physics simulator renderings. Denote a video sequence of the physics simulator's RGB rendering overlayed with the input initial frame as c^{sim}, then $p^{\text{sim}}(x) :\propto \exp\left(-w\|x - c^{\text{sim}}\|_2^2\right)$ with constant $w \in \mathbb{R}$. In this case, $x_{1:t}$ represents the first $T+1-t$ frame sections of a video x. We approximate sampling from this distribution with its MAP solution, which amounts to gradient updates w.r.t. loss $\|x-c^{\text{sim}}\|_2^2$. Qualitative results are in Fig. 5, suggesting that composing per-expert constraints enforces desired object motion on top of output photorealism.

Full-Scene Simulation. In the above setting, simulation inputs only describe foreground objects. We further evaluate cases where full-scene simulation is available. We use the simulation results from 9 released scenes from PhysGen3D (Chen et al., 2025), a method that reconstructs and animates images in physical simulators, as evaluation inputs. To better preserve scene content, we use the flow-inverted noise vector for particle initialization as opposed to random noise, and compare with this inversion baseline. Results are included in Table 2 with qualitative samples in Fig. 11.

4.3 Text-to-Image Generation with Layout Control

Task and Evaluation. This task aims to generate an image given an input global text prompt and a set of object bounding boxes with paired object text descriptions. We evaluate on 50 scenes ran-

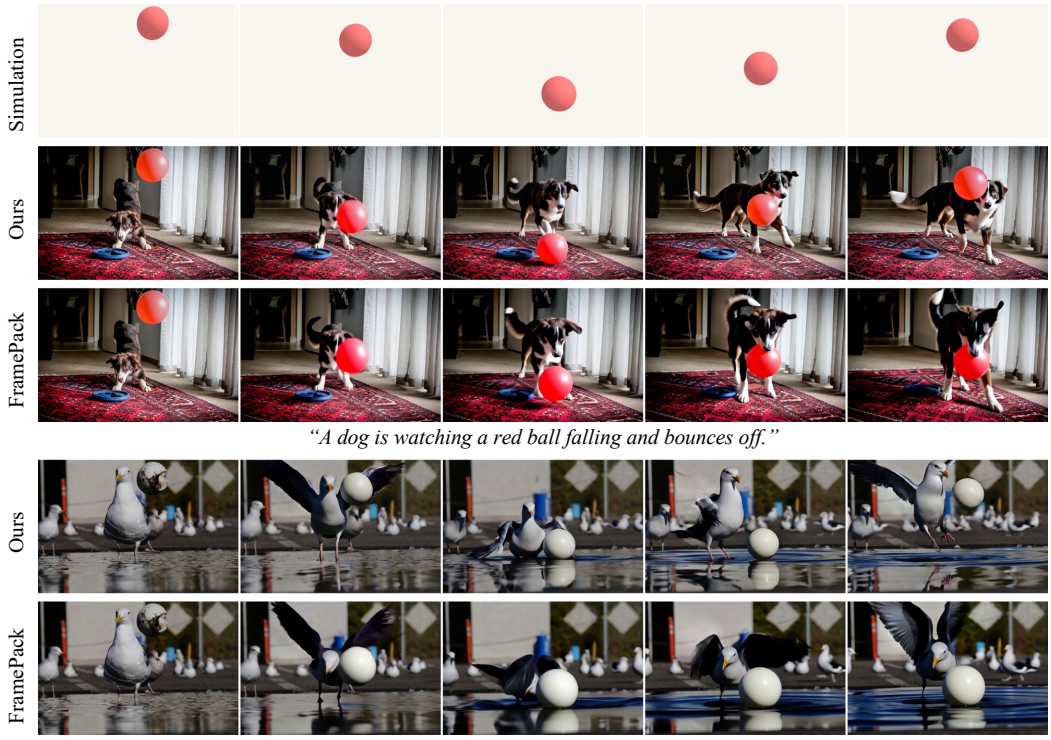


Figure 5: **Physical-Simulation-Instructed Video Generation** compared with the baseline with the same backbone. Our method better adheres to the input object motion trajectory.

Method	mIoU (↑)	VQAScore (↑)	VQAScore-R (↑)
3DIS-FLUX	0.673±0.182	0.800±0.190	0.808±0.161
$L = 1$ (Du et al. (2023))	0.675±0.200	0.567±0.272	0.567±0.189
$L = 4$	0.695±0.172	0.802±0.190	0.795±0.155
$L = 8$	0.715±0.177	0.879±0.115	0.843±0.135
$L = 16$	<u>0.723</u>±0.176	<u>0.897</u>±0.106	<u>0.870</u>±0.128
$L = 32$	**0.728**±0.170	**0.904**±0.092	**0.881**±0.115

Table 3: **Text-to-Image Generation with Layout Controls** on MIG-Bench (Zhou et al., 2024) dataset, comparing our method with varying compute budgets and an application-specific baseline.

domly sampled from the full MIG-Bench (Zhou et al., 2024) dataset of 800 scenes. Metrics include bounding box mIoU, detected using GroundingDINO (Liu et al., 2024b), and VQAScore (Lin et al., 2024) measuring output alignment with global and regional prompts.

Results. Table 3 and Fig. 8 contains results on ablating the number of particles L and also comparisons with 3DIS-FLUX (Zhou et al., 2025), a state-of-the-art method designed for text-to-image generation with the same backbone model as ours. In the case of $L = 1$, the method reduces to Du et al. (2023). The performance of our method improves with higher computation budgets, and alleviates the need for application-specific designs, such as cross-attention layer intervention as used in Zhou et al. (2025). The ablation on SMC resampling is deferred to Section D.5.

5 Conclusion

We have introduced a probabilistic framework where knowledge from heterogeneous sources is composed in the form of product distributions, and proposed an efficient sampling algorithm interleaving annealed MCMC sampling and SMC resampling to integrate guidance from multiple generative and discriminative models. Our training-free approach can effectively integrate visual generative priors, VLM guidance, and physics-based constraints. Empirical evaluation suggests that this method generates images and videos with improved controllability and fidelity compared to prior works, providing a practical recipe for controllable visual synthesis tasks.

Reproducibility Statement. Experiment details including implementation details for the proposed method and for baselines, evaluation protocol, and raw results are included in the appendix and the supplementary webpage. To promote reproducibility and to facilitate future research, we will release the code upon acceptance.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

Michael Bacharach. Scientific disagreement. Unpublished manuscript, 1972. 1

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 1

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *CVPR*, pp. 843–852, 2023. 2

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. *arXiv preprint arXiv:2302.08113*, 2023. 2

Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient nd image segmentation. *International journal of computer vision*, 70(2):109–131, 2006. 1

Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 1, pp. 105–112. IEEE, 2001. 1

Arwen Bradley, Preetum Nakkiran, David Berthelot, James Thornton, and Joshua M Susskind. Mechanisms of projective composition of diffusion models. *arXiv preprint arXiv:2502.04549*, 2025. 2

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024. URL https://openai.com/research/video-generation-models-as-world-simulators. 1

Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, et al. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. *arXiv preprint arXiv:2501.08331*, 2025. 2

Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. Physgen3d: Crafting a miniature interactive world from a single image. *CVPR*, 2025. 2, 7, 8, 20

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022. 5

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021. 2, 21

Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo methods. *Sequential Monte Carlo methods in practice*, pp. 3–14, 2001a. 1

Arnaud Doucet, Nando De Freitas, Neil James Gordon, et al. *Sequential Monte Carlo methods in practice*. Springer, 2001b. 4

Yilun Du and Leslie Kaelbling. Compositional generative modeling: A single model is not all you need. *arXiv preprint arXiv:2402.01103*, 2024. 1, 2, 3

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020. 1, 2, 3

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *ICLR*, pp. 8489–8510. PMLR, 2023. 1, 2, 9

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. 5

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 16

Leonhard Euler. *Institutiones calculi integralis*, volume 1. impensis Academiae imperialis scientiarum, 1792. 16

FLUX. Flux. https://github.com/black-forest-labs/flux, 2024. 6

Timur Garipov, Sebastiaan De Peuter, Ge Yang, Vikas Garg, Samuel Kaski, and Tommi Jaakkola. Compositional sculpting of iterative generative processes. *arXiv preprint arXiv:2309.16115*, 2023. 2

Sachit Gaudi, Gautam Sreekumar, and Vishnu Boddeti. Coind: Enabling logical compositions in diffusion models. *arXiv preprint arXiv:2503.01145*, 2025. 2

Tomas Geffner, George Papamakarios, and Andriy Mnih. Compositional score modeling for simulation-based inference. In *International Conference on Machine Learning*, pp. 11098–11116. PMLR, 2023. 1

Christian Genest and James V Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135, 1986. 1

Ben Glocker, Nikos Komodakis, Georgios Tziritas, Nassir Navab, and Nikos Paragios. Dense image registration through mrfs and efficient linear programming. *Medical image analysis*, 12(6):731–741, 2008. 1

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019. 2

Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, Wenping Wang, and Yuan Liu. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. *arXiv preprint arXiv:2501.03847*, 2025. 2

Jiajun He, José Miguel Hernández-Lobato, Yuanqi Du, and Francisco Vargas. Rne: plug-and-play diffusion inference-time control and energy-based training, 2025. URL https://arxiv.org/abs/2506.05668. 2

Xuming He, Richard S Zemel, and Miguel A Carreira-Perpinán. Multiscale conditional random fields for image labeling. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pp. II–II. IEEE, 2004. 1

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, et al. Manifold preserving guided diffusion. *arXiv preprint arXiv:2311.16424*, 2023. 2

Geoffrey E Hinton. Products of experts. *International Conference on Artificial Neural Networks*, 1999. 1

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002. 3

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 21

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3, 16

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 1

Xun Huang, Arun Mallya, Ting-Chun Wang, and Ming-Yu Liu. Multimodal conditional image synthesis with product-of-experts gans. In *ECCV*, pp. 91–109. Springer, 2022. 1, 2

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, pp. 21807–21818, 2024. 8

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 20

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *ICCV*, pp. 5148–5157, 2021. 8

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 18

Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *European conference on computer vision*, pp. 82–96. Springer, 2002. 1

Sanjiv Kumar and Martial Hebert. Discriminative fields for modeling spatial dependencies in natural images. *Advances in neural information processing systems*, 16, 2003. 1

Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 29–37. JMLR Workshop and Conference Proceedings, 2011. 3

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023. 1

Shuang Li, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, and Igor Mordatch. Composing ensembles of pre-trained models via iterative consensus. *arXiv preprint arXiv:2210.11522*, 2022. 2

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pp. 366–384. Springer, 2024. 9, 19

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 3

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024. 16

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *ECCV*, pp. 423–439. Springer, 2022a. 2

Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision*, pp. 360–378. Springer, 2024a. 2

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024b. 9

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022b. 3

Grace Luo, Jonathan Granskog, Aleksander Holynski, and Trevor Darrell. Dual-process image generation. *arXiv preprint arXiv:2506.01955*, 2025. 2

Divyat Mahajan, Mohammad Pezeshki, Ioannis Mitliagkas, Kartik Ahuja, and Pascal Vincent. Compositional risk minimization. *arXiv preprint arXiv:2410.06303*, 2024. 2

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 7

Antonio Montanaro, Luca Savant Aira, Emanuele Aiello, Diego Valsesia, and Enrico Magli. Motioncraft: Physics-based zero-shot video generation. *Advances in Neural Information Processing Systems*, 37:123155–123181, 2024. 2

Radford M Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001. 1, 3

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 3

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 7

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021. 1

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714. 8, 20

Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999. 3

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*, pp. 10684–10695, 2022. 1

Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. In *ICLR*, 2025. 2

Nataniel Ruiz, Yuanzhen Li, Neal Wadhwa, Yael Pritch, Michael Rubinstein, David E Jacobs, and Shlomi Fruchter. Magic insert: Style-aware drag-and-drop. *arXiv preprint arXiv:2407.02489*, 2024. 6

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

Christoph Schuhmann and Romain Beaumont. Laion-aesthetics. *LAION. AI*, 2022. 6

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025. 2

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. *arXiv preprint arXiv:2503.02819*, 2025. 2

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015. 3

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pp. 32483–32498. PMLR, 2023. 2

Jocelin Su, Nan Liu, Yanbo Wang, Joshua B Tenenbaum, and Yilun Du. Compositional image decomposition with diffusion models. *arXiv preprint arXiv:2406.19298*, 2024. 2

Xiyang Tan, Ying Jiang, Xuan Li, Zeshun Zong, Tianyi Xie, Yin Yang, and Chenfanfu Jiang. Physmotion: Physics-grounded dynamics from a single image. *arXiv preprint arXiv:2411.17189*, 2024. 2

Yoad Tewel, Rinon Gal, Dvir Samuel, Yuval Atzmon, Lior Wolf, and Gal Chechik. Add-it: Training-free object insertion in images with pretrained diffusion models. *arXiv preprint arXiv:2411.07232*, 2024. 7

James Thornton, Louis Bethune, Ruixiang Zhang, Arwen Bradley, Preetum Nakkiran, and Shuangfei Zhai. Composition and control with distilled energy diffusion models and sequential monte carlo. *arXiv preprint arXiv:2502.12786*, 2025. 2

Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 7

Chaohui Wang, Nikos Komodakis, and Nikos Paragios. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117(11):1610–1627, 2013. 1

Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming rectified flow for inversion and editing. *arXiv preprint arXiv:2411.04746*, 2024. 6

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023. 8

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 681–688, 2011. 3

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *NeurIPS*, 36:31372–31403, 2023. 2

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *CVPR*, 2024. 8

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *NeurIPS*, pp. 15903–15935, 2023. 6

Sherry Yang, Yilun Du, Bo Dai, Dale Schuurmans, Joshua B. Tenenbaum, and Pieter Abbeel. Probabilistic adaptation of black-box text-to-video models. In *ICLR*, 2024. 2

Xindi Yang, Baolu Li, Yiming Zhang, Zhenfei Yin, Lei Bai, Liqian Ma, Zhiyong Wang, Jianfei Cai, Tien-Tsin Wong, Huchuan Lu, et al. Towards physically plausible video generation via vlm planning. *arXiv preprint arXiv:2503.23368*, 2025. 2

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Y Zou, and Stefano Ermon. Tfg: Unified training-free guidance for diffusion models. *NeurIPS*, 37: 22370–22417, 2024. 2

Lvmin Zhang and Maneesh Agrawala. Packing input frame contexts in next-frame prediction models for video generation. *Arxiv*, 2025. 7, 8

Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming-Yu Liu. Diffcollage: Parallel generation of large content with diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10188–10198. IEEE, 2023. 2

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic inference in language models via twisted sequential monte carlo. *arXiv preprint arXiv:2404.17546*, 2024. 2

Dewei Zhou, You Li, Fan Ma, Xiaoting Zhang, and Yi Yang. Migc: Multi-instance generation controller for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6818–6828, 2024. 9

Dewei Zhou, Ji Xie, Zongxin Yang, and Yi Yang. 3dis-flux: simple and efficient multi-instance generation with dit rendering. *arXiv preprint arXiv:2501.05131*, 2025. 9, 21

A Overview

The supplementary material contains discussions (Section B), extended descriptions of the framework (Section C), and additional experimental results (Section D) and details (Section E).

B Discussions

Compute Requirement. All image experiments are run using 1 NVIDIA H200 GPUs and all video experiments with 2 NVIDIA H200 GPUs. Generating one sample in Section 4.1 takes approximately 4min. It takes 5min for flow-based and 30min for autoregressive-based models in Section 4.2. Wall-clock time for the task in Section 4.3 with different numbers of particles is included in Table 5.

Limitations. The proposed method requires extra computation due to intermediate MCMC steps and parallel sampling compared to vanilla feedforward inference of the image or video generative model backbones. Furthermore, the sampling efficacy relies on the assumption that expert predictions are reasonably compatible, and strong incompatibilities might require other sampling techniques or training. Qualitative examples are discussed in Section D.4.

This work proposes a framework and provides empirical validation with small number of experts and we leave large-scale sampling as future work. We mainly highlight two straightforward sources for parallelism: MCMC sampling across particles, and inference for different generative experts within each MCMC step evaluation.

Societal Impact. We believe outputs of the proposed framework does not directly possess negative societal impact. The underlying technology enable more content creation tools and benefits understanding model behaviors. However, highly controllable generation could be exploited to create misinformation. While our work does not inherently use sensitive data, we are aware of such risks and closely follow ethical guidelines in the community to help mitigate these risks.

C Framework Details

C.1 Flow Models

We use bar notations for pre-trained models in the following discussions.

Consider the i-th generative expert, which is a flow model that defines a time-dependent continuous probability path $\bar{p}^{(i)}(x;\bar{t}), x \in \mathbb{R}^d, \bar{t} \in [0,1]$, with boundary conditions $\bar{p}^{(i)}(x;0) = \mathcal{N}(0,I)$ and $\bar{p}^{(i)}(x;1) \approx p^{(i)}_{\text{data}}(x)$. Each flow model is associated with a scheduler $\bar{\alpha}, \bar{\sigma} : [0,1] \to \mathbb{R}$ that defines a conditional probability path $\bar{p}^{(i)}(x \mid y;\bar{t}) = \mathcal{N}(x \mid \bar{\alpha}(\bar{t})y, \bar{\sigma}(\bar{t})I), y \sim p^{(i)}_{\text{data}}(y)$, and is trained to predict a velocity field $\bar{v}^{(i)}(x;\bar{t},\theta_i)$ that generates the marginalized path $\bar{p}^{(i)}(x;\bar{t})$, where θ_i denotes model parameters. While the training-time scheduler $(\bar{\alpha},\bar{\sigma})$ can differ across experts, they can be aligned during inference time via a scale-time transformation (Lipman et al., 2024).

Given a time remapping function $\xi : \{T, T-1, \cdots, 1\} \to [0,1]$, one can construct the per-expert product component from Eq. (3) via discretizing the path $\bar{p}^{(i)}(x;\bar{t})$ into

$$p^{(i)}_t(x) := \bar{p}^{(i)}(x;\xi(t)). \tag{7}$$

We require that ξ is monotonic and $\xi(T) = 0, \xi(1) = 1$. By construction, $p^{(i)}_T(x) = \mathcal{N}(0,I)$ is easy to sample from, and $p^{(i)}_1(x) = p^{(i)}_{\text{data}}(x)$ encapsulates the prior data distribution of the expert model, fulling our motivation in Section 3.1. For all experiments, we define ξ to be the Euler discretization (Esser et al., 2024; Euler, 1792), with additional alignment to the DDPM scheduler (Ho et al., 2020) for experiments in Section 4.3.

Line 5 from Algorithm 1 is computed as follows:

$$
\begin{aligned}
v_t^{(i)}(x) &= \bar{v}^{(i)}(x; \xi(t), \theta_i), \\
\alpha_t &= \bar{\alpha}(\xi(t)), \\
\sigma_t &= \bar{\sigma}(\xi(t)), \\
s_t^{(i)}(x) &= \frac{-\alpha_t v_t^{(i)}(x) + \dot{\alpha}_t x}{\dot{\sigma}_t \sigma_t \alpha_t - \dot{\alpha}_t \sigma_t^2}, \\
v_t(x) &= \sum_i \lambda^{(i)}(x) v^{(i)}(x), \\
s_t(x) &= \sum_i \lambda^{(i)}(x) s^{(i)}(x), \\
\mathcal{K}_{t \leftarrow t+1}(x' \mid x) &= \delta(x' - (x + v_t(x)(\xi(t-1) - \xi(t)))), \\
\mathcal{K}_t(x' \mid x) &= \mathcal{N}(x'; x + \frac{\kappa^2}{2} s_t(x), \kappa^2 I),
\end{aligned}
$$

where $\lambda^{(i)}(x)$ is defined in Section E.1, with all entries of $\sum_i \lambda^{(i)}(x)$ to be 1.

The end-point prediction is available as

$$
\hat{x} = \frac{\sigma_t}{\dot{\alpha}_t \sigma_t - \dot{\sigma}_t \alpha_t} v_t(x) - \frac{\dot{\sigma}_t}{\dot{\alpha}_t \sigma_t - \dot{\sigma}_t \alpha_t} x. \tag{8}
$$

C.2 Approximation for Conditional Sampling

We explain Eq. (6) below. Let x_i be any expert. We approximate the distribution $p(x_i \mid x_{\text{pa}(i)})$ as

$$
p(x_i \mid x_{\text{pa}(i)}) \propto p(x_i) p(x_{\text{pa}(i)} \mid x_i) = p(x_i) \prod_{i' \in \text{pa}(i)} p(x_{i'} \mid x_i), \tag{9}
$$

where parent regions $x_{i'}$ are assumed to be conditionally independent given x_i. To model each conditional distribution $p(x_{i'} \mid x_i)$, we define a Gaussian distribution of the deviation of the flow vector predicted at $x_{i'}$ and x_i, where $p(x_{i'} \mid x_i) \propto e^{-w \| v(x_{i'}) - v(x_i) \|^2}$ with a constant $w \in \mathbb{R}$. Under this parametric distribution, when $x_{i'}$ and x_i have consistent flow predictions, the likelihood is high (which is reasonable as this indicates both $x_{i'}$ and x_i are mutually compatible). We can convert the probability expression in Eq. (9) to a corresponding score function, resulting in Eq. (6).

In this work, $\text{pa}(i)$ is defined as the generative expert defining a natural data distribution for global $x \in \mathbb{R}^d$, e.g., a text-conditioned generative model for the full image/video x. Each generative expert is additionally conditioned on a context signal $c_i \in \mathbb{R}^{d_i^{\text{context}}}$, e.g., texts or depth maps. Note that the framework allows for different types of conditions across experts, enabling flexible control handles that are typically application-dependent.

D Extended Experimental Results

D.1 Synthetic Experiments

We conducted experiments using a synthetic mixture-of-Gaussians dataset with known ground-truth product distributions for further quantitative evaluation.

Dataset. We use a 2D mixture-of-Gaussian as the target distribution for two experts $p_{\text{data}}^{(1)}$ and $p_{\text{data}}^{(2)}$. The first expert mixture consists of 8 Gaussian whose means are equally spaced along the x-axis with anisotropic covariance; the second is defined analogously along the y-axis. Fig. 6 shows the visualization of ground truth distributions.

Metrics. W1 and W2 are empirical 1- and 2-Wasserstein distances between method samples and ground truth ones. MMD is the squared maximum mean discrepancy with RBF kernels. TV is total variation distance. NLL is the Monte Carlo estimate of the negative log-likelihood $-\mathbb{E}\left[\log p^{\text{data}}(x)\right]$ of method samples under the analytic density $p^{\text{data}}(x)$.

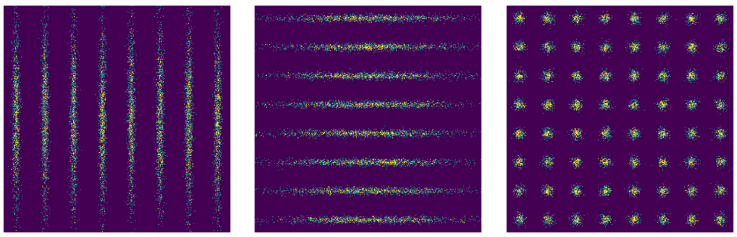


Figure 6: **Mixture-of-Gaussian Data Visualization.**

Method	W1 (↓)	W2 (↓)	MMD (↓)	TV (↓)	NLL (↓)
Expert 1 ($p^{(1)}$)	0.906	1.033	0.047	0.709	7.727
Expert 2 ($p^{(2)}$)	0.883	1.023	0.045	0.702	8.197
No-annealing ($T = 1$)	3.765	4.005	5.075	0.999	21.129
No-MCMC ($K = 0$)	0.728	0.909	0.027	0.604	5.008
Ours	**0.580**	**0.777**	**0.018**	**0.583**	**4.517**

Table 4: **Comparisons on Mixture-of-Gaussian Data.**

Models. Each expert model is a 5-layer MLP with hidden dimension 512. Models are trained with flow matching for 10000 iterations with Adam (Kingma, 2014) optimizer and learning rate $1e-3$. During inference, for each method, we choose the number of discretization steps T and the number of Langevin steps K so that the number of function evaluation (NFE) is constant across models. NFE is computed with $(K+1)(T-1)$ since each step involves 1 query for initialization and K for MCMC updates, and there are $T-1$ steps in the for loop in Algorithm 1.

Results. Results are reported in Table 4. Our method achieves better sampling quality compared to all baselines, indicating improved mixing on this dataset, with sharply peaked energy surface of the target product distribution.

D.2 Qualitative Video Results

Please view the "index.html" in the supplementary material folder using a web browser.

D.3 Graphics-Engine Instructed Image Editing

Fig. 7 contains qualitative samples on the Magic Insert dataset for experiments in Section 4.1.

D.4 Text-to-Image Generation with Regional Composition

Fig. 8 contains qualitative samples on the MIG-Bench dataset for experiments in Section 4.3.

This task exemplifies challenges when multiple expert models (one for each image region) are composed, and we observe two failure modes expanding the discussions from Section B.

The first arises when input conditions are contradictory. An example is shown in the second-to-last row of Fig. 8, where the regions for "a yellow horse" and "a blue horse" overlap. In this case, the support of the product distribution becomes very small, and although our method draws samples from the product, the resulting samples may not be visually desirable.

The second occurs when some input conditions are complex and our method inherits limitations of the underlying expert models. For example, in the last row of Fig. 8, the text prompt for the global region specifies attributes for multiple objects, which leads to attribute leakage: the global expert makes all three hot dogs blue in the result for $L = 32$.

Method	mIoU (↑)	VQAScore (↑)	VQAScore-R (↑)	Wall-Clock Time (min)
$L = 1$	0.675±0.200	0.567±0.272	0.567±0.189	∼1
$L = 1$ (No SMC)	0.675±0.200 (+0.000)	0.567±0.272 (+0.000)	0.567±0.189 (+0.000)	∼1
$L = 4$	0.695±0.172	0.802±0.190	0.795±0.155	∼4
$L = 4$ (No SMC)	0.715±0.170 (+0.020)	0.795±0.170 (−0.007)	0.767±0.159 (−0.028)	∼4
$L = 8$	0.715±0.177	0.879±0.115	0.843±0.135	∼6
$L = 8$ (No SMC)	0.735±0.163 (+0.020)	0.807±0.182 (−0.072)	0.795±0.153 (−0.048)	∼6
$L = 16$	0.723±0.176	0.897±0.106	0.870±0.128	∼12
$L = 16$ (No SMC)	0.698±0.177 (−0.025)	0.814±0.186 (−0.083)	0.806±0.162 (−0.064)	∼12
$L = 32$	**0.728**±0.170	**0.904**±0.092	**0.881**±0.115	∼25
$L = 32$ (No SMC)	0.691±0.183 (−0.037)	0.854±0.139 (−0.050)	0.838±0.136 (−0.043)	∼25

Table 5: **SMC resampling ablations** on the task from Section 4.3. Values in green (red) indicate "No SMC" is better (worse). Results suggest that SMC benefits sampling efficiency. Resampling has relatively marginal computation.

D.5 Ablation on SMC

We conduct ablation for SMC with results shown in Table 5.

D.6 Text-to-Image Generation

We further demonstrate the efficacy of distribution composition on the task of text-to-image generation, where we first query an LLM to parse an input text prompt into spatial bounding boxes with regional prompts (Fig. 9 left), aiming to compose the distributions from text-to-image model (FLUX.1 [dev]) conditioned on regional text prompts, one for each region, as generative experts, and from VQAScore (Lin et al., 2024) as a discriminative expert, which gives a scalar value predicting the probability that its input image is aligned with the input text prompt. We compare our methods using $L = 16, 4$ particles and 1 particle (which amounts to not using the discriminative expert), a variant using no generative composition (equivalent to best-of-N sampling for vanilla FLUX) while matching the compute budget for 16 particles, and with the vanilla FLUX backbone. In the examples shown in Fig. 9, our compositional approach achieves better text alignment compared to the baselines, and the discriminative expert provides crucial knowledge, e.g., counting, to steer the output distribution.

The parallel sampling process is visualized in Fig. 10. For each scene from Section 4.3, we use an LLM to convert a text prompt into a layout of bounding boxes following the procedure described in Section E.5. The input text prompts and resulting layouts and regional prompts are visualized in Fig. 9. Then, we follow Algorithm 1 and initialize $L = 4$ particles and annealed schedule length $T = 28$, which are reweighed at three key timesteps $t = 21, 19, 1$ using the discriminative expert, VQAScore (Lin et al., 2024), which measures image-text alignment with a probability score. At $t = 21, 19$, we binarize the scores with the threshold set to be the median score of the current 4 particles, discard the two below the threshold, and duplicate the remaining two high-score particles two times to maintain 4 particles in each round. At $t = 1$, we keep the highest-scoring particle as the final output sample.

Fig. 10 shows end-point predictions to visualize the intermediate samples from the annealing process from $t = 28$ to $t = 1$. VQAScore measurements (higher means more aligned) are overlayed on the top-left corner for each sample. In the proposed framework, integrating multiple generative experts allows for composing simpler regional prompts, construction of the annealed distributions allows for iterative refinement of samples, and parallel sampling further integrates the knowledge from the discriminative expert and improves output accuracy, as evidenced in the comparisons from Fig. 9.

D.7 Physical-Simulator-Instructed Video Generation

Qualitative samples are included in Fig. 11.

E EXPERIMENT DETAILS

E.1 IMPLEMENTATION DETAILS

As described in Eq. (4), a generative expert can be defined over the entire scene $x \in \mathbb{R}^d$ or a region $x_i \in \mathbb{R}^{d^{(i)}}$ with $d^{(i)} \leq d$, in which case the expert has prior over only part of the scene, e.g., a spatial region in Section 4.3. In these cases, we zero-pad the model prediction from $\mathbb{R}^{d^{(i)}}$ to $\mathbb{R}^d$.

In other cases, one may restrict the influence of a generative expert to update only a scene region, even when the expert is defined over the full scene $x \in \mathbb{R}^d$, e.g., when the expert only has knowledge over foreground object informed by the graphics engine or physics simulators (Sections 4.1 and 4.2). We apply a regional (spatial for images and spatio-temporal for videos) weight $\lambda^{(i)}(x) \in [0,1]^d$ on the velocity prediction $v_t^{(i)}(x)$ in these cases, where $\lambda^{(i)}(x)$ is a Gaussian-blurred foreground mask.

E.2 AUTOMATIC EVALUATION WITH VLMS

We follow the protocol proposed in PhysGen3D (Chen et al., 2025) to run automatic evaluation for Sections 4.2 and 4.3 using GPT-4o (Hurst et al., 2024). For each scene, all methods' output images, or uniformly sampled 10 frames when outputs are videos, are sent to GPT-4o together with template prompts as specified in PhysGen3D (Chen et al., 2025). GPT-4o is asked to give scores from 0 to 1 for each video on all metrics. The scores for the same metric are averaged across all scenes.

E.3 GRAPHICS-ENGINE-INSTRUCTED IMAGE EDITING

In Section 4.1, the product distribution is defined as $p(x) \propto p^{\text{depth2image}}(x \mid c^{\text{text}}, c^{\text{depth}}) p^{\text{imagefill}}(x \mid c^{\text{text}}, c^{\text{image}}, c^{\text{mask}})$. We use $T = 28, L = 1$, with 1 MCMC sampling step and 2 conditional sampling updates (Eq. (6)) per iteration. The spatial resolution is 960×1664 for the graphics engine dataset and 1024×1024 for Magic Insert dataset.

For the graphics engine dataset, assets are rendered in Genesis[1].

For Magic Insert data preprocessing, we use SAM2 (Ravi et al., 2024) to obtain the segmentation masks of foreground characters, and feed the images of segmented characters, resized to the bottom-left quadrant and overlaid with backgrounds, into GPT o4-mini for image captions to serve as text prompts for evaluated methods. Magic Insert results are downloaded from the official demo page. Add-it receives background images from the dataset as inputs, together with the GPT-generated captions (of overlaid images) appended with "{character} is at the bottom left quarter" to indicate the desired locations. SDEdit takes in input overlaid images and adds Gaussian noise with standard deviation 0.5 before denoising.

E.4 PHYSICAL-SIMULATOR-INSTRUCTED VIDEO GENERATION

In Section 4.2, for flow models, the product distribution is defined as $p(x) \propto p^{\text{depth2video}}(x \mid c^{\text{text}}, c^{\text{depth}}) p^{\text{image2video}}(x \mid c^{\text{text}}, c^{\text{image}})$. We use $T = 30, L = 1$ with 1 MCMC sampling step per iteration. We use no conditional sampling updates for this task due to GPU memory constraint. Each generated video has 81 frames and spatial resolution 480×832.

For autoregressive models, the product distribution is defined as $p(x) \propto p^{\text{sim}}(x \mid c^{\text{sim}}) p^{\text{image2video}}(x \mid c^{\text{text}}, c^{\text{image}})$ with $p^{\text{sim}}(x \mid c^{\text{sim}})$ defined in Section 4.2. We approximate sampling from this distribution with first sampling from $p^{\text{image2video}}$, then do 8 gradient updates with respect to $\mathcal{L}_2$ loss $\|x - c^{\text{simulation}}\|^2$. Results in Fig. 5 use 81 frames per sequence and spatial resolution 512×768. We set $L = 1$, with 1 MCMC sampling step per iteration for this experiment.

E.5 TEXT-TO-IMAGE GENERATION WITH REGIONAL CONTROL

For Section 4.3, the product distribution is $p(x) \propto \left(\prod_i p^{\text{depth2image}}(x_i \mid c^{\text{text}}, c^{\text{depth}})\right) q^{\text{VLM}}(x \mid c^{\text{text}})$. Here, $p^{\text{depth2image}}$ is a regional generative experts (FLUX-Depth), which predict scores for bounding-box-cropped images conditioning on regional text prompts and on regional depth maps cropped from

[1] https://github.com/Genesis-Embodied-AI/Genesis

global depth maps predicted using the first stage of 3DIS-FLUX (Zhou et al., 2025) (layout-to-depth generation); q^{VLM} a discriminative expert VQAScore, where for each image sample, it assigns the summed score computed over a full image and its regions cropped with input bounding boxes as the reward. We use $T = 28$ and image resolution 1024×1024, with 1 MCMC sampling step and 2 conditional sampling steps per iteration.

F Extended Discussions

F.1 Guidance for Diffusion Models

Classifier guidance (CG) (Dhariwal & Nichol, 2021) and classifier-free guidance (CFG) (Ho & Salimans, 2022) are canonical approaches for improving sampling quality of diffusion models. While our framework is a general sampler for product distributions and can be instantiated with diffusion-based and autoregressive models, we proceed with the discussion fully within the context of diffusion models, and expand on the connection and differences of this work with CG and CFG below.

CG aims to sample from a class-conditioned distribution by adding the score of a base generative model and a classifier trained on noisy inputs. However, because the product of diffused marginals does not equal the diffused marginal of the product distribution, CG results in biased scores. We formalize this claim in the follows.

Let P_t be an operator on a density function $f(x), x \in \mathbb{R}^d$, with

$$P_t f(x) := \int \mathcal{N}(x; \alpha_t x_0, \sigma_t^2 I) f(x_0)\, \mathrm{d}x_0, \tag{10}$$

where $\alpha_t, \sigma_t \in \mathbb{R}$ are constants, then $P_t f$ is the diffused marginal of f corresponding to a noise level t as defined in standard diffusion or flow-based models. In the case of CG for a fixed class c, the desired target distribution is $p^{\text{CG}}(x) := p(x)p^s(c \mid x; t)$ with unconditional distribution $p(x)$, classifier $p(c \mid x; t)$, and gradient scale $s \in \mathbb{R}$. CG aims to sample from p^{CG} following a Markov chain $(P_t p^{\text{CG}})_{t \geq 0}$. Doing so requires scores for $P_t p^{\text{CG}}$. However, in general cases,

$$(P_t p^{\text{CG}})(x) \neq (P_t p)(x)\, p(c \mid x; t)^s, \tag{11}$$

and therefore the score of the right-hand side of Eq. (11) as computed in CG is biased for the true marginal $P_t(p^{\text{CG}})$ which corresponds to the left-hand side.

Similarly, in CFG, for a fixed class c and guidance scale s, $p^{\text{CFG}}(x) := p^s(x \mid c)p^{1-s}(x \mid \emptyset)$. For general densities u, v, $P_t(uv) \neq P_t u P_t v$. In particular,

$$P_t p^{\text{CFG}} = P_t \left[p(\cdot \mid c)^s p(\cdot \mid \emptyset)^{1-s}\right] \neq (P_t p(\cdot \mid c))^s (P_t p(\cdot \mid \emptyset))^{1-s}, \tag{12}$$

also leading to biased scores estimates.

On the other hand, when applied to sampling from p^{CG}, our proposed sampler explicitly sets the intermediate annealing target at step t to the right-hand side of Eqs. (11) and (12) and uses additional MCMC transitions to asymptotically converge to the target, yielding unbiased samples.

F.2 Evaluation on Class-Conditioned Sampling

In the case of class-conditioned sampling for p^{CG} and p^{CFG}, vanilla CG can be viewed as a special case with $K = 0$ MCMC steps with K defined in Algorithm 1. The advantages of setting $K > 0$ are empirically verified in the following experiments.

Baselines. We compare the following methods (s is guidance scale as defined in Section F.1, and K is the number of MCMC steps): vanilla class-conditional sampling without guidance with $s = 1, K = 0$; CFG with guidance scale with $s = 2, K = 0$; and our sampler with additional MCMC refinement with $s = 1, K > 0$. All models are 5-layer MLPs with hidden dimension 512, trained with flow matching. We zero out class conditioning with probability 0.5 during training. Similar to Section D.1, the step number T for each method is determined to maintain constant NFEs; each step for CFG involves 2 NFEs while others involve 1.

Dataset and Metrics. We use a synthetic 2D dataset forming an 8×8 checkerboard, where each

Method	W1 (↓)	W2 (↓)	MMD (↓)	TV (↓)	NLL (↓)
Vanilla ($s = 1, K = 0$)	0.581	0.724	0.389	0.833	2.796
CFG ($s = 2, K = 0$)	**0.296**	0.476	0.026	0.671	2.631
Ours ($s = 1, K = 1$)	**0.296**	**0.468**	**0.024**	**0.667**	**2.534**

Table 6: **Evaluation on Class-Conditioned Sampling.**

column corresponds to one class. Ground truth data visualization is shown in Fig. 12. Metrics follow Section D.1.

Results. Results are reported in Table 6. Applying additional MCMC steps to the class-conditioned sampling tasks, with total NFE controlled, results in better sampling quality compared to diffusion model inference with and without CFG.

Figure 7: **Comparisons on Character Insertion Task.**

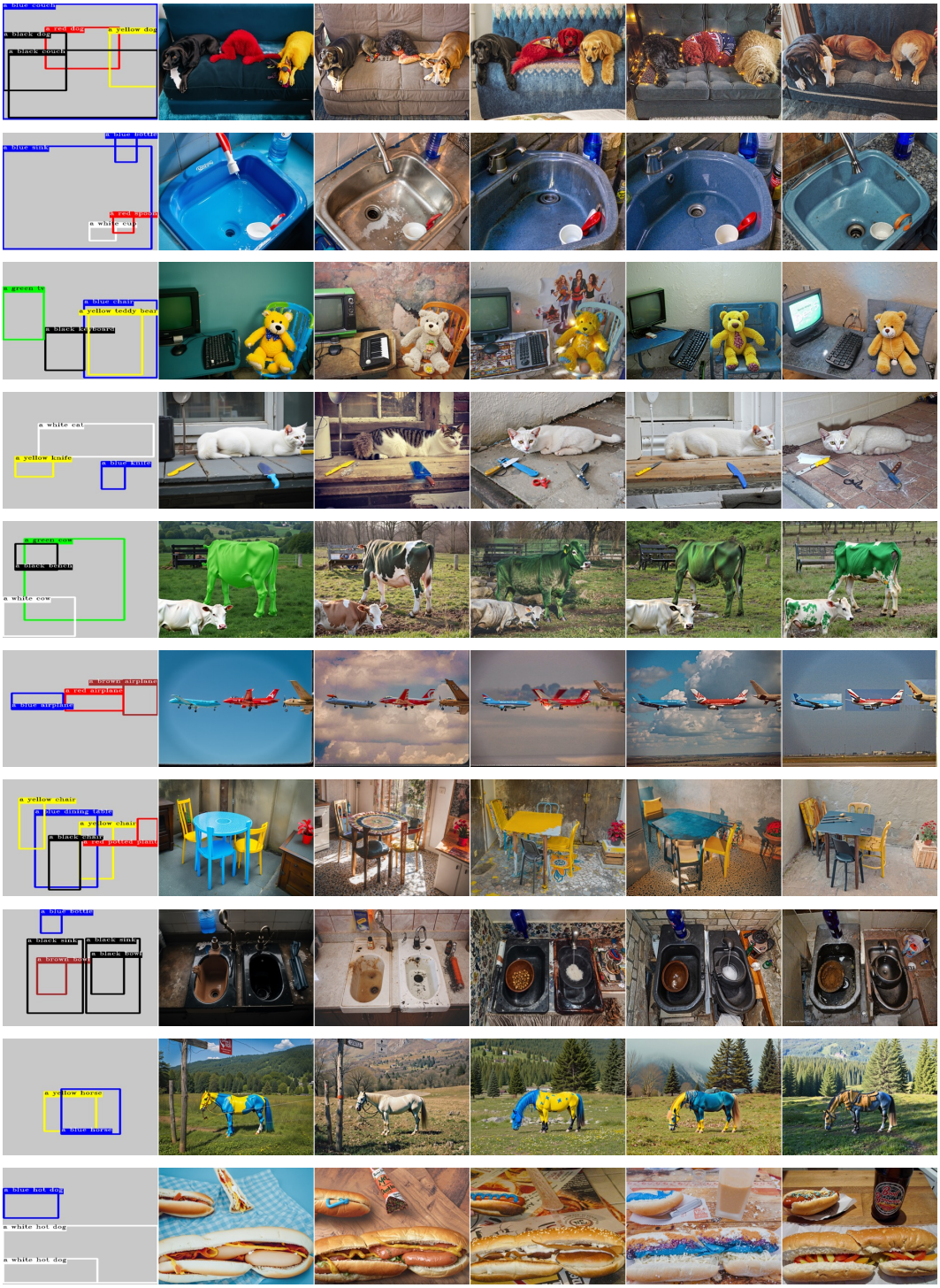
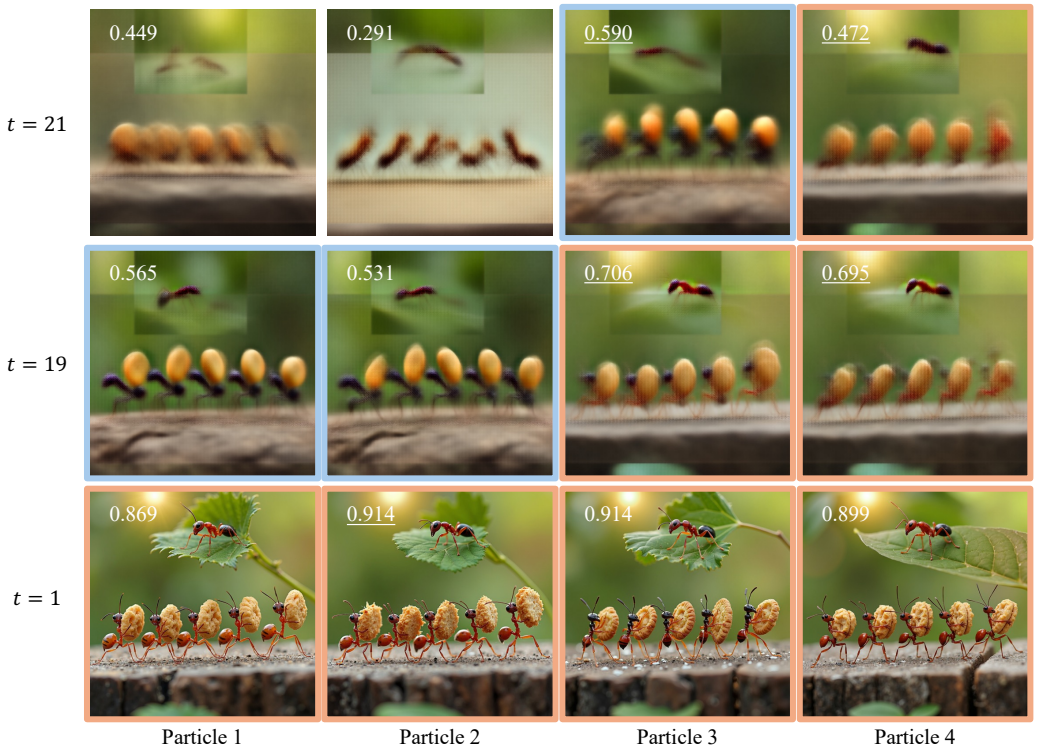


Figure 8: **Comparisons on Layout-Conditioned Image Generation Task.**

Figure 9: **Text-to-Image Generation**.



"Five ants are carrying biscuits, and an ant that is not carrying biscuits is standing on a green leaf directing them."

"In a field of flowers, every bloom is yellow, save for one that is blue."

Figure 10: **Visualizations of Parallel Sampling with Discriminative Experts.** Discriminative expert (VQAScore) scores are annotated on the top-left corners; score underlining means the sample proceeds to the next annealed distributions (with smaller t). Images with the same boundary color share the same initial seed.

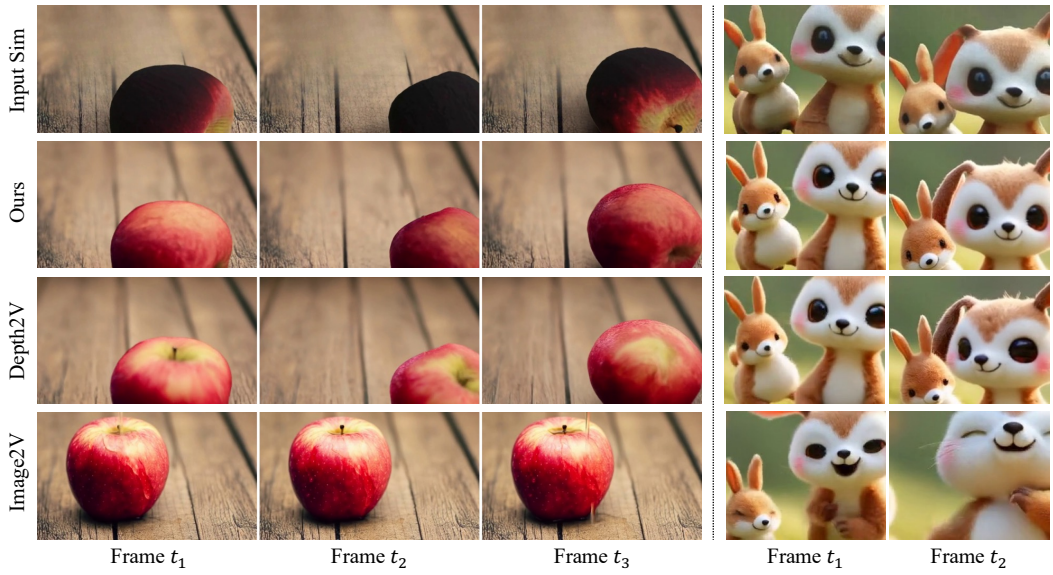


Figure 11: **Baseline Comparisons on PhysGen3D Data.** Our method improves the visual quality compared to the input simulation and adheres more closely to the input object motions compared to baselines.

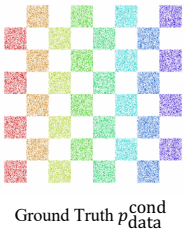


Figure 12: **Class-Conditioned Data Visualization.**