Utilizing noise space manipulation for sampling, composable diffusion and video diffusion

Vishesh Gupta, Jente Vandersanden, Gurprit Singh

Max Planck Institute for Informatiks

Abstract. We explore noise-space manipulation in generative modelling, venturing into multiple use cases - compositional generation, accelerated sampling, and noise guidance. We then use it as a light weight alternative for video diffusion, enabling temporally coherent generation via frozen image models.

1 Introduction

Diffusion models [1] have established themselves as the leading generative paradigm in a broad spectrum of domains, including image synthesis, speech generation, molecular modeling, video generation, and 3D content creation. This is due to their remarkable ability to faithfully approximate and sample diverse, multi-modal distributions with stable, high-fidelity outputs [2]. By enabling reliable, scalable generation across domains, diffusion models have become central to generative modeling innovation in both academic research and industry deployment.

However, training diffusion models at the scale required for foundational generative systems is computationally prohibitive, often demanding weeks—or even months—of distributed GPU computation over massive multimodal datasets. To enable reuse of such pre-trained models for downstream tasks, recent work by Liu et al. [3] and Du et al. [4] propose efficient compositional generation frameworks that leverage independently trained diffusion models without requiring joint retraining.

Another major bottleneck arises at inference time: sampling typically requires tens to hundreds of neural function evaluations to denoise the sample, rendering such models impractical for real-time or latency-constrained scenarios. To address this, recent approaches have introduced dramatically accelerated alternatives. Consistency Models [5] train a single network to map directly from noise to data in one step (though multi-step refinement remains optional), Shortcut Models [6] condition the generator on the desired step budget and DPM-Solver++ [7] improves upon numerical ODE-based sampling methods by employing a third-order integrator.

In parallel, there has been a growing shift in research focus from image-based diffusion models to video diffusion models, with efforts either repurposing pretrained image models or designing architectures specifically tailored for video generation. Video Diffusion Models [8] extended the standard 2D diffusion process into the temporal dimension, laying foundational groundwork for video

generation, but struggled with spatio-temporal coherence. Latent Video Diffusion [9] introduced a more computationally efficient pipeline by operating in a compressed latent space, yet temporal smoothness remained an open problem. More recent work such as Latte [10] incorporated transformer-based architectures within the latent diffusion framework, leading to improved modeling of long-range temporal dependencies.

Our work began with an exploration of compositional diffusion methods [3, 4], but we found these approaches to be unstable and computationally expensive in practice. This limitation motivated a shift toward directly manipulating the noise space of pretrained diffusion models. Inspired by recent works such as **A Noise is Worth Diffusion Guidance** [11] and **PYoCo** [12], our method seeks to map a reference noise vector to a sequence of temporally aligned noise samples that, once denoised, yield smooth and consistent video frames—all without modifying the base diffusion model or requiring access to motion labels.

2 Related Work / Reviewed Work

2.1 Compositional Generation

Compositional generation addresses the challenge of synthesizing novel combinations not seen during training by combining pretrained diffusion models without retraining.

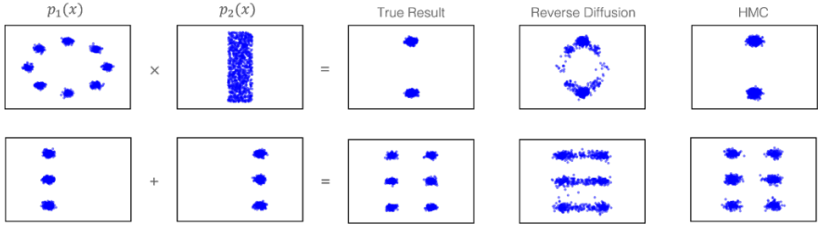


Fig. 1: Reverse Diffusion is the process of direct summation of the scores of the diffusion priors, whereas HMC is the Hamiltonian Markov Chain sampling algorithm employed over the energy based parametrization of the diffusion priors.

Composable diffusion [3] frames each pretrained diffusion model as an energy-based component representing a specific concept. At inference, multiple concepts are composed by summing their score functions during Langevin dynamics, enabling structured generalization to unseen attribute combinations. While compelling, this naive summation method lacks theoretical justification and often fails in practice.

To remedy this *Reduce, Reuse, Recycle* [4] observes that compositional failure is rooted in sampling dynamics rather than modeling alone. They introduce

explicit energy-based parameterizations of diffusion priors and employ sophisticated MCMC samplers. A smaller example is shown in figure 1 to showcase how these samplers improve upon the previous method. Though this approach improves compositional fidelity, it incurs at least five-fold greater computational overhead and suffers from slow convergence and instability inherent to energy-based sampling.

2.2 Training Energy Based Models

Energy-based models (EBMs) offer flexible generative modeling, but their training and inference suffer from instability and high computational cost, primarily due to reliance on MCMC sampling and intractability of the partition function. The Diffusion Recovery Likelihood (DRL) [13] framework addresses this by training a sequence of EBMs on increasingly noisy versions of the data. This formulation simplifies sampling dynamics and delivers improved sample quality.

To further improve efficiency and sample quality, Cooperative Diffusion Recovery Likelihood (CDRL) [14] couples each EBM with a neural initializer that proposes better starting point that is subsequently refined via a few MCMC steps. Through cooperative training, the initializer learns to mimic the EBM's transitions, accelerating convergence, improving stability, and effectively halving the inference cost compared to DRL while the performance gap with diffusion models.

Inspired by **Jump Restore Light Transport** [15] we attempted jump based acceleration for EBM training samples. While this yielded improvements in low dimensional settings, it failed to scale effectively to high-dimensional domains such as images. An example of our attempts in lower dimensions can be found in figure 2.

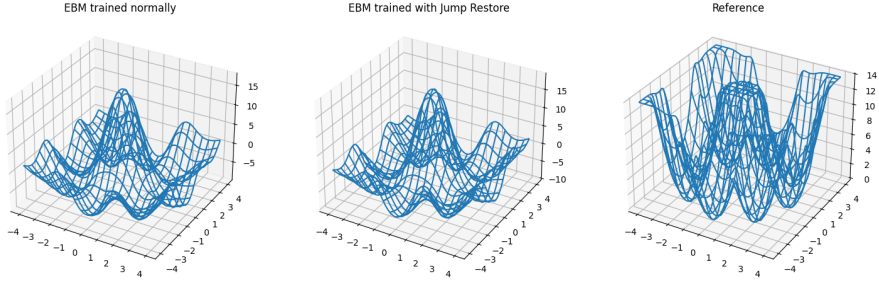


Fig. 2: Representation of the negative log likelihood of an analytical function that we used to train our EBMs. Trained on a 4070 Laptop, Normal training took 66secs while Jump Restore took around 40secs.

2.3 Video Diffusion

Ho et al. [8] introduced the foundational Video Diffusion Model (VDM), which extends image-based diffusion to videos using a space-time factorized 3D U-Net architecture. Spatial 2D convolutions are replaced with space-only 3D convolutions, and after each spatial block a temporal attention block aggregates information across frames, enabling effective temporal modeling while preserving computational efficiency. Subsequently, **Latent Video Diffusion** [9] achieves significant computational savings by leveraging a pretrained image latent diffusion model-typically Stable Diffusion-as the backbone and inserting trainable temporal layers atop its frozen spatial components. This strategy yields high-resolution video synthesis (up to 1280x2048) with temporal coherence and without retraining a full model from scratch. A general pipeline of modern video diffusion models is shown in figure 3

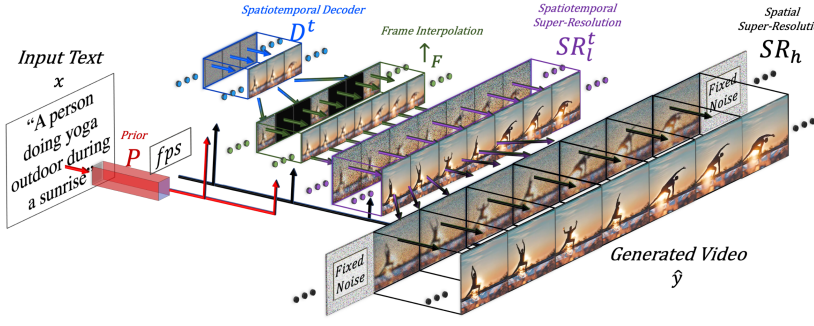


Fig. 3: General architecture of most larger scale video diffusion models. Image source: Make-A-Video [16]

Recent advances have explored Transformer-based architectures for video generation, with Latte [10] representing a notable shift from U-Net to Transformer backbones. Latte extracts spatio-temporal tokens from input videos and processes them through Transformer blocks in latent space. The Transformer architecture's flexibility has proven particularly advantageous for video generation, as demonstrated by models like Sora [17], which leverage scalable attention mechanisms to handle variable-duration, resolution, and aspect ratio videos.

2.4 Noise-Space Manipulation for Efficient and Coherent Generation

A growing body of work suggests that manipulating the noise space of pretrained diffusion models can lead to efficient and high-quality coherent generation. In **A Noise is Worth Diffusion Guidance** [11] the authors empirically show that certain structured initial noise vectors, referred to as guidance free noise,

can yield high-quality images without any guidance during inference. To operationalize this, they introduce NoiseRefine, a lightweight network that transforms standard Gaussian noise into the guidance-free noise space in a single forward pass, enabling faster sampling.

Recent works have demonstrated that temporal coherence in video generation can be achieved by explicitly structuring the evolution of noise vectors over time, rather than modifying the diffusion model itself. **How I warped your noise** [18] proposes Integral Warp, a method that temporally advects a single Gaussian noise vector across video frames using optical flow—either estimated between frames or provided as conditioning. The warping is applied directly in noise space, preserving the Gaussian statistical properties of the transformed vectors. **Go with the Flow** [19] eliminates the need for flow-based warping by learning a noise trajectory generator that predicts a smooth sequence of temporally correlated noise vectors conditioned on motion cues. While both methods leverage structured noise to produce temporally consistent videos, the former relies on explicit motion fields to guide noise transformation, whereas the latter learns motion-aligned trajectories end-to-end, enabling controllable generation without external flow estimation. Together, these approaches underscore the effectiveness of noise-space manipulation as a lightweight and scalable alternative to training full-fledged video diffusion models.

PYoCo [12] observes that when video frames are inverted through an image diffusion model, their corresponding noise vectors lie in a tightly clustered manifold, revealing inherent temporal structure in noise space. This can be seen in figure 4

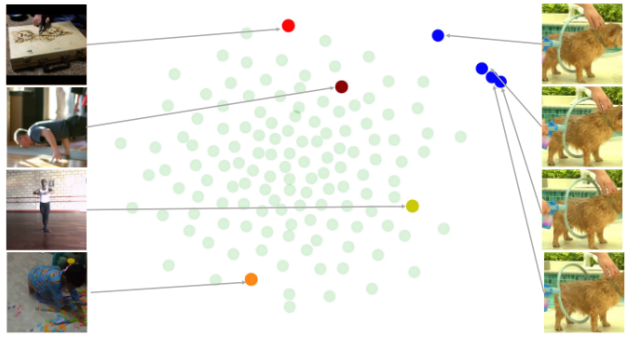


Fig. 4: visualizes the t-SNE plot of the noise maps corresponding to input frames randomly sampled from videos. These noise maps are obtained by inverting the images using an image diffusion model. The green dots in the background denote the reference noise maps sampled from an i.i.d. Gaussian distribution. The red dots and yellow dots are noise maps corresponding to input frames coming from different videos. It was found that they are spread out and share no correlation. On the other hand, the noise maps corresponding to the frames coming from the same video (shown in blue dots) are clustered together. Image source: PYoCo [12]

Building on this insight our model attempts to predict such temporally consistent noise manifold. In contrast to approaches that require optical flow estimation [18] or motion conditioning [19], we propose to directly learn frame-to-frame noise mappings that capture temporal continuity using only the structural priors embedded in pretrained image diffusion models.

3 Method / Research

This section presents the step-by-step progression of our noise-space video generation framework. We begin by defining the initial objective, which is to learn a mapping function that transforms a Gaussian noise vector into temporally structured noise, enabling a frozen image diffusion model to generate coherent video. We then introduce two distinct training techniques, each exploring a different supervision paradigm and loss calculation. Based on observed limitations in temporal coherence, applicability, and computational efficiency, we propose a revised mapping objective that better aligns with the intrinsic characteristics of video generation. Finally, we describe the final training protocol, incorporating insights from previous iteration to enhance stability, consistence, and compute friendly training.

3.1 Goal

Given an initial noise vector $\epsilon \sim \mathcal{N}(0, I)$, we want to define a learnable mapping function f_ϕ which produces a temporal sequence of noise vectors $\tilde{\epsilon} = f_\phi(\epsilon) = \left[\tilde{\epsilon}_1, \tilde{\epsilon}_2, \ldots, \tilde{\epsilon}_T\right]$, where each $\tilde{\epsilon}_t$ is a noise frame which when denoised via a pretrained image model D_{img}, yeilds $V_{\text{gen}} = \left[D_{\text{img}}(\tilde{\epsilon}_1), D_{\text{img}}(\tilde{\epsilon}_2), \ldots, D_{\text{img}}(\tilde{\epsilon}_T)\right]$.

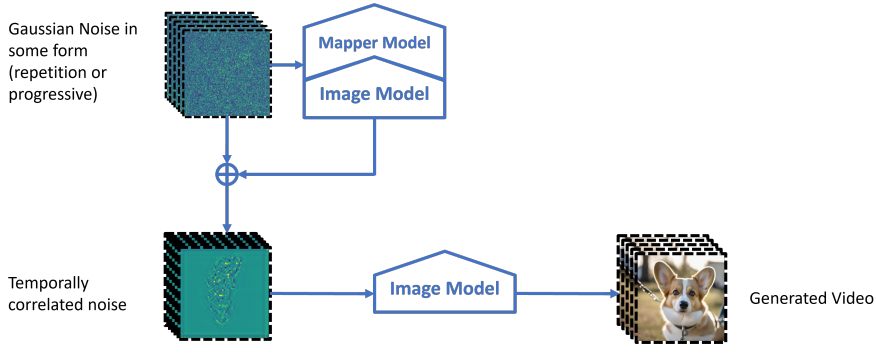


Fig. 5: Our noise-mapping model takes as input a Gaussian noise vector—either replicated or progressively correlated and transforms it into a noise sequence enriched with temporal dependencies. When this mapped noise is passed through a pretrained image diffusion model (applied frame-by-frame), it yields video frames that exhibit temporal coherence.

3.2 Paradigm One - Training on inverted noise

Let a training video be $X_{1:\mathrm{T}} = \{X_1, \ldots, X_\mathrm{T}\}$. Using a frozen image diffusion model D_{img} and its inversion operator $\mathcal{I}_{\mathrm{img}}$. We use this to compute frame wise target noise latents represented by

$$\hat{\epsilon}_{1:T} \;=\; \mathcal{I}_{\mathrm{img}}(X_{1:T}) \;=\; \big[\hat{\epsilon}_1, \ldots, \hat{\epsilon}_T\big]. \tag{1}$$

We now want to train a model f_ϕ to reproduce these latents from a base gaussian input

$$\epsilon \sim \mathcal{N}(0, I), \qquad \tilde{\epsilon}_{1:T} \;=\; f_\phi(\epsilon), \tag{2}$$

by minimizing the $\mathcal{L}_2$ norm between $\tilde{\epsilon}_{1:\mathrm{T}}$ and $\hat{\epsilon}_{1:\mathrm{T}}$.

Limitation - Because f_ϕ is trained to match $\mathcal{I}_{img}(X_{1:\mathrm{T}})$ without propogating through the image model, it effectively acts as an one step generator of temporal noise decoupled from the denoiser's dynamics. This leads to a reduced fidelity in final generation as seen in the ablation studies by Ahn et. al [11].It is hence better to differentiate through D_{img} or distill the information from a teacher's guidance.

3.3 Paradigm Two - Training via distillation

Sample a noise vector $\epsilon \sim \mathcal{N}(0, I)$.Using a pre-trained video diffusion model D_{vid}, generate a target video $V_{ref} = D_{vid}(\epsilon_{1:T})$. The learnable mapping model f_ϕ converts the same noise vector into temporally structured noise sequence

$$\epsilon \sim \mathcal{N}(0, I), \qquad \tilde{\epsilon}_{1:T} \;=\; f_\phi(\epsilon). \tag{3}$$

Each $\tilde{\epsilon}_t$ is then denoised by a frozen image diffusion model D_{img} to produce the generated video

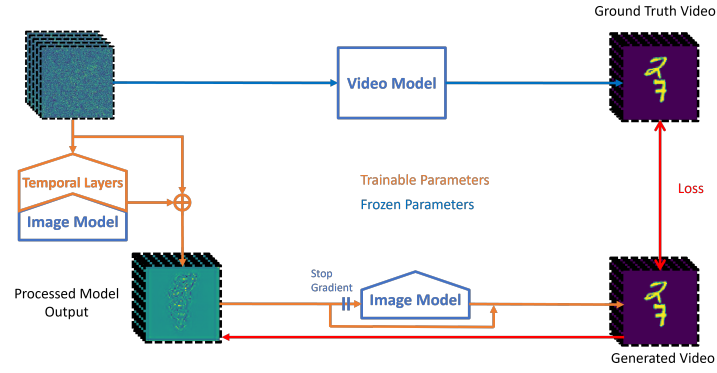
$$V_{\mathrm{gen}} = \big[D_{\mathrm{img}}(\tilde{\epsilon}_1), D_{\mathrm{img}}(\tilde{\epsilon}_2), \ldots, D_{\mathrm{img}}(\tilde{\epsilon}_T)\big]. \tag{4}$$

The training objective minimizes the framewise $\mathcal{L}_2$ loss with gradients propogated only through f_ϕ as both pretrained models remain frozen.

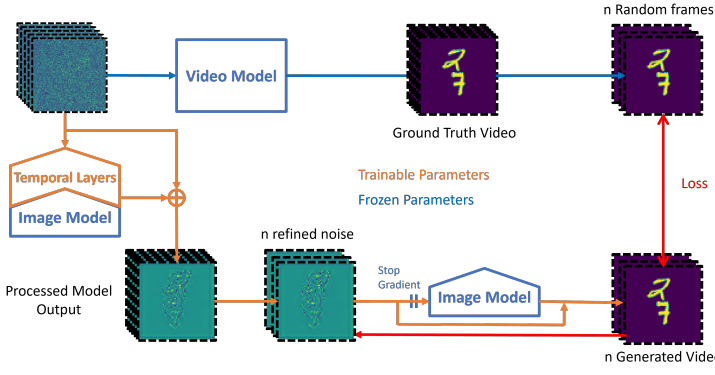
Limitations - Each iteration requires a full video-model denoising pass to obtain V_{ref} and T per-frame image-model denoising passes to obtain V_{gen}. Even with cost cutting methods such as reduced-step samplers or reduced frame sampling, the training remains compute-intensive and requires access to strong pretrained video and image diffusion models.

3.4 Updated Goal - Noise Modulation During Denoising

Given the limitations of the preceding methods, we revise the objective to inject temporal structure inside the reverse diffusion process of a frozen image model, rather than only setting up the initial noise. Concretely, we aim to steer the noise-to-image trajectory so that a single denoising run can be branched into multiple, closely related trajectories whose final images form a temporally coherent video.



(a)



(b)

Fig. 6: This the distillation training paradigm we can use to distill the training knowledge of a frozen video model to our mapping model. (a) is the compute heavy version of training, where as (b) will have a lower backward graph hence reducing memory requirements while getting similar results.

Operationally, the steps involved in this process are -

1. Initialize. Start from one image-shaped noise sample at the terminal noise level and begin denoising with the frozen image denoiser.

2. Step-wise branching. At selected reverse steps, apply a learnable, step-conditioned transition that takes the current noisy latent and a desired frame offset (how far ahead in "time" we wish to move), and returns a new latent at the same noise level. This transition preserves the marginal noise statistics but introduces controlled correlation with the source latent.

3. Parallel denoising. Continue denoising both the original and the branched latents in parallel. Repeat branching at subsequent steps until the required number of frame trajectories is obtained.

4. Finalize. Run all trajectories to the clean endpoint to obtain a sequence of frames that maintain per-frame fidelity while exhibiting temporal coherence.

For compactness, the transition can be written as:

$$x_j \leftarrow g_\phi(x_i, s, \Delta_{ij}), \tag{5}$$

indicating that the learnable operator g_ϕ acts at denoising step s and does not change the noise level, but nudges the latent to be temporally related to x_i. Here i and j are the frame time and Δ_{ij} is the time offset between the two. This formulation leverages the full diffusion trajectory, avoids dependence on a separate video teacher, and concentrates learning on a small operator that modulates noise within denoising.

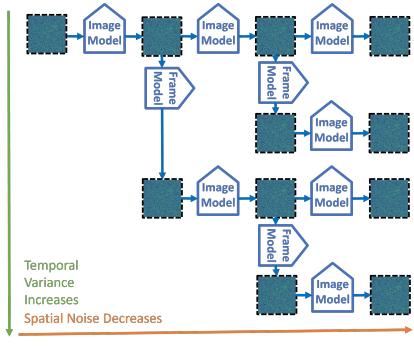


Fig. 7: Temporal perturbations are introduced during reverse diffusion at higher-noise steps, where the spatial noise budget permits larger, controlled modulations of the latent. This enables larger temporal offsets between branched trajectories while preserving the per-step noise level. As denoising progresses and spatial noise diminishes, modulation strength is reduced to maintain per-frame fidelity; consequently, earlier (noisier) states afford greater modulation capacity and exert a proportionally larger influence on the final video frames.

3.5 Final Training Paradigm

Given a video $V = \{X_1, \ldots, X_T\}$, we sample two frames X_t and $X_{t+\Delta}$. We add noise to X_t exactly as in standard image diffusion training: at a sampled step s with schedule (α_s, σ_s) to obtain a noisy latent:

$$x_s^{(t)} \;=\; \alpha_s\, X_t \;+\; \sigma_s\, z, \qquad z \sim \mathcal{N}(0, I), \tag{6}$$

A learnable, step-conditioned transition operator g_ϕ maps $x_s^{(t)}$ and the desired temporal offset Δ to a *new latent at the same noise level*:

$$\hat{x}_s^{(t+\Delta)} \;=\; g_\phi\big(x_s^{(t)},\, s,\, \Delta\big). \tag{7}$$

intended to be temporally correlated with $x_s^{(t)}$ while preserving the step-s noise.

We then pass $\hat{x}_s^{(t+\Delta)}$ through the frozen image diffusion model to get a single-step denoised estimation $\hat{X}_{t+\Delta}$. The supervision is the ground-truth frame $X_{t+\Delta}$. The training loss is an $\mathcal{L}_2$ term between the target and the prediction, while the gradients are propogated only through g_ϕ keeping the image model frozen. This yields an efficient training signal that exploits the diffusion trajectory while avoiding full-length video-teacher rollouts.

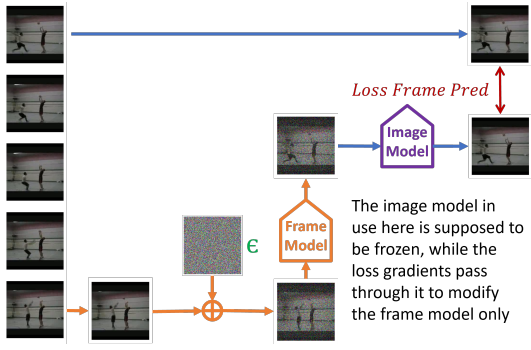


Fig. 8: Final Training Methodology

4 Experiments

We evaluate our framework on the Sky Timelapse dataset, which provides naturally coherent sequences well-suited for assessing temporal consistency in generative models. As a baseline, we compare against Latte, a state-of-the-art latent video diffusion model, while employing a frozen DiT image diffusion model as the backbone in our method. Training was performed on a single NVIDIA H100 GPU with a batch size of 16, learning rate of 1×10^{-4}, and Adam optimizer. Each video sample consisted of 16 frames. The model required approximately 20 hours of training to reach visually coherent outputs.

Qualitative inspection reveals that the generated videos exhibit smooth temporal transitions, though individual frames tend to appear slightly more blurred compared to the initial sample. We also observe that branching for new frame predictions near the later stages of the denoising process leads to improved temporal fidelity and stability, suggesting that temporal modulation is more effective when applied at lower-noise states. These findings highlight both the strengths and limitations of our approach: while capable of preserving coherence and leveraging pretrained image priors, it remains sensitive to the denoising schedule and exhibits some degradation in frame sharpness.

These results validate the feasibility of noise-space manipulation for video generation while pointing toward refinements in noise scheduling and sharpness enhancement as future directions.

Fig. 9: Sample of a generated video using our method. The frames go from left to right then top to bottom. This was generated when we branched off from the last 50 steps.

5 Future Work

While our current framework demonstrates the feasibility of manipulating noise trajectories for video generation, several promising directions remain open for exploration. Future research can aim to enhance temporal coherence, scalability, and quality of generated videos by leveraging alternative formulations of dynamics, more structured latent representations, and improved loss functions for our noise manipulation. Below, we outline three potential avenues that could guide subsequent work.

One promising direction is to learn a continuous dynamics model (e.g. a neural ODE) that transforms the start frame into the end frame over time, such that every intermediate time corresponds to a plausible in-between frame. In this approach, the first frame's representation would be treated as an initial state, and a learned differential equation would be integrated from time $t = 0$ (start) to $t = 1$ (end) to produce the final frame's representation. All intermediate states along this trajectory would then decode to intermediate frames. Unlike discrete or recurrent models, a continuous-time ODE formulation allows synthesizing frames at arbitrary time points (for interpolation or extrapolation) using a single unified model. In other words, by "solving" the learned ODE forward for fractional timesteps, one could generate any intermediate frame between the given start and end frames. This approach would essentially treat video synthesis as integrating a learned flow field over time. By learning a continuous transformation

between frames, the model might better capture temporal dynamics and avoid the jitter or discontinuities that purely discrete models sometimes exhibit.

Another complementary direction is to learn a latent space in which a video's frames lie along a simple linear path, enabling easy interpolation by linear movement in that latent space. For example, one could train a Variational Autoencoder (VAE) (or a flow-based model) to encode each frame of a video such that the latent codes for a sequence of frames form a roughly linear trajectory (as opposed to an arbitrary curve) in the latent manifold. If successful, this would mean that given the latent code of the first and last frame, any point along the straight line segment connecting these two codes should decode to a realistic intermediate frame. Prior work on frame interpolation provides some evidence that this idea is feasible: even if the true latent manifold of images is nonlinear, approximating it with a straight line between two endpoint frame embeddings can yield reasonable intermediate frames [20]. Future models could explicitly encourage such linear latent progressions, perhaps by adding a regularization term that penalizes curvature in the latent trajectory or by designing the latent dynamics to follow a straight line (e.g. constant-velocity latent movement). If successful, this would allow video synthesis by simply taking an image's latent and moving in a straight line direction to create a smooth sequence – essentially treating the latent space as a linear motion space for that video.

Enhancing the training objective for our mapping model another important avenue for future improvement. Better loss function could significantly improve the realism and temporal coherence of the generated frames. Possible enhancements include perceptual loss, temporal consistency loss, noise aware control, optical flow loss, etc.

These ideas are hypothetical but grounded in trends seen in recent research – for instance, continuous latent dynamics for video generation, linear latent interpolation for frame synthesis, and specialized loss terms for perceptual fidelity and temporal consistency. Pursuing such directions could significantly push the capabilities of video synthesis models in future work.

References

1. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models (2020)
2. Chen, M., Mei, S., Fan, J., Wang, M.: An overview of diffusion models: Applications, guided generation, statistical rates and optimization (2024)
3. Liu, N., Li, S., Du, Y., Torralba, A., Tenenbaum, J.B.: Compositional visual generation with composable diffusion models (2023)
4. Du, Y., Durkan, C., Strudel, R., Tenenbaum, J.B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., Grathwohl, W.: Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc (2024)
5. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models (2023)
6. Frans, K., Hafner, D., Levine, S., Abbeel, P.: One step diffusion via shortcut models (2025)
7. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. Machine Intelligence Research **22**(4) (June 2025) 730–751
8. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models (2022)
9. He, Y., Yang, T., Zhang, Y., Shan, Y., Chen, Q.: Latent video diffusion models for high-fidelity long video generation (2023)
10. Ma, X., Wang, Y., Chen, X., Jia, G., Liu, Z., Li, Y.F., Chen, C., Qiao, Y.: Latte: Latent diffusion transformer for video generation (2025)
11. Ahn, D., Kang, J., Lee, S., Min, J., Kim, M., Jang, W., Cho, H., Paul, S., Kim, S., Cha, E., Jin, K.H., Kim, S.: A noise is worth diffusion guidance (2024)
12. Ge, S., Nah, S., Liu, G., Poon, T., Tao, A., Catanzaro, B., Jacobs, D., Huang, J.B., Liu, M.Y., Balaji, Y.: Preserve your own correlation: A noise prior for video diffusion models (2024)
13. Gao, R., Song, Y., Poole, B., Wu, Y.N., Kingma, D.P.: Learning energy-based models by diffusion recovery likelihood (2021)
14. Zhu, Y., Xie, J., Wu, Y.N., Gao, R.: Learning energy-based models by cooperative diffusion recovery likelihood. In: The Twelfth International Conference on Learning Representations. (2024)
15. Holl, S., Seidel, H.P., Singh, G.: Jump restore light transport (2024)
16. Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., Parikh, D., Gupta, S., Taigman, Y.: Make-a-video: Text-to-video generation without text-video data (2022)
17. Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R., Yuan, Z., Huang, Y., Sun, H., Gao, J., He, L., Sun, L.: Sora: A review on background, technology, limitations, and opportunities of large vision models (2024)
18. Chang, P., Tang, J., Gross, M., Azevedo, V.C.: How i warped your noise: a temporally-correlated noise prior for diffusion models. In: The Twelfth International Conference on Learning Representations. (2024)
19. Burgert, R., Xu, Y., Xian, W., Pilarski, O., Clausen, P., He, M., Ma, L., Deng, Y., Li, L., Mousavi, M., Ryoo, M., Debevec, P., Yu, N.: Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise (2025)
20. Nguyen, A.D., Kim, W., Kim, J., Lee, S.: Video frame interpolation by plug-and-play deep locally linear embedding (2018)