ADAPTIVE WEIGHTED h-TRANSFORM SAMPLING FOR COARSE-GUIDED VISUAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Coarse-guided visual generation, which synthesizes fine visual samples from degraded or low-fidelity coarse references, is essential for various real-world applications. While training-based approaches are effective, they are inherently limited by high training costs and restricted generalization due to paired data collection. Accordingly, recent training-free works propose to leverage pretrained diffusion models and incorporate guidance during the sampling process. However, these training-free methods either require knowing the forward (fine-to-coarse) transformation operator, *e.g.*, bicubic downsampling, or are difficult to balance between guidance and synthetic quality. To address these challenges, we propose a novel guided method by using the h-transform, a tool that can constrain stochastic processes (*e.g.*, sampling process) under desired conditions. Specifically, we modify the transition probability at each sampling timestep by adding to the original differential equation with a drift function h, which approximately steers the generation toward the ideal fine sample. To address unavoidable approximation errors, we introduce an adaptive weight scheduler that combines a noise-level-aware initialization with a correction based on cross-timestep consistency, balancing guidance adherence and synthesis quality. Extensive experiments across diverse image and video generation tasks demonstrate its effectiveness and generalization.

1 INTRODUCTION

Recent advances in diffusion (Ho et al., 2020; Song et al., 2020) and flow matching (Liu et al., 2022; Lipman et al., 2022) models have demonstrated remarkable visual generation capability. Benefiting from large-scale pretraining, high-quality samples can be generated by existing unconditional and text-to-visual (T2V) models (Rombach et al., 2022; Labs, 2024; Wan et al., 2025) in the case of no guidance or only with text guidance. However, some real-world applications (*e.g.*, deblurring and super-resolution) always require a coarse visual sample to guide synthesis, which these pretrained unconditional and T2V models cannot handle directly. Thus, recent works (Song et al., 2020; Choi et al., 2021; Chung et al., 2022a; Kawar et al., 2022; Chung et al., 2022b; Kim et al., 2025b; Meng et al., 2021; Singer et al., 2025) aim to solve such a **coarse-guided visual generation** task: given a *coarse visual sample* (*e.g.*, a blurred image, or a warped video) as the guidance, the model needs to generate the *corresponding refined sample* (*e.g.*, a clear image, or an undistorted video).

A straightforward approach is to train translation networks. Shown in Fig. 1(a), previous studies (Zhou et al., 2023; De Bortoli et al., 2021; Zhang et al., 2023) try to train from scratch or fine-tune a translation model with paired coarse-fine data. For different types of coarse samples (*e.g.*, low-resolution images, blurred images, and warped videos), additional/existing models must be trained/fine-tuned with the corresponding paired data. Thus, these training-based approaches suffer from inherent *high costs* (*i.e.*, training and data collection) and *limited generalization*.

Subsequent work resorts to training-free solutions that only modify the sampling process to achieve guided generation. There are two mainstream training-free strategies: 1) **Solving Inverse Problem** (Song et al., 2020; Choi et al., 2021; Chung et al., 2022a; Kawar et al., 2022; Chung et al., 2022b; Kim et al., 2025b): As shown in Fig. 1(b), instead of sampling from the marginal distribution, they approximate the posterior probability term and sample directly from the conditional distribution. However, the approximation used in these methods usually relies on a known forward fine-to-coarse operator, *i.e.*, the transformation from fine samples to coarse samples must be

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

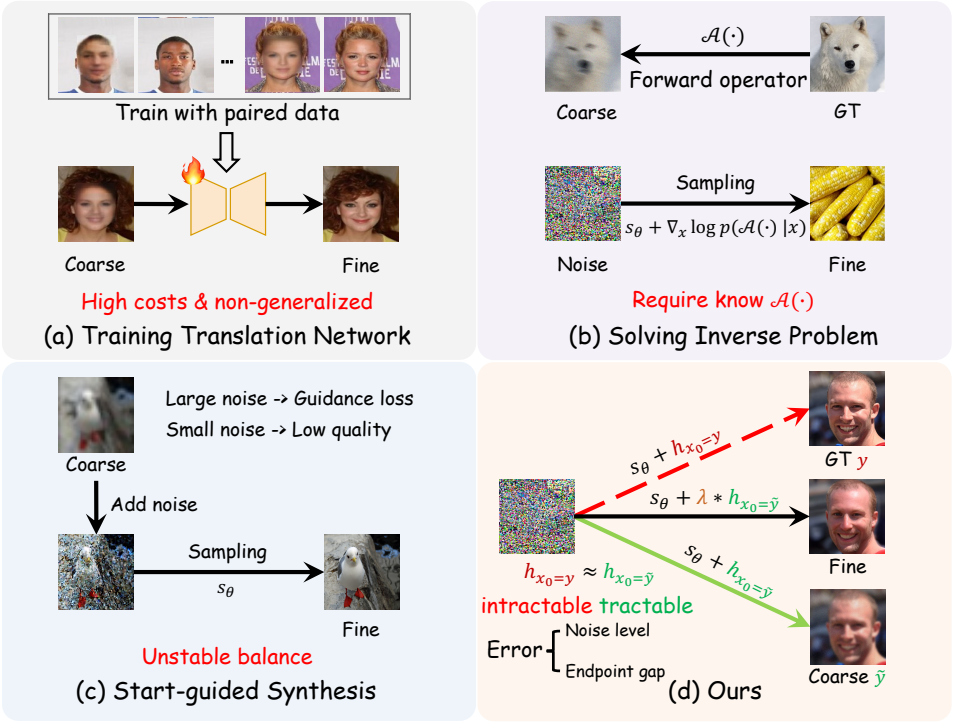

Figure 1: **Existing and our solutions.** (a) Training translation networks based on paired data, which is costly and non-generalizable to different types of coarse samples. (b) Solving inverse problems based on a known forward operator, which is not robust. (c) Adding noise to the coarse sample and denoising it, which is difficult to balance guidance and quality. (d) Our method leverages the h-transform to achieve training-free, operator-free, and stable coarse-guided generation.

known, such as bicubic downsampling or a Gaussian mask operator. Thus, this prior (known operator) requirement reduces the robustness of these methods, as the operator may be unknown in many scenarios. 2) **Start-Guided Synthesis** (Meng et al., 2021; Singer et al., 2025): As shown in Fig. 1(c), they first get the sampling start by adding noise to the coarse sample. Then they can sample from it with a pretrained diffusion model to get the refined sample. However, the guidance is entirely dependent on setting the noisy coarse sample as the starting point. Adding a large amount of noise results in loss of guidance signals, whereas adding a small amount of noise yields only limited quality improvement. In that case, this start-dependent guided manner leads to an *unstable balance* between guidance faithfulness and generation quality.

Based on the above discussions, we observe that current solutions are constrained by training, the requirement for known operators, and an unstable balance. To this end, inspired by Doob's h-transform (Doob & Doob, 1984; Rogers & Williams, 2000), a tool that can constrain the sampling process with the given guidance, we propose a novel coarse-guided generation method, **Adaptive Weighted h-Transform Sampling**. Intuitively, our method incorporates guidance signals during sampling by modifying the transition probability at each time step and finally generates a guided refined result. This probability modification can be interpreted as introducing additional traction toward the *ideal underlying refined result* on top of the original sampling route.

Specifically, we achieve traction by adding a new drift adjustment $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}$ ($\boldsymbol{x}$ is the time-index updating latent and $\boldsymbol{y}$ is the ideal result) to the original predicted score s_θ (*c.f.*, Figure 1(d)). By sampling with a new score ($s_\theta + \boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}$), the result can be guaranteed to be $\boldsymbol{y}$. However, this $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}$ is not tractable because it depends on knowing the ground truth, *i.e.*, the ideal result $\boldsymbol{y}$. Since we can not directly obtain $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}$, we found that we can leverage another tractable $\boldsymbol{h}_{\boldsymbol{x}_0=\widetilde{\boldsymbol{y}}}$ to approximate it, where $\widetilde{\boldsymbol{y}}$ is the given coarse sample. Subsequently, we analyze the approximation error and find that it depends on two factors: 1) It is negatively correlated with the **noise level** of $\boldsymbol{x}$. To mitigate the error influence, we use a noise-level-aware schedule as initialization to adjust the weight of $\boldsymbol{h}_{\boldsymbol{x}_0=\widetilde{\boldsymbol{y}}}$ across time steps. That is, as the approximation error increases, we decrease the weight of $\boldsymbol{h}_{\boldsymbol{x}_0=\widetilde{\boldsymbol{y}}}$

during the sampling process. 2) It is proportional to the **endpoint gap** (the distance between ideal $\boldsymbol{y}$ and coarse $\widetilde{\boldsymbol{y}}$). Thus, we build a cross-time consistency criterion as the endpoint gap proxy to further adjust the noise-level-aware initialization into a better adaptive weight for each timestep. In this way, the generated sample not only adheres to the coarse guidance but also improves in quality.

Our contributions are threefold: 1) We propose *Adaptive Weighted h-Transform Sampling*, a training-free method for coarse-guided visual generation. It derives a tractable coarse-endpoint approximation to the ideal h-transform and requires no known forward degradation operator. 2) To mitigate the influence of approximation error, we factor the error into the noise level and unknown endpoint gap. Based on these two guidelines, we design an adaptive weight schedule that contains a noise-level-aware initialization complemented by a consistency correction. 3) Extensive experiments across different image and video generation tasks show our effectiveness and generalization.

2 Background

2.1 Diffusion Models

Diffusion models capture the target data distribution p_0 by learning a transport process from a prior distribution p_T (*e.g.*, a Gaussian) to p_0. To achieve that, a forward diffusion process from p_0 to p_T is first determined and can be described with such a stochastic differential equation (SDE):

$$\mathrm{d}\boldsymbol{x} = \mathbf{f}(\boldsymbol{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \tag{1}$$

where $t \in [0, T]$ is the time-index, $\boldsymbol{x}_0 \sim p_0$, $\mathbf{f}$ is the drift function, g is the diffusion coefficient, and $\mathbf{w}$ is a Brown motion. Meanwhile, the marginal distribution of $\boldsymbol{x}_t$ at time t corresponding to this SDE is $p_t(\boldsymbol{x}_t)$. Then, we have a reversed SDE to describe the reversed process (from p_T to p_0):

$$\mathrm{d}\boldsymbol{x} = [\mathbf{f}(\boldsymbol{x}_t, t) - g^2(t)\nabla_{x_t} \log p_t(\boldsymbol{x}_t)]\mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}, \tag{2}$$

where $\overline{\mathbf{w}}$ is the reversed Brown motion. According to the Fokker–Planck equation (Maoutsa et al., 2020) (F-P Equation), we can convert this reversed SDE into an ordinary differential equation (ODE) that has the same marginal distribution $p_t(\boldsymbol{x}_t)$:

$$\mathrm{d}\boldsymbol{x} = [\mathbf{f}(\boldsymbol{x}_t, t) - \frac{1}{2}g^2(t)\nabla_{x_t} \log p_t(\boldsymbol{x}_t)]\mathrm{d}t. \tag{3}$$

We can sample synthetic samples by solving the SDE in Eq. (2) or the ODE in Eq. (3). However, $\nabla_{x_t} \log p_t(\boldsymbol{x}_t)$ (also known as the score) is not known. Thus, a parameterized network $s_\theta(\cdot)$ with parameters θ can be trained to predict it. The optimization target is:

$$\min_\theta \mathbb{E}_{\boldsymbol{x}_t, \boldsymbol{x}_0, t}\left[||s_\theta(\boldsymbol{x}_t, t) - \nabla_{x_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)||_2^2\right], \tag{4}$$

where $\boldsymbol{x}_0 \sim p_0$, $t \in [0, T]$ and $\boldsymbol{x}_t \sim p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$. $\nabla_{x_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)$ is **tractable** since the conditional distribution is defined by the forward SDE in Eq. (1), and it has closed-form solutions. After training, s_θ can be used as a score predictor to replace $\nabla_{x_t} \log p_t(\boldsymbol{x}_t)$ for solving Eq. (2) or Eq. (3).

2.2 Doob's h-Transform

For a given SDE like Eq. (1), the corresponding process can transport $\boldsymbol{x}_0 \sim p_0$ to Gaussian noise. However, in some situations, such as image translation, we may need to establish a mapping relation between $\boldsymbol{x}_0$ and a fixed point $\boldsymbol{y}$. Thus, Doob's h-transform (Doob & Doob, 1984; Rogers & Williams, 2000) proposes to modify the probability transition of the original SDE by adding a $g^2(t)\boldsymbol{h}_{\boldsymbol{x}_T=\boldsymbol{y}}$ term to the original drift term $\mathbf{f}$:

$$\mathrm{d}\boldsymbol{x} = [\mathbf{f}(\boldsymbol{x}_t, t) + g^2(t)\boldsymbol{h}_{\boldsymbol{x}_T=\boldsymbol{y}}]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}, \tag{5}$$

where $\boldsymbol{h}_{\boldsymbol{x}_T=\boldsymbol{y}}$ is called the h function. The new transition guarantees that the SDE ends at the fixed point $\boldsymbol{y}$, where $\boldsymbol{h}_{\boldsymbol{x}_T=\boldsymbol{y}}$ is designed as $\nabla_{x_t} \log p_t(\boldsymbol{x}_T = \boldsymbol{y}|\boldsymbol{x}_t)$. It improves the transition probability of $p_t(\boldsymbol{x}_T = \boldsymbol{y}|\boldsymbol{x}_t)$ and guarantees $\boldsymbol{x}_T = \boldsymbol{y}$ by adjusting the drift coefficient across timesteps.

3 Adaptive Weighted h-Transform Sampling

Problem Formulation. Assuming that an underlying joint distribution $(\widetilde{\boldsymbol{y}}, \boldsymbol{y}) \sim q$ describes the probability measurement between a coarse visual sample $\widetilde{\boldsymbol{y}}$ and its corresponding fine sample $\boldsymbol{y}$. For the coarse-guided visual generation task, given a coarse sample as the guidance $\widetilde{\boldsymbol{y}}$, our objective is to leverage a pretrained diffusion model s_θ to generate a corresponding fine sample.

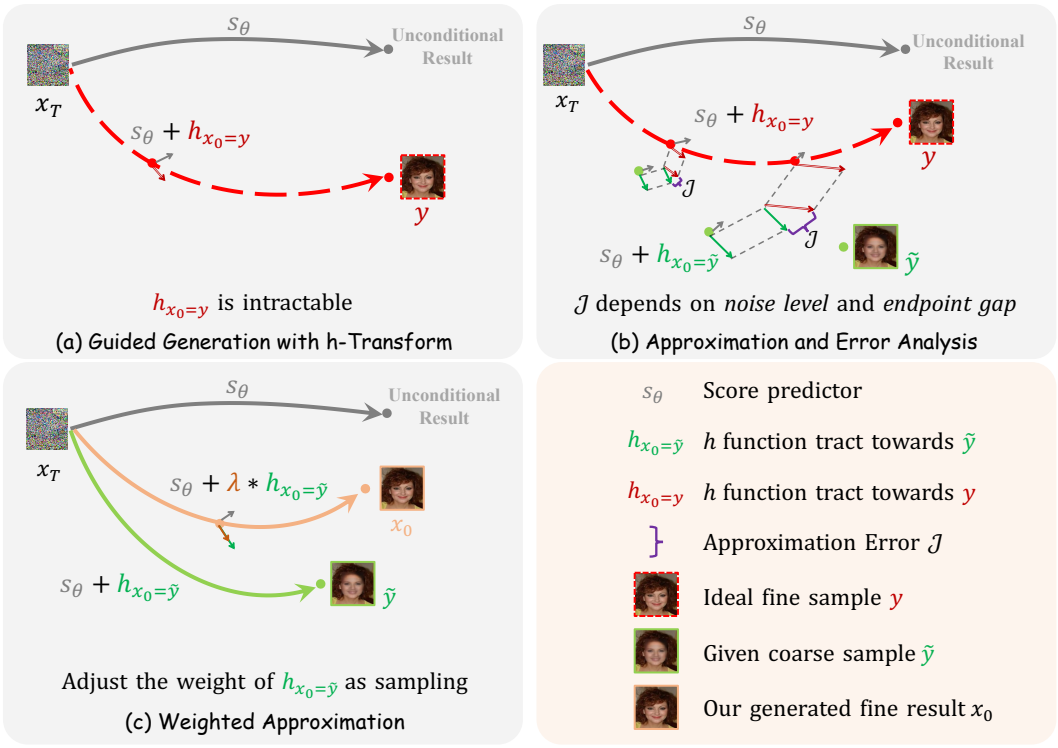

Figure 2: **Overview of Weighted h-Transform Sampling**. (a) If we have $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}$, the generation result will be the ideal sample. (b) We leverage $\boldsymbol{h}_{\boldsymbol{x}_0=\tilde{\boldsymbol{y}}}$ to approximate the intractable $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}$ and derive that the error depends on the noise level and endpoint gap. (c) To mitigate the error influence, we adjust the approximation weight and finally generate a high-quality refined sample.

3.1 Guided Generation with h-Transform

If the ideal fine sample $\boldsymbol{y}$ were known, Doob's h-transform (Section 2.2) would condition the reverse process on $\boldsymbol{x}_0 = \boldsymbol{y}$. Define $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}} = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_0 = \boldsymbol{y} \mid \boldsymbol{x}_t)$. The resulting reverse SDE is

$$\mathrm{d}\boldsymbol{x} = \left[\mathbf{f}(\boldsymbol{x}_t, t) - g^2(t)\left(\nabla_{x_t} \log p_t(\boldsymbol{x}_t) + \boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}\right)\right] \mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}}, \tag{6}$$

where time runs from T to 0. With the exact conditional score, this process is a bridge to $\boldsymbol{y}$, as illustrated in Figure 2(a). Its probability-flow ODE is

$$\mathrm{d}\boldsymbol{x} = \left[\mathbf{f}(\boldsymbol{x}_t, t) - \frac{1}{2} g^2(t)\left(\nabla_{x_t} \log p_t(\boldsymbol{x}_t) + \boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}\right)\right] \mathrm{d}t. \tag{7}$$

The SDE and ODE share conditional marginals when initialized from the same conditional distribution. However, the required drift adjustment $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}$ depends on the unknown target $\boldsymbol{y}$, making this ideal construction unavailable for coarse-guided generation.

4 Approximating the Intractable $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}$

As $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}} = \nabla_{x_t} \log p_t(\boldsymbol{x}_0 = \boldsymbol{y} | \boldsymbol{x}_t)$ is intractable, we manage to convert it into a tractable and appropriate approximation. To this end, we propose to approximate it with the following:

$$\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}} = \nabla_{x_t} \log p_t(\boldsymbol{x}_0 = \boldsymbol{y} | \boldsymbol{x}_t) \approx \nabla_{x_t} \log p_t(\boldsymbol{x}_0 = \tilde{\boldsymbol{y}} | \boldsymbol{x}_t). \tag{8}$$

We name $\nabla_{x_t} \log p_t(\boldsymbol{x}_0 = \tilde{\boldsymbol{y}} | \boldsymbol{x}_t)$ as $\boldsymbol{h}_{\boldsymbol{x}_0=\tilde{\boldsymbol{y}}}$, which is another h function (*c.f.*, Figure 2(b)). Then, we use Bayes' rule to change $\boldsymbol{h}_{\boldsymbol{x}_0=\tilde{\boldsymbol{y}}}$ into:

$$\boldsymbol{h}_{\boldsymbol{x}_0=\tilde{\boldsymbol{y}}} = \nabla_{x_t} \log p_t(\boldsymbol{x}_0 = \tilde{\boldsymbol{y}} | \boldsymbol{x}_t) = \nabla_{x_t} \log p_t(\boldsymbol{x}_t | \boldsymbol{x}_0 = \tilde{\boldsymbol{y}}) - \nabla_{x_t} \log p_t(\boldsymbol{x}_t), \tag{9}$$

where the second term $\nabla_{x_t} \log p_t(\boldsymbol{x}_t)$ can be replaced by the given trained score predictor s_θ. Besides, we already known the conditional distribution $p_t(\boldsymbol{x}_t | \boldsymbol{x}_0)$:

$$p_t(\boldsymbol{x}_t | \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \alpha_t \boldsymbol{x}_0, {\sigma_t}^2 \mathbf{I}), \tag{10}$$

where α_t and σ_t (noise level) are determined by $\mathbf{f}(\boldsymbol{x}_t, t)$, $g^2(t)$ of Eq. (1). When $\boldsymbol{x}_0 = \widetilde{\boldsymbol{y}}$, which is also given as the coarse sample, $p_t(\boldsymbol{x}_t|\boldsymbol{x}_0 = \widetilde{\boldsymbol{y}})$ is closed-form as $\mathcal{N}(\boldsymbol{x}_t; \alpha_t\widetilde{\boldsymbol{y}}, {\sigma_t}^2\mathbf{I})$ and $\nabla_x \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0 = \widetilde{\boldsymbol{y}})$ is tractable:

$$\begin{aligned}\nabla_x \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0 = \widetilde{\boldsymbol{y}}) &= \nabla_{\boldsymbol{x}_t}\left(-\frac{1}{2\sigma_t^2}(\boldsymbol{x}_t - \alpha_t\widetilde{\boldsymbol{y}})^{\mathbf{T}}(\boldsymbol{x}_t - \alpha_t\widetilde{\boldsymbol{y}})\right)\\ &= \frac{1}{\sigma_t^2}(\alpha_t\widetilde{\boldsymbol{y}} - \boldsymbol{x}_t).\end{aligned} \tag{11}$$

After that, we convert the intractable $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}$ into the following tractable approximation:

$$\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}} \approx \boldsymbol{h}_{\boldsymbol{x}_0=\widetilde{\boldsymbol{y}}} = \frac{1}{\sigma_t^2}(\alpha_t\widetilde{\boldsymbol{y}} - \boldsymbol{x}_t) - \nabla_{x_t}\log p_t(\boldsymbol{x}_t). \tag{12}$$

4.1 Approximation Error Analysis

Our proposed approximation in Eq. (8) inevitably introduces approximation errors. We can calculate the Euclidean distance between $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}$ and $\boldsymbol{h}_{\boldsymbol{x}_0=\widetilde{\boldsymbol{y}}}$ as the approximation error $\mathcal{J}$:

$$\begin{aligned}\mathcal{J} &= ||\nabla_x \log p_t(\boldsymbol{x}_0 = \boldsymbol{y}|\boldsymbol{x}_t) - \nabla_x \log p_t(\boldsymbol{x}_0 = \widetilde{\boldsymbol{y}}|\boldsymbol{x}_t)||_2\\ &= ||\nabla_x \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0 = \widetilde{\boldsymbol{y}}) - \nabla_x \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0 = \boldsymbol{y})||_2 = ||\frac{\alpha_t}{\sigma_t^2}(\widetilde{\boldsymbol{y}} - \boldsymbol{y})||_2.\end{aligned} \tag{13}$$

Taking the variance-preserving (VP) diffusion model as the example ($\alpha_t^2 + \sigma_t^2 = 1$), we can get:

$$\mathcal{J} = ||\frac{\sqrt{(1-\sigma_t^2)}}{\sigma_t^2}(\widetilde{\boldsymbol{y}} - \boldsymbol{y})||_2. \tag{14}$$

We can see that the first factor of approximation error $\mathcal{J}$ is negatively correlated with the *noise level* σ_t. Meanwhile, when $\sigma_t \to 0$, $\mathcal{J} \to \infty$ and when $\sigma_t \to 1$, $\mathcal{J} \to 0$. Besides, the error is also proportional to the second factor $\|\widetilde{\boldsymbol{y}} - \boldsymbol{y}\|_2$, *i.e.*, the *endpoint gap*.

4.2 Adaptive Weighted Approximation for Error Control

We weight the approximate adjustment by a scalar $\lambda_t \in [0, 1]$, giving the guided score $\mathrm{s}_\theta(\boldsymbol{x}_t, t) + \lambda_t\boldsymbol{h}_{\boldsymbol{x}_0=\widetilde{\boldsymbol{y}}}$. In the ODE notation of Eq. (7), this yields the controlled sampling dynamics:

$$\mathrm{d}\boldsymbol{x} = \left[\mathbf{f}(\boldsymbol{x}_t, t) - \frac{1}{2}g^2(t)\left(\mathrm{s}_\theta(\boldsymbol{x}_t, t) + \lambda_t\left(\frac{\alpha_t\widetilde{\boldsymbol{y}} - \boldsymbol{x}_t}{\sigma_t^2} - \mathrm{s}_\theta(\boldsymbol{x}_t, t)\right)\right)\right]\mathrm{d}t. \tag{15}$$

Noise-level-aware initialization. The noise-dependent amplification in Eq. (13) motivates a smooth weight function $\lambda_{\sigma,t} \in [0, 1]$ of the noise level σ_t. We typically parameterize this schedule as a power function of σ_t. As the noise level decreases and the approximation error grows, the weight gradually decreases, with $\lambda_{\sigma,t} \to 1$ as $\sigma_t \to 1$ and $\lambda_{\sigma,t} \to 0$ as $\sigma_t \to 0$. This retains coarse guidance when the approximation is reliable and reduces its influence as the error is amplified.

Consistency for endpoint gap. To account for the unknown endpoint gap, we use prediction consistency as a model-based proxy and define the consistency residual:

$$\boldsymbol{r}_t(\lambda) = \widehat{\boldsymbol{x}}_{0|t^-}^{(\lambda)} - \widehat{\boldsymbol{x}}_{0|t}^{(\lambda)}. \tag{16}$$

where t^- denotes the next sampling time. The current guided clean prediction is

$$\widehat{\boldsymbol{x}}_{0|t}^{(\lambda)} = (1-\lambda)\widehat{\boldsymbol{x}}_{0|t}^{(0)} + \lambda\widetilde{\boldsymbol{y}}, \qquad \widehat{\boldsymbol{x}}_{0|t}^{(0)} = \frac{\boldsymbol{x}_t + \sigma_t^2\mathrm{s}_\theta(\boldsymbol{x}_t, t)}{\alpha_t}. \tag{17}$$

Meanwhile, $\widehat{\boldsymbol{x}}_{0|t^-}^{(\lambda)}$ is the model's clean prediction after taking one sampling step with weight λ, without additional coarse guidance. A smaller residual norm indicates stronger agreement between the guided endpoint and the model's next-step prediction. On top of that, we evaluate two weights: the initialized λ_σ and a nearby $\lambda_\sigma^{'} \in [0, 1]$, perturbed toward the interval interior. Let $\boldsymbol{r}_\sigma, \boldsymbol{r}_\sigma^{'}$ denote their residual vectors in Eq. (16). A local linear approximation of the residual gives the damped least-squares update (Marquardt, 1963):

$$\Delta\lambda = -\frac{(\lambda_\sigma^{'} - \lambda_\sigma)\langle\boldsymbol{r}_\sigma^{'} - \boldsymbol{r}_\sigma, \boldsymbol{r}_\sigma\rangle}{(1+\kappa)\|\boldsymbol{r}_\sigma^{'} - \boldsymbol{r}_\sigma\|_2^2}, \quad \lambda_t = (\lambda_\sigma + \Delta\lambda)\,. \tag{18}$$

Here, $\kappa \geq 0$ controls damping. The resulting sampler combines noise-aware initialization with consistency-based weight refinement, as summarized in Algorithm 1.

Algorithm 1 Adaptive Weighted h-Transform Sampling

Require: Coarse sample $\widetilde{\boldsymbol{y}}$, score model s_θ, weight schedule λ_σ, damping κ.

1: Initialize $\boldsymbol{x}_T \sim p_T$.
2: **for** each sampling time t **do**
3: Set $\lambda_t = \lambda_\sigma$ and choose a nearby $\lambda'_\sigma \in [0, 1]$.
4: $\boldsymbol{r}_\sigma = \widehat{\boldsymbol{x}}_{0|t^-}^{(\lambda_\sigma)} - \widehat{\boldsymbol{x}}_{0|t}^{(\lambda_\sigma)}, \quad \boldsymbol{r}'_\sigma = \widehat{\boldsymbol{x}}_{0|t^-}^{(\lambda'_\sigma)} - \widehat{\boldsymbol{x}}_{0|t}^{(\lambda'_\sigma)}$.
5: $\lambda_t = \lambda_\sigma - \dfrac{(\lambda'_\sigma - \lambda_\sigma)\langle \boldsymbol{r}'_\sigma - \boldsymbol{r}_\sigma, \boldsymbol{r}_\sigma \rangle}{(1+\kappa)\|\boldsymbol{r}'_\sigma - \boldsymbol{r}_\sigma\|_2^2}$.
6: $\boldsymbol{x}_{t-\Delta_t} = \boldsymbol{x}_t - \left[\mathbf{f}(\boldsymbol{x}_t, t) - \dfrac{1}{2} g^2(t) \left(\mathrm{s}_\theta(\boldsymbol{x}_t, t) + \lambda_t \left(\dfrac{\alpha_t \widetilde{\boldsymbol{y}} - \boldsymbol{x}_t}{\sigma_t^2} - \mathrm{s}_\theta(\boldsymbol{x}_t, t) \right) \right) \right] \Delta_t$.
7: **end for**

5 Related Work

Diffusion Models and Bridges. Diffusion models have achieved state-of-the-art visual modeling capability. Based on these, various large-scale pre-trained diffusion models, such as Flux (Labs, 2024; Labs et al., 2025) and Wan (Wan et al., 2025), have emerged and demonstrated remarkable generative performance. Besides noise-to-visual generation, some works (Zhou et al., 2023; Su et al., 2022; De Bortoli et al., 2021; Zheng et al., 2024) propose diffusion bridges to achieve the visual-to-visual translations. The typical paradigm is leveraging Doob's h-transform (Doob & Doob, 1984; Rogers & Williams, 2000) to link the paired samples and learn a joint distribution to transport between them. Our paper is inspired by the h-transform to achieve guided visual generation.

Conditional Visual Generation. Conditional generation (Zhan et al., 2024; Batzolis et al., 2021; Ho & Salimans, 2022) aims to handle various modality conditions (*e.g.*, text, action, and visual samples) as guidance to constrain the generation. Beyond these common conditions, current research on conditional generation mainly focuses on visual condition signals (*e.g.*, coarse sample, stroke, and pose map). Based on different types, the task can be specified as image restoration (Wang et al., 2020; Liang et al., 2021; Ren et al., 2023), inverse problem (Chung et al., 2022a; Kim et al., 2025b; Tarantola, 2005; Choi et al., 2021), and conditional control (Zhang et al., 2023; Shen & Tang, 2024), and so on. Meanwhile, the solutions also contain training-based (Wang et al., 2022; Murez et al., 2018; Parmar et al., 2024) and inference-stage guidance (Yu et al., 2023; Du et al., 2024; Gao et al., 2025; Bu et al., 2026). Our paper focuses on conditional generation w/ coarse samples as guidance and generates corresponding high-quality fine samples in a training-free manner.

6 Experiments

6.1 Coarse-Image Guided Generation

We first evaluated on image restoration tasks that aim to recover high-quality fine images from degraded samples. Specifically, this task entails recovering a clean image $\boldsymbol{y}$ given a coarse image $\widetilde{\boldsymbol{y}}$.

Benchmark. Following DPS (Chung et al., 2022a), we use the validation set of FFHQ 256x256 (Karras et al., 2019) to serve as the evaluation dataset. This dataset contains 1000 clean images of human faces in $256 * 256$ resolution. For the evaluation metrics, we adopt Frechet Inception Distance (Heusel et al., 2017) (*FID*) and Learned Perceptual Image Patch Similarity (Zhang et al., 2018) (*LPIPS*) to quantify the effectiveness. *FID* can indicate the distribution similarity between synthetic images and ground-truth images, while *LPIPS* focuses on image-level similarity.

Implementations. Following the protocol in DPS (Chung et al., 2022a), we transform all clean images $\boldsymbol{y}$ by using a forward operator $\mathcal{A}$ and additive noise $\mathbf{n}$ into coarse images $\widetilde{\boldsymbol{y}}$, *i.e.*, $\widetilde{\boldsymbol{y}} = \mathcal{A}(\boldsymbol{y}) + \mathbf{n}$. Depending on the forward operator type, it can be divided into four image restoration tasks: super-resolution (*SR*), inpainting (*Inpaint*), motion deblur (*MD*), and Gaussian deblur (*GD*). These coarse images $\widetilde{\boldsymbol{y}}$ serve as guidance in the generation process. For fairness, we follow previous work (Chung et al., 2022a) and use the same pre-trained diffusion model from (Dhariwal & Nichol, 2021) and maintain other settings the same. During sampling, we set the initialized weight $\lambda_t = \sigma^5$.

Table 1: Quantitative results of coarse-image guided generation on the FFHQ 256x256 validation dataset. **Bold**: best, underline: second best.

Method	Known Operator	SR		Inpaint		GD		MD	
		FID$_\downarrow$	LPIPS$_\downarrow$	FID$_\downarrow$	LPIPS$_\downarrow$	FID$_\downarrow$	LPIPS$_\downarrow$	FID$_\downarrow$	LPIPS$_\downarrow$
ADMM-TV	✓	110.6	0.428	181.5	0.463	186.7	0.507	152.3	0.508
ILVR (Choi et al., 2021; Song et al., 2020)	✓	96.72	0.563	76.54	0.612	109.0	0.403	292.2	0.657
PnP-ADMM (Chan et al., 2016)	✓	66.52	0.353	123.6	0.692	90.42	0.441	89.08	0.405
MCG (Chung et al., 2022b)	✓	87.64	0.520	29.26	0.286	101.2	0.340	310.5	0.702
DDRM (Kawar et al., 2022)	✓	62.15	0.294	69.71	0.587	74.92	0.332	-	-
DPS (Chung et al., 2022a)	✓	39.35	0.214	**21.19**	**0.212**	44.05	0.257	39.92	0.242
SDEdit (Meng et al., 2021)		33.73	0.274	52.77	0.404	**35.30**	0.296	**34.41**	0.289
BlindDPS (Chihaoui et al., 2024)		-	-	-	-	39.92	**0.216**	38.89	**0.204**
HiFlow (Bu et al., 2026)		33.48	0.225	-	-	-	-	-	-
Ours		**33.31**	**0.213**	48.55	0.334	38.17	0.253	35.23	0.237

Baselines. We compared our method against: 1) six inverse-problem solutions (*c.f.*, Fig. 1(b)): ADMM-TV, ILVR (Choi et al., 2021; Song et al., 2020), PnP-ADMM (Chan et al., 2016), MCG (Chung et al., 2022b), DDRM (Kawar et al., 2022), and DPS (Chung et al., 2022a). These solutions require additional prior knowledge of the forward operator. We report their results from (Chung et al., 2022a). 2) The represented start-guided solution (*c.f.*, Fig. 1(c)): SDEdit (Meng et al., 2021), which does not require knowing the forward operator. We set its $t_0 = 500$. 3) BlindDPS (Chihaoui et al., 2024), which relies on a trained blur kernel (3M steps) to mitigate the unknown operator problem. 4) HiFlow (Bu et al., 2026), an image super-resolution method.

Results. While the first 6 baselines and BlindDPS require known forward operators or trained kernels, and HiFlow is designed for image super-resolution, our method generalizes across diverse tasks w/o needing task-specific operators or kernels. Despite its broad applicability, our approach remains highly competitive, ranking in the top two for $6/8$ metrics. This verifies that our method provides a robust mechanism for coarse-image guidance, leading to structural preservation and fidelity.

6.2 Coarse-Video Guided Generation

To further verify our effectiveness and generalization, we evaluated it on a camera-controlled video generation task. The camera-controlled video generation task is: given the first frame, corresponding camera intrinsics, and the pose, we aim to generate a video that follows a sequence of prescribed camera motions. To achieve this, we first use pretrained 3-D models to render a coarse video from the first frame and the prescribed camera motions. This coarse video is low-quality (warped) but can capture camera motion. Then it will serve as the guidance for the subsequent video generation.

Dataset. Following previous camera-controlled generation work, we use DL3DV-10K (Ling et al., 2024) as our dataset for constructing coarse videos and evaluation. It contains annotated samples, each comprising a video with camera intrinsics and poses of all frames. We randomly select 160 samples, retain the first 49 frames, and resize them to $480p$. We use this subset as our dataset.

Implementations. We use DepthPro (Bochkovskii et al., 2024) to extract the depth map of the first frame. Then project it into a 3-D point cloud. For the given sequence of camera motion poses, we render a sequence of corresponding images and concatenate them into the coarse video. The coarse video contains two parts. 1) Valid part: pixels that exist in the given first frame. 2) Invalid part: Pixels that do not exist in the given first frame. For the invalid part, the pixels are filled by nearest-neighbor assignment. All details about obtaining the coarse video are in the Appendix. During sampling, we only use the noise-level-aware initialization and drop the consistency adjustment for efficiency in video generation tasks, *i.e.*, we set the weight function $\lambda_t = \sigma^\alpha$, and give these two parts different α. For the valid part, $\alpha = 4$, and for the invalid part, $\alpha = 8$. For all hyperparameters, we followed the default settings of previous work (Singer et al., 2025; Yang et al., 2024b).

Baselines. We compared our method with 1) Training-based GWTF (Burgert et al., 2025): learning a noise warping based on optical flows to model the motion. We used the official trained model and set its key hyperparameter degradation value γ to 0.5 and 0.7 as recommended in the original paper. 2) Training-free TTM (Singer et al., 2025): leverages start-guided synthesis for two decoupled parts. This method also depends on a coarse video. For fairness, we use the same warped coarse as ours. We also took $t_{weak} = 4$ (t_w), $t_{strong} = 8$ (t_s), and $t_{weak} = 4$, $t_{strong} = 9$ according to the default and recommendations in their paper and code.

Table 2: Quantitative results of camera-controlled video generation on DL3DV.

Method	MSE↓	LPIPS↓	FVD↓	DINOv2↓	CLIP Cons.↑	Optical Flow↓
GT	-	-	-	-	0.974	-
Coarse Video	11.46	0.276	15.55	0.220	0.973	41.3
GWTF($\gamma = 0.5$)	26.08	0.360	15.31	0.149	0.975	118.5
GWTF($\gamma = 0.7$)	36.45	0.457	21.25	0.173	**0.984**	145.2
TTM($t_w = 4, t_s = 8$)	23.50	0.382	15.69	0.147	0.980	157.2
TTM($t_w = 4, t_s = 9$)	23.15	0.380	15.59	0.147	0.980	158.8
Ours	**11.45**	**0.272**	**13.26**	**0.143**	0.972	**38.7**



Figure 3: **Qualitative comparisons on the subset of DL3DV-10K.** Our method shows better appearance alignment to the ground truth (see highlighted blue boxes).

Metrics. We evaluated the quality of generated videos at the frame level using Mean Squared Error (*MSE*) and *LPIPS* (Zhang et al., 2018) relative to the ground-truth videos. Additionally, we calculated the distributional difference between the ground-truth videos and the synthetic videos using Frechet Video Distance (Unterthiner et al., 2018) (*FVD*). We also calculated the average semantic similarity of each frame between ground-truth videos and the synthetic videos with DINOv2 Distance (Oquab et al., 2023) (*DINOv2*). We used the CLIP similarity (Radford et al., 2021) between two consistent frames (*CLIP Cons.*) to evaluate the temporal consistency. For camera motion consistency, we calculated the Mean Squared Error between the RAFT-estimated optical flows (Teed & Deng, 2020) of the ground-truth videos and those of the synthetic videos (*Optical Flow*).

Results. As shown in Table 2, our method achieves the best performance across all metrics except *CLIP Cons.*. This indicates that our synthetic videos are closest to ground-truth videos and have the best motion consistency. For the *CLIP Cons.*, although our method has the lowest 0.972, we should note that the ground truth only achieves 0.974 on this metric. This indicates that other methods even outperform the ground truth on this metric by sacrificing motion consistency to get quite similar consecutive frames. Additionally, we provide qualitative comparisons in Figure 3.

6.3 Ablations of the Weight Function λ

λ	SR	Inpaint	GD	MD
0.0	0.586	0.586	0.586	0.586
0.5	0.456	0.989	0.445	0.491
1.0	0.472	1.008	0.483	0.546
2.0	0.466	1.009	0.486	0.552
$\sin(\frac{\pi t}{2T})$	0.224	0.448	0.293	0.344
$\frac{t}{T}$	0.223	0.406	0.274	0.344
$(\frac{t}{T})^3$	0.380	0.369	0.393	0.428
$(\frac{t}{T})^5$	0.466	0.461	0.472	0.480
σ	0.268	0.752	0.359	0.412
σ^3	**0.208**	0.433	0.257	0.340
σ^5	0.229	0.341	0.260	0.247
ours (σ^5)	0.213	**0.334**	**0.253**	**0.237**

Table 3: The ablation of weight schedule λ.

Table 3 compares constant, sinusoidal, timestep-based, and noise-level-aware designs of λ across four image restoration tasks using *LPIPS* (lower is better). Among the non-adaptive designs, schedules based on powers of σ consistently yield the best performance. This agrees with the error analysis in Section 4.1: the approximation error increases as the noise level decreases, motivating a schedule that progressively reduces the influence of the coarse approximation. These results distinguish our principled, noise-level-aware weighting from naive score blending and highlight its role in achieving high-quality guided synthesis. Adding the consistency-based correction further improves performance and yields the best overall results, demonstrating the effectiveness of our error control guided by both factors during sampling.

6.4 Additional Visual Generation Tasks

Method	Source Consistency					Semantic Alignment	
	Distance$_\downarrow$	PSNR$_\uparrow$	LPIPS$_\downarrow$	MSE$_\downarrow$	SSIM$_\uparrow$	CLIP Entire$_\uparrow$	CLIP Edited$_\uparrow$
ODE-Inv	0.074	17.57	0.240	0.024	0.691	24.57	21.73
SDEdit	0.036	22.57	0.119	0.008	0.747	24.56	21.95
iRFDS	0.069	18.81	0.191	0.021	0.738	25.12	21.95
FlowEdit	0.036	23.02	0.082	0.007	0.842	**25.98**	**22.81**
FlowAlign	0.028	25.50	**0.053**	**0.004**	**0.879**	25.28	22.00
Ours	**0.017**	**25.98**	0.097	**0.004**	0.819	25.41	22.37

Table 4: Quantitative comparisons with five baselines for image editing.

Image Editing. To further explore the generalization ability of our method, we conducted experiments on the image editing task, *i.e.*, given a source image, a source prompt, and a target prompt, we edit the source image into a target image. Specifically, our method uses the source image as the $\widetilde{\boldsymbol{y}}$, and s_θ is the predicted score by a pretrained text-to-image models that adopt the target prompt as the condition. We set the sampling timestep to 50, the target guidance scale to 13.5, and $\lambda_t = \sigma_t$. We compared our method with five state-of-the-art flow-based editing methods that were implemented with SD-3-medium (Esser et al., 2024): *ODE inversion*, *SDEdit* (Meng et al., 2021), *iRFDS* (Yang et al., 2024a), *FlowEdit* (Kulikov et al., 2024) and *FlowAlign* (Kim et al., 2025a). The results for these baselines are from *FlowAlign* on PIE-Bench (Ju et al., 2023) and we report the seven common metrics to evaluate both source consistency (*Distance*, *PSNR*, *LPIPS*, *MSE*, *SSIM*) and Semantic Alignment (*Entire CLIP* and *Edited CLIP Score*). The results are shown in Tab. 4.



Figure 4: Generalize to video motion transfer.

Video Motion Transfer. We further apply our method to video motion transfer, translating the temporal dynamics of a source video into a new semantic context. By using the source video (*e.g.*, a car driving in a field) as the coarse guidance $\widetilde{\boldsymbol{y}}$ and a target text prompt (*e.g.*, "a tiger walking in a forest") to dictate the semantic score s_θ, our framework seamlessly integrates both signals. As illustrated in Figure 4, the generated video successfully preserves the original motion trajectories while synthesizing high-fidelity target semantics, demonstrating the potential applicability of our method across diverse modalities.

7 Conclusion

We proposed **Adaptive Weighted h-Transform Sampling** for coarse-guided visual generation. We inject the guidance during the sampling process by designing an h function that can approximately drive the generation towards the underlying ideal sample. We further design a weight scheduler to restrict the approximation error for better synthetic results. Extensive ablations show the generalization and effectiveness. In the future, probability transition modification mechanisms (represented by h-transform) can be further explored in more general conditional generation applications.

AI Use Statement. Generative AI tools were used to polish the manuscript and improve the clarity and readability of the writing. We have not used generative AI tools for research ideation, data collection, mathematical proofs, experimental implementation, or writing the scientific claims of this paper. All AI-assisted revisions were thoroughly reviewed and verified by the authors. We take full responsibility for the final content of this work, including all text, claims, and artifacts.

REFERENCES

Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021.

Aleksei Bochkovskii, AmaãĢl Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024.

Jiazi Bu, Pengyang Ling, Yujie Zhou, Pan Zhang, Tong Wu, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Hiflow: Training-free high-resolution image generation with flow-aligned guidance. *Advances in Neural Information Processing Systems*, 38:144317–144351, 2026.

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. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13–23, 2025.

Stanley H Chan, Xiran Wang, and Omar A Elgendy. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1): 84–98, 2016.

Hamadi Chihaoui, Abdelhak Lemkhenter, and Paolo Favaro. Blind image restoration via fast diffusion inversion. *Advances in Neural Information Processing Systems*, 37:34513–34532, 2024.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021.

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*, 2022a.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022b.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in neural information processing systems*, 34:17695–17709, 2021.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Joseph L Doob and JI Doob. *Classical potential theory and its probabilistic counterpart*, volume 262. Springer, 1984.

Ruoyi Du, Dongliang Chang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Demofusion: Democratising high-resolution image generation with no $$$. In *CVPR*, 2024.

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.

Zheng Gao, Jifei Song, Zhensong Zhang, Jiankang Deng, and Ioannis Patras. Frequency-guided diffusion for training-free text-driven image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19195–19205, 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

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

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

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.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Pnp inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations*, 2023.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in neural information processing systems*, 35:23593–23606, 2022.

Jeongsol Kim, Yeobin Hong, and Jong Chul Ye. Flowalign: Trajectory-regularized, inversion-free flow-based image editing. *arXiv preprint arXiv:2505.23145*, 2025a.

Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. Flowdps: Flow-driven posterior sampling for inverse problems. *arXiv preprint arXiv:2503.08136*, 2025b.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. *arXiv preprint arXiv:2412.08629*, 2024.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1833–1844, 2021.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

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

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*, 2022.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

Dimitra Maoutsa, Sebastian Reich, and Manfred Opper. Interacting particle solutions of fokker–planck equations through gradient–log–density estimation. *Entropy*, 22(8):802, 2020.

Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

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

Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4500–4509, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Gaurav Parmar, Taesung Park, Srinivasa Narasimhan, and Jun-Yan Zhu. One-step image translation with text-to-image models. *arXiv preprint arXiv:2403.12036*, 2024.

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.

Mengwei Ren, Mauricio Delbracio, Hossein Talebi, Guido Gerig, and Peyman Milanfar. Multiscale structure guided diffusion for image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10721–10733, 2023.

L Chris G Rogers and David Williams. *Diffusions, Markov processes, and martingales*, volume 2. Cambridge university press, 2000.

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.

Fei Shen and Jinhui Tang. Imagpose: A unified conditional framework for pose-guided person generation. *Advances in neural information processing systems*, 37:6246–6266, 2024.

Assaf Singer, Noam Rotstein, Amir Mann, Ron Kimmel, and Or Litany. Time-to-move: Training-free motion controlled video generation via dual-clock denoising. *arXiv preprint arXiv:2511.08633*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. *arXiv preprint arXiv:2203.08382*, 2022.

Albert Tarantola. *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022.

Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3365–3387, 2020.

Xiaofeng Yang, Cheng Chen, Xulei Yang, Fayao Liu, and Guosheng Lin. Text-to-image rectified flow as plug-and-play priors. *arXiv preprint arXiv:2406.03293*, 2024a.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024b.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23174–23184, 2023.

Zheyuan Zhan, Defang Chen, Jian-Ping Mei, Zhenghe Zhao, Jiawei Chen, Chun Chen, Siwei Lyu, and Can Wang. Conditional image synthesis with diffusion models: A survey. *arXiv preprint arXiv:2409.19365*, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Kaiwen Zheng, Guande He, Jianfei Chen, Fan Bao, and Jun Zhu. Diffusion bridge implicit models. *arXiv preprint arXiv:2405.15885*, 2024.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. *arXiv preprint arXiv:2309.16948*, 2023.

Appendix

This appendix is organized as follows:

- Section A.1 gives the proof of equivalent marginal distributions between the reversed-SDE and its corresponding PF-ODE.
- Section A.2 demonstrates the detailed derivation of $\boldsymbol{h}_{\boldsymbol{x}_0=\widetilde{\boldsymbol{y}}}$.
- Section A.3 provides the Weighted h-Transform Inference equation for velocity-based Optimal Transport Flow Matching (OT-FM) and noise-based Variance-Preserving SDE (VP-SDE).
- Section A.4 indicates the implementation details of our method, including how we obtain the coarse videos, text prompts of video generation, and how we set related hyperparameters.
- Section A.5 provides additional qualitative results.

A.1 Equivalence Proof of Marginal Distributions

In this section, we provide the formal proof that the stochastic differential equation (SDE) in Eq. (6) and the ordinary differential equation (ODE) in Eq. (7) share the same marginal distribution.

Let $q_t(\boldsymbol{x}_t)$ denote the marginal probability density of the latent $\boldsymbol{x}_t$ at time t. We first write the drift term of the SDE (Eq. (6)) as:

$$\boldsymbol{\mu}_{\text{SDE}}(\boldsymbol{x}_t, t) = \mathbf{f}(\boldsymbol{x}_t, t) - g^2(t)\nabla_{x_t} \log p_t(\boldsymbol{x}_t) - g^2(t)\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}} \tag{A19}$$

The evolution of the probability density $q_t(\boldsymbol{x}_t)$ for this reverse-time SDE is governed by the Fokker-Planck equation (Maoutsa et al., 2020) (F-P equation). Taking into account the negative time increment in the reverse process, the corresponding F-P equation is given by:

$$\frac{\partial q_t}{\partial t} = -\nabla \cdot (\boldsymbol{\mu}_{\text{SDE}} q_t) - \frac{1}{2} g^2(t) \Delta q_t \tag{A20}$$

Then, we substitute the mathematical identity $\Delta q_t = \nabla \cdot (\nabla q_t) = \nabla \cdot (q_t \nabla \log q_t)$ into Eq. (A20) yields:

$$\frac{\partial q_t}{\partial t} = -\nabla \cdot (\boldsymbol{\mu}_{\text{SDE}} q_t) - \frac{1}{2} g^2(t) \nabla \cdot (q_t \nabla \log q_t) \tag{A21}$$

We can reformulate the equation as a standard continuity equation:

$$\frac{\partial q_t}{\partial t} = -\nabla \cdot \left(\underbrace{\left[\boldsymbol{\mu}_{\text{SDE}} + \frac{1}{2} g^2(t) \nabla \log q_t \right]}_{\boldsymbol{v}_{\text{ODE}}} q_t \right) \tag{A22}$$

Eq. (A22) takes the form $\frac{\partial q_t}{\partial t} = -\nabla \cdot (\boldsymbol{v}_{\text{ODE}} q_t)$, which precisely describes the density evolution of a deterministic ODE $\mathrm{d}\boldsymbol{x} = \boldsymbol{v}_{\text{ODE}} \mathrm{d}t$. Therefore, the equivalent deterministic drift $\boldsymbol{v}_{\text{ODE}}$ that shares the same marginal density q_t is:

$$\boldsymbol{v}_{\text{ODE}} = \boldsymbol{\mu}_{\text{SDE}} + \frac{1}{2} g^2(t) \nabla \log q_t(\boldsymbol{x}_t) \tag{A23}$$

In the context of diffusion models, the goal of introducing the additional term $\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}}$ is to shift the unconditional score to the target conditional score. By definition, the exact marginal score of the process is structurally related to the unconditional score via:

$$\nabla \log q_t(\boldsymbol{x}_t) = \nabla_{x_t} \log p_t(\boldsymbol{x}_t) + \boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}} \tag{A24}$$

Finally, we substitute the definition of $\boldsymbol{\mu}_{\text{SDE}}$ and the score identity from Eq. (A24) into Eq. (A23):

$$\begin{aligned} \boldsymbol{v}_{\text{ODE}} = &\left[\mathbf{f}(\boldsymbol{x}_t, t) - g^2(t)\nabla_{x_t} \log p_t(\boldsymbol{x}_t) - g^2(t)\boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}} \right] \\ &+ \frac{1}{2} g^2(t) \left(\nabla_{x_t} \log p_t(\boldsymbol{x}_t) + \boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}} \right) \end{aligned} \tag{A25}$$

Simplifying the terms, we obtain:

$$\boldsymbol{v}_{\text{ODE}} = \mathbf{f}(\boldsymbol{x}_t, t) - \frac{1}{2} g^2(t) \left(\nabla_{x_t} \log p_t(\boldsymbol{x}_t) + \boldsymbol{h}_{\boldsymbol{x}_0=\boldsymbol{y}} \right) \tag{A26}$$

This derived drift $\boldsymbol{v}_{\text{ODE}}$ is identical to the drift of the ODE presented in Eq. (7). Since the same continuity equation governs the probability density of the given SDE and this resulting ODE, we conclude that they exhibit the same marginal distribution $q_t(\boldsymbol{x}_t)$ at all times t.

A.2 DERIVATION OF $\boldsymbol{h}_{\boldsymbol{x}_0=\widetilde{\boldsymbol{y}}}$

In this section, we provide the detailed derivation of $\boldsymbol{h}_{\boldsymbol{x}_0=\widetilde{\boldsymbol{y}}}$.

Based on Eq. (10), we can get the probability density function of $p_t(\boldsymbol{x}_t|\boldsymbol{x}_0=\widetilde{\boldsymbol{y}})$:

$$p_t(\boldsymbol{x}_t|\boldsymbol{x}_0=\widetilde{\boldsymbol{y}})=\frac{1}{(2\pi\sigma_t^2)^{d/2}}\exp\left(-\frac{\|\boldsymbol{x}_t-\alpha_t\widetilde{\boldsymbol{y}}\|^2}{2\sigma_t^2}\right). \tag{A27}$$

Then take the $\log$ of both sides of the equation and calculate the gradient:

$$\begin{aligned}\nabla_{x_t}\log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0=\widetilde{\boldsymbol{y}})&=\nabla_{\boldsymbol{x}_t}\left[-\frac{d}{2}\log(2\pi\sigma_t^2)-\frac{\|\boldsymbol{x}_t-\alpha_t\widetilde{\boldsymbol{y}}\|^2}{2\sigma_t^2}\right]\\&=-\frac{1}{2\sigma_t^2}\nabla_{\boldsymbol{x}_t}\|\boldsymbol{x}_t-\alpha_t\widetilde{\boldsymbol{y}}\|^2\\&=-\frac{2(\boldsymbol{x}_t-\alpha_t\widetilde{\boldsymbol{y}})}{2\sigma_t^2}\\&=-\frac{\boldsymbol{x}_t-\alpha_t\widetilde{\boldsymbol{y}}}{\sigma_t^2}=\frac{1}{\sigma_t^2}(\alpha_t\widetilde{\boldsymbol{y}}-\boldsymbol{x}_t).\end{aligned} \tag{A28}$$

Then we substitute it into Eq. (9) and get the final $\boldsymbol{h}_{\boldsymbol{x}_0=\widetilde{\boldsymbol{y}}}$, *i.e.*, Eq. (12).

A.3 Weighted h-Transform Inference for Various SDEs

We derived our method, Weighted h-Transform Inference, from the Score Matching perspective in our paper, and we claim that our method is compatible with other diffusion models like DDPM and Flow Matching. In this section, we provide the derivation of the inference equation of v-based Optimal Transport Flow Matching (OT-FM) and ϵ-based Variance-Preserving SDE (VP-SDE).

A.3.1 OT-FM with v-prediction.

In Flow Matching, the velocity field $v_\theta(\boldsymbol{x}_t, t)$ corresponds to the PF-ODE drift. Thus, we have the relation:

$$v_\theta = \mathbf{f}(\boldsymbol{x}_t, t) - \frac{1}{2} g^2(t) \mathrm{s}_\theta \quad \Longrightarrow \quad \frac{1}{2} g^2(t) \mathrm{s}_\theta = \mathbf{f}(\boldsymbol{x}_t, t) - v_\theta \tag{A29}$$

Substituting this relation into the ODE (Eq. (15)), we rewrite the drift in terms of v_θ:

$$\begin{aligned} \mathrm{d}\mathbf{x} &= \left[v_\theta + \lambda_\sigma \left(\frac{g^2(t)}{2\sigma_t^2} (\boldsymbol{x}_t - \alpha_t \widetilde{\boldsymbol{y}}) + \frac{1}{2} g^2(t) \mathrm{s}_\theta \right) \right] \mathrm{d}t \\ &= \left[v_\theta + \lambda_\sigma \left(\frac{g^2(t)}{2\sigma_t^2} (\boldsymbol{x}_t - \alpha_t \widetilde{\boldsymbol{y}}) + \mathbf{f}(\boldsymbol{x}_t, t) - v_\theta \right) \right] \mathrm{d}t \end{aligned} \tag{A30}$$

For OT-FM, the trajectory is defined by $\sigma_t = 1 - \alpha_t$. Under a linear schedule where $\alpha_t = 1 - t$ and $\sigma_t = t$, we have $\dot{\alpha}_t = -1$ and $\dot{\sigma}_t = 1$. The corresponding drift and squared diffusion coefficients are:

$$\mathbf{f}(\boldsymbol{x}_t, t) = \frac{\dot{\alpha}_t}{\alpha_t} \boldsymbol{x}_t \tag{A31}$$

$$g^2(t) = 2\sigma_t \dot{\sigma}_t - 2\frac{\dot{\alpha}_t}{\alpha_t} \sigma_t^2 = -2\dot{\alpha}_t \sigma_t \left(1 + \frac{\sigma_t}{\alpha_t} \right) = -2\dot{\alpha}_t \frac{\sigma_t}{\alpha_t} \tag{A32}$$

We now simplify the guidance term from Eq. (A30) using these coefficients:

$$\begin{aligned} \frac{g^2(t)}{2\sigma_t^2} (\boldsymbol{x}_t - \alpha_t \widetilde{\boldsymbol{y}}) + \mathbf{f}(\boldsymbol{x}_t, t) &= \frac{-2\dot{\alpha}_t \frac{\sigma_t}{\alpha_t}}{2\sigma_t^2} (\boldsymbol{x}_t - \alpha_t \widetilde{\boldsymbol{y}}) + \frac{\dot{\alpha}_t}{\alpha_t} \boldsymbol{x}_t \\ &= -\frac{\dot{\alpha}_t}{\alpha_t \sigma_t} (\boldsymbol{x}_t - \alpha_t \widetilde{\boldsymbol{y}}) + \frac{\dot{\alpha}_t}{\alpha_t} \boldsymbol{x}_t \\ &= \boldsymbol{x}_t \dot{\alpha}_t \left(\frac{1}{\alpha_t} - \frac{1}{\alpha_t \sigma_t} \right) + \frac{\dot{\alpha}_t}{\sigma_t} \widetilde{\boldsymbol{y}} \end{aligned} \tag{A33}$$

Using the OT constraint $\sigma_t - 1 = -\alpha_t$, the term simplifies perfectly:

$$\boldsymbol{x}_t \dot{\alpha}_t \left(\frac{\sigma_t - 1}{\alpha_t \sigma_t} \right) + \frac{\dot{\alpha}_t}{\sigma_t} \widetilde{\boldsymbol{y}} = -\frac{\dot{\alpha}_t}{\sigma_t} \boldsymbol{x}_t + \frac{\dot{\alpha}_t}{\sigma_t} \widetilde{\boldsymbol{y}} = -\dot{\alpha}_t \left(\frac{\boldsymbol{x}_t - \widetilde{\boldsymbol{y}}}{\sigma_t} \right) \tag{A34}$$

Since $\dot{\alpha}_t = -1$, this term evaluates exactly to $\frac{\boldsymbol{x}_t - \widetilde{\boldsymbol{y}}}{\sigma_t}$. Substituting this back into Eq. (A30) yields the final velocity-based guided ODE:

$$\mathrm{d}\mathbf{x} = \left[v_\theta + \lambda_\sigma \left(\frac{\boldsymbol{x}_t - \widetilde{\boldsymbol{y}}}{\sigma_t} - v_\theta \right) \right] \mathrm{d}t \tag{A35}$$

A.3.2 VP-SDE with ϵ-Prediction.

For a VP forward process $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$ constrained by $\alpha_t^2 + \sigma_t^2 = 1$, the drift and diffusion coefficients are also uniquely determined by the schedule α_t, *i.e.*, Eq. (A31) and Eq. (A32).

In the ϵ-prediction framework, the score function is parameterized as $\mathrm{s}_\theta = -\frac{\epsilon_\theta}{\sigma_t}$. Substituting $\mathbf{f}$, $g^2(t)$, and s_θ into Eq. (15), we obtain:

$$\begin{aligned} \mathrm{d}\mathbf{x} &= \left[\frac{\dot{\alpha}_t}{\alpha_t}\boldsymbol{x}_t - \frac{1}{2}\left(-2\frac{\dot{\alpha}_t}{\alpha_t}\right)\left(-\frac{\epsilon_\theta}{\sigma_t} + \lambda_\sigma\left(\frac{\alpha_t\widetilde{\boldsymbol{y}} - \boldsymbol{x}_t}{\sigma_t^2} + \frac{\epsilon_\theta}{\sigma_t}\right)\right)\right]\mathrm{d}t \\ &= \frac{\dot{\alpha}_t}{\alpha_t}\left[\boldsymbol{x}_t - \frac{1}{\sigma_t}\left(\epsilon_\theta - \lambda_\sigma\left(\frac{\alpha_t\widetilde{\boldsymbol{y}} - \boldsymbol{x}_t}{\sigma_t} + \epsilon_\theta\right)\right)\right]\mathrm{d}t \end{aligned} \tag{A36}$$

We define the pseudo-target noise corresponding to the condition $\widetilde{\boldsymbol{y}}$ as:

$$\widetilde{\boldsymbol{\epsilon}} = \frac{\boldsymbol{x}_t - \alpha_t\widetilde{\boldsymbol{y}}}{\sigma_t} \tag{A37}$$

By observing that $\frac{\alpha_t\widetilde{\boldsymbol{y}} - \boldsymbol{x}_t}{\sigma_t} = -\widetilde{\boldsymbol{\epsilon}}$, we can substitute this into the inner guidance term in Eq. (A36):

$$\epsilon_\theta - \lambda_\sigma\left(-\widetilde{\boldsymbol{\epsilon}} + \epsilon_\theta\right) = \epsilon_\theta + \lambda_\sigma\left(\widetilde{\boldsymbol{\epsilon}} - \epsilon_\theta\right) \tag{A38}$$

This reveals the effective guided noise prediction $\hat{\epsilon}_\theta$:

$$\hat{\epsilon}_\theta = \epsilon_\theta + \lambda_\sigma\left(\widetilde{\boldsymbol{\epsilon}} - \epsilon_\theta\right) \tag{A39}$$

Consequently, the guided PF-ODE simplifies elegantly to the standard unguided formulation, driven entirely by the linearly interpolated noise $\hat{\epsilon}_\theta$:

$$\mathrm{d}\mathbf{x} = \frac{\dot{\alpha}_t}{\alpha_t}\left(\boldsymbol{x}_t - \frac{\hat{\epsilon}_\theta}{\sigma_t}\right)\mathrm{d}t \tag{A40}$$

A.4 Implementation details

A.4.1 Obtaining the Coarse Videos

In Section 6.2, we use 3-D models to render an image into a corresponding coarse warped video based on its camera intrinsics and poses. First, we leverage a pretrained depth estimation model, DepthPro (Bochkovskii et al., 2024), to get the depth map of the first frame image. Then we reproject this depth map into a point cloud and render a sequence of images for given camera motion poses. To ensure motion quality, we filter out static or minimal-movement sequences where the camera translation extent is less than 8.

To resolve the scale ambiguity of monocular depth maps, we align them with the camera trajectories using SIFT feature matching (Lowe, 2004) between the reference frame and subsequent frames (sampled at 5-frame intervals). Matches are refined using a k-nearest neighbor search ($k = 2$) and a Lowe's ratio test threshold of 0.75. After computing candidate scales via projection constraints, we discard the top and bottom 5% outliers and apply the median value as the final scale. The reference image is warped to target views by unprojecting valid pixels (depth > 0) into 3-D space using the inverse intrinsic matrix and the calibrated depth. These points are transformed into the target camera coordinate system and splatted onto the 2-D image plane.

To correctly resolve self-occlusions during rendering, we implement a Z-buffering approach by sorting the projected points in descending order of depth. To address deocclusion holes inherently caused by forward splatting, we implement a global nearest-neighbor filling strategy utilizing a k-d tree built on valid pixel coordinates. The corresponding visibility masks are further refined using morphological opening with a $5 * 5$ rectangular kernel to eliminate isolated projection noise and smooth edges. More details can be found in the code.

A.4.2 Image-to-Video Prompts Preparation.

For prompts generation, we follow the protocol of CogVideoX (Yang et al., 2024b), *i.e.*, leverage the pretrained Vision-Language model GPT-4o (Hurst et al., 2024) to generate caption for the given first frame. These generated prompts will be used as the text condition for subsequent image-to-video generation.

A.4.3 The Implementation of Using Wan2.2 for Camera-controlled Video Generation.

For Wan2.2 (Wan et al., 2025), we change the length and resolution of evaluated videos, *i.e.*, the frame number from 49 to 81 and the resolution from 720×480 to 832×480. During sampling, we set the weight function $\lambda_\sigma = \sigma^\alpha$ and give these two parts different α. For the valid part, $\alpha = 16$, and for the invalid part, $\alpha = 20$. For other hyperparameters of Wan2.2, we follow the implementations of TTM (Singer et al., 2025).

A.5 Additional results

A.6 Compatibility with Flow Matching Models

To further verify the generalization ability of our method, we evaluated it on the camera-controlled video generation task (*c.f.*, Section 6.2) by using a flow-based model, Wan2.2 (Wan et al., 2025). We gave qualitative results of Wan2.2 in Figure A5. We can observe that when the pretrained model is Wan2.2, the synthetic videos also yield a decent synthetic result (even better than CogVideoX since Wan2.2 is a stronger base model) under the guidance of the coarse video. This indicates that our method is compatible with both score-based models like CogVideoX and flow-based models like Wan2.2, showing good generalization. This indicates that, although our proposed Weighted h-Transform Sampling is based on the score perspective, it can generally be compatible with flow matching and score matching.



Figure A5: Our method is compatible with both score-based CogVideoX and Flow-based Wan2.2.

A.6.1 Additional Qualitative Results

We ablated α in the initialization weight of $\lambda = \sigma^{\alpha}$ and provide visualizations in Figure A6. Besides, we provide additional qualitative comparisons with SDEdit (Meng et al., 2021) (*c.f.*, Section 6.1) on four image restoration tasks in Figure A7 and Figure A8. Then, we provide additional qualitative comparisons with TTM (Singer et al., 2025) and GWTF (Burgert et al., 2025) (*c.f.*, Section 6.2) on camera-controlled video generation tasks in Figure A9. Moreover, we gave more camera-controlled video generation results with Wan2.2 as the base model in Figure A10. Finally, we provided some text-based image editing cases of our method in Figure A11.



Figure A6: **Super-resolution results by using different** α. A large α leads to a small weight, and vice versa (weight scheduler $\lambda_\sigma = \sigma_t^\alpha$).

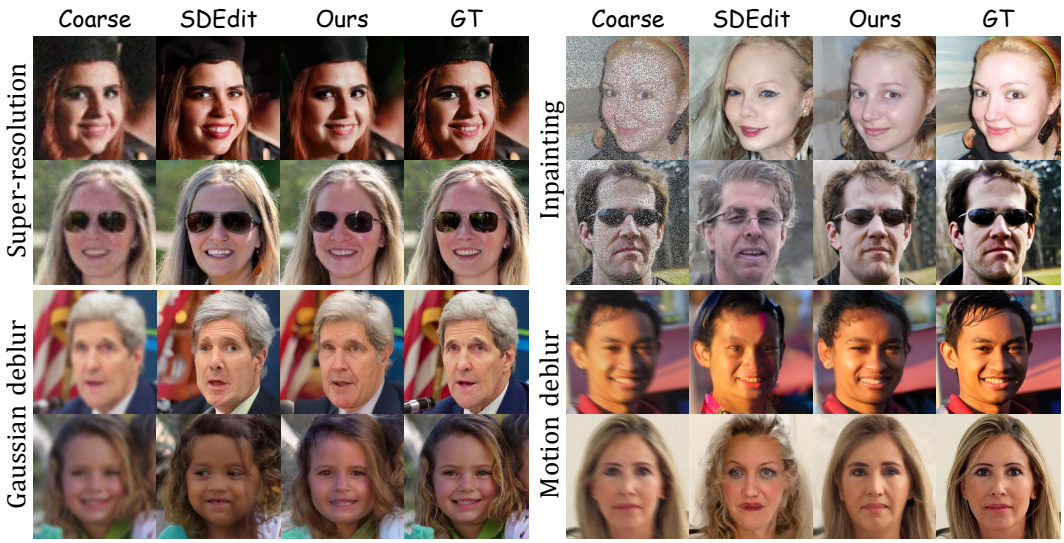

Figure A7: **Additional qualitative results on image restoration tasks.**

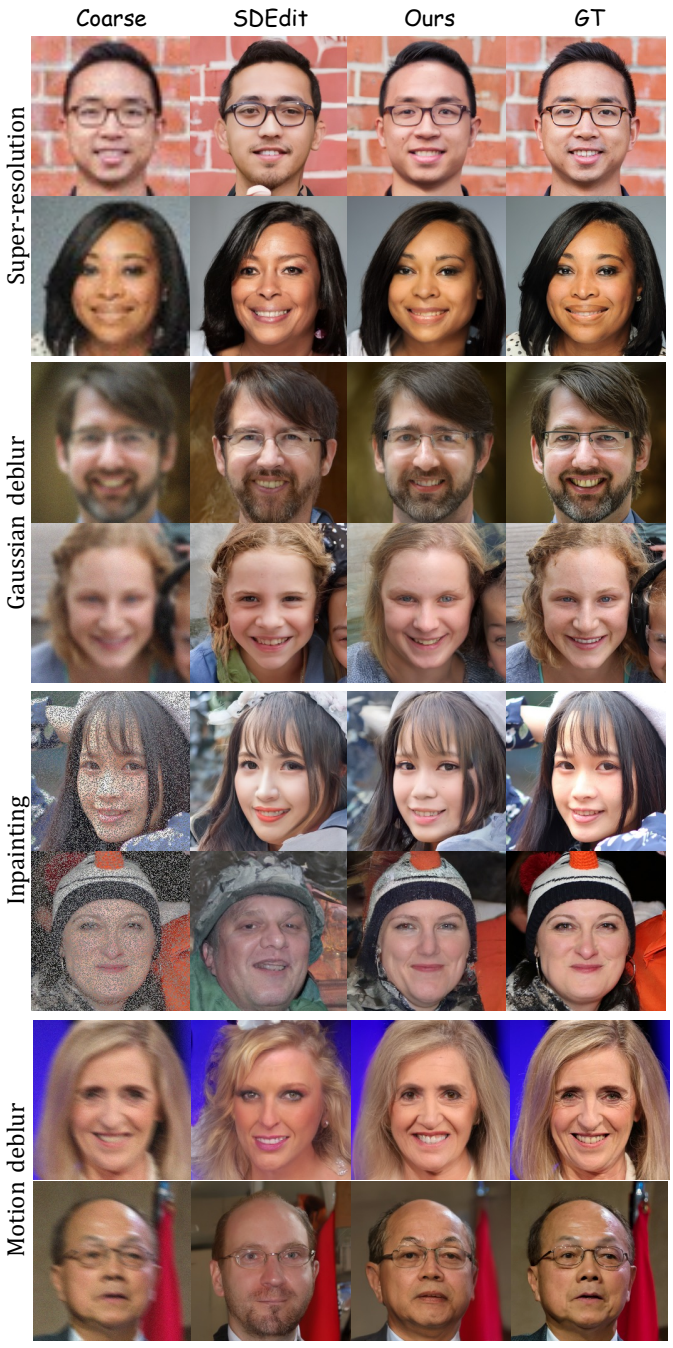

Figure A8: Additional qualitative comparisons with SDEdit on image restoration tasks.

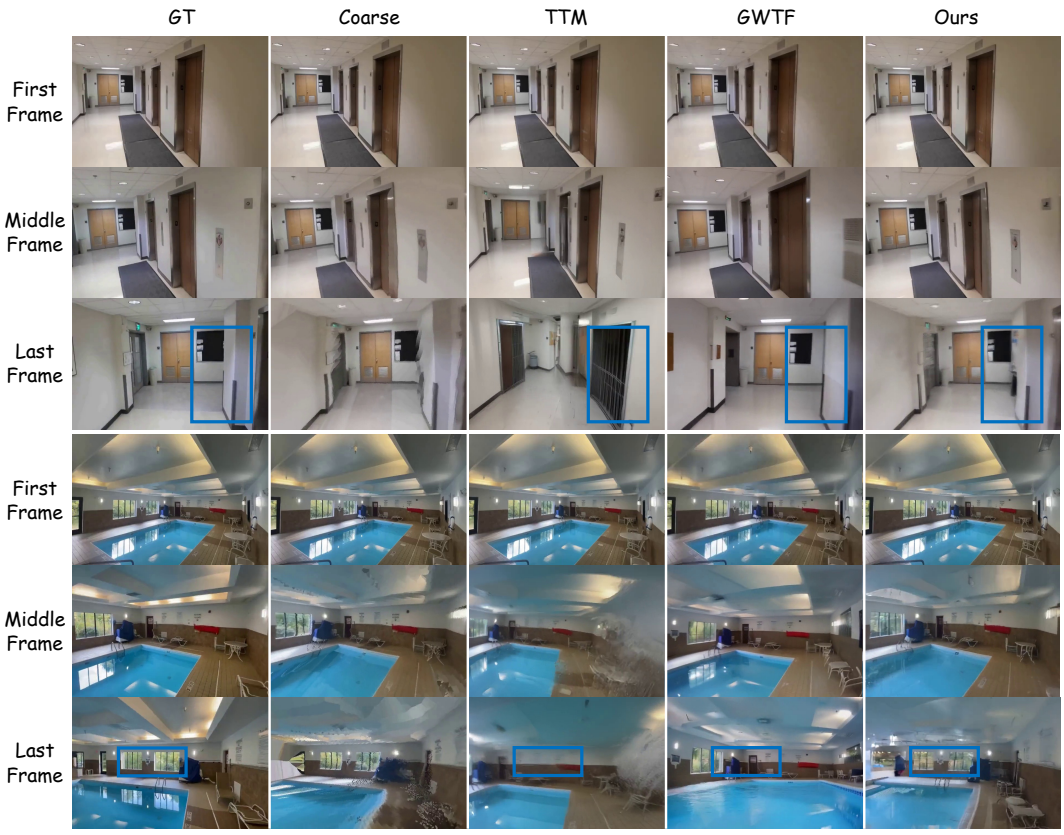

Figure A9: Additional camera-controlled video generation qualitative comparisons.

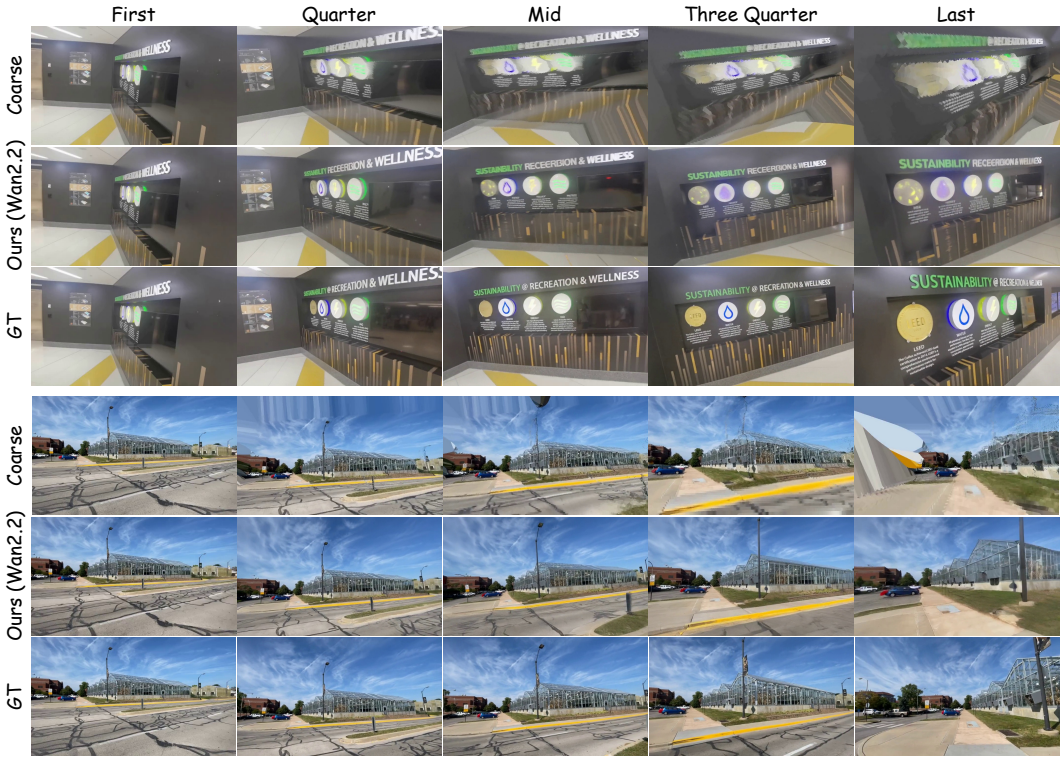

Figure A10: Additional camera-controlled video generation results using Wan2.2.

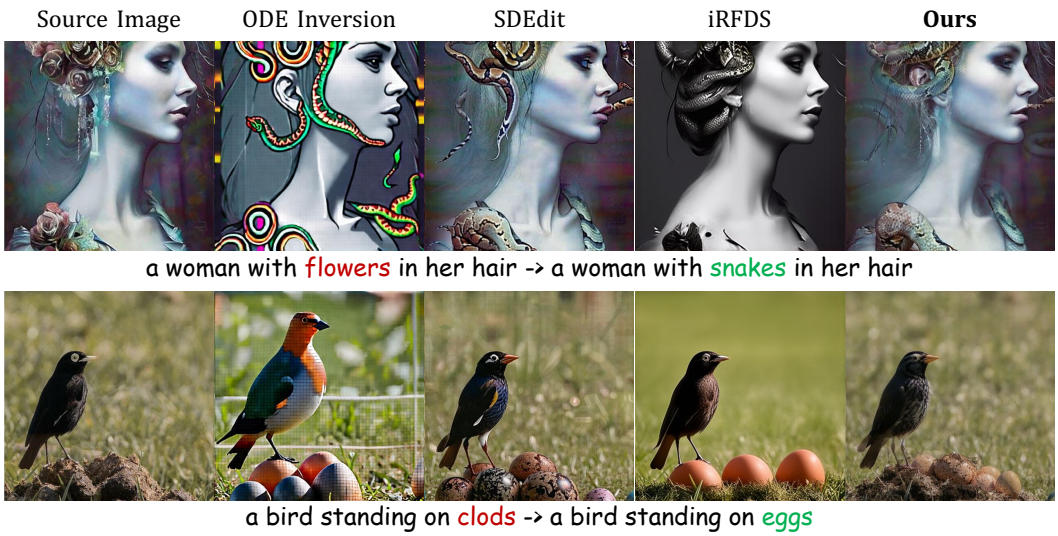

Figure A11: Text-based image editing comparisons with three baselines on PIE-Bench