Coupled Guidance for Flow-Matching Models

Mateo Clemente
Huawei
mateo.clem@gmail.com

Léo Maxime Brunswic
Huawei
leo@brunswic.fr

Rui Heng Yang
Huawei
rui.heng.yang@huawei.com

Amir Rasouli
Huawei
amir.rasouli@huawei.com

Abstract

Guidance methods are essential tools for steering the generation process in diffusion models, improving the generalizability of stochastic robotic controllers and enabling high-quality controlled image synthesis. Despite strong parallels between flow-matching and diffusion, integrating guidance into flow-matching models remains quite challenging. Recent work has proposed general guidance frameworks for flow-matching, however, they are direct adaptations of diffusion guidance methods and can prove unstable. In this work, we conduct a theoretical analysis of guidance in flow-matching and show that a naive translation of diffusion-based guidance to flow-matching breaks the coupling distribution used during training, resulting in weaker control over sampling accuracy. Building on this insight, we propose **Coupled Guidance**, a flow-matching-specific guidance mechanism derived from a geometric perspective that leverages the coupling structure inherent to flow-matching training. To validate the effectiveness of our approach, we conduct extensive experimentation on a custom robotic collision avoidance task using MANISKILL3, four robotic dexterous manipulation tasks from the ADROIT benchmark, and three image generation tasks, namely inpainting, super-resolution, and deblurring, using the ImageNet-128 and CelebA-HQ-256 datasets. Our findings show that our framework improves sample quality and guidance stability across all tasks, without imposing any additional overhead.

1 Introduction

Flow-matching [37, 39] has recently emerged as a powerful class of generative models, unifying advantages from both normalizing flows [45, 24] and diffusion models [27, 31]. The diffusion-inspired MSE loss in flow-matching removes the need for backpropagation through an ODE solver, improves numerical stability, and introduces a beneficial geometric bias [63]. Meanwhile, Neural ODE parameterizations [8] offer highly expressive continuous-time dynamics and efficient sampling. Owing to this combination of features, flow-matching achieves strong results across various domains of generative AI, including robotic control [64, 4, 68], image generation [58, 54], audio synthesis [65, 33], and forecasting [23, 51].

As flow-matching continues to scale to more complex domains, mechanisms for *controlling* its sampling process become increasingly important. Guidance has played this role with great success in diffusion models [26, 1, 32], enabling interactive editing [16], improved action consistency [5], and improved robustness in downstream tasks. In robotics, for example, diffusion-based controllers are often trained offline, hence may never encounter obstacles during training. Guidance allows the injection of collision-avoidance heuristics at inference time [70, 69], adapting to previously unseen environments. In general, guidance methods steer the generation by perturbing model parameters [49],

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

reweighting distributions via task objectives [43], or conditioning on auxiliary cues, such as class labels [15]. Despite their success in diffusion models [35, 71, 70, 55, 69], analogous techniques for flow-matching remain underdeveloped. Recently, a flow-matching guidance framework [20], obtained by adapting diffusion-based guidance mechanisms to flow-matching, demonstrated promising results in locomotion and image synthesis. However, these adaptations are direct translations of their diffusion counterparts and do not explicitly account for the training coupling that is intrinsic to flow-matching but absent in diffusion models.

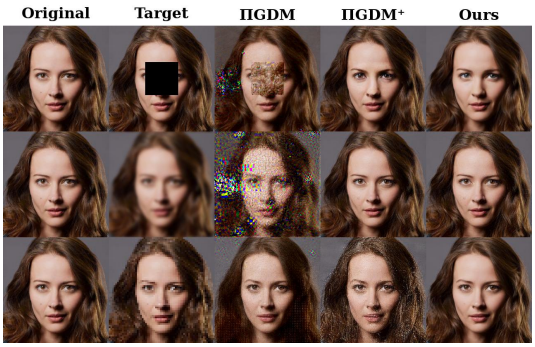


Figure 1: Qualitative examples of image reconstructions for noiseless inpainting, deblurring, and super-resolution, comparing naive ΠGDM , ΠGDM_{+}, and our coupled guidance method.

In this work, we revisit training-free guidance for flow-matching from the perspective of the latent-data coupling on which flow-matching is built. This coupling, fixed during training, ensures that the model learns a consistent joint evolution of latent variables and data. We show that naive guidance violates this coupling, producing mismatched trajectories that hinder controllability and sampling accuracy. Leveraging this insight, we propose *Coupled Guidance*, an efficient correction to the guided sampling process that preserves the training coupling. Our method employs a coordinated pair of guidance terms: a *data-guidance* component acting in *data space* and a corresponding *latent-guidance* component acting in *latent space*. Together, these terms maintain the structure of the flow-matching coupling, resulting in robust and controllable guided sampling with negligible additional overhead and minimal hyperparameter tuning. We evaluate the proposed Coupled Guidance method on robotic motion control with obstacle avoidance, four dexterous-hand manipulation tasks, and three image generation problems, namely inpainting, deblurring, and super-resolution (see Figure 1).

In sum, the main contributions of our work are as follows: i) A correction of guidance for flow-matching that jointly controls guidance in data space and in latent space; ii) A theoretical bound demonstrating the necessity of synchronizing guided latent and data distributions; iii) Extensive empirical evaluations on robotics and image generation tasks to validate the proposed guidance mechanism.

2 Related Works

Diffusion Models and Flow-Matching Diffusion models [15, 57] are a class of generative frameworks that operate by adding random noise to data samples to learn a reverse denoising process in order to generate new samples. Although diffusion models have been very effective in various generative tasks [70, 10, 16, 28], they suffer from multiple shortcomings, such as noise distribution assumptions and computationally expensive denoising processes. These have led to a number of recent works on solving these issues [53, 31, 74, 11].

Flow-matching [37] remedies these shortcomings by directly learning a continuous transport vector field, mapping the latent distribution to the data distribution. Unlike diffusion models, flow-matching enables the choice of any latent distributions (not necessarily Gaussian) and coupling, enhancing the fidelity of generative modeling and simplifying the transport plans [39]. As a result, high-quality samples can be achieved with very few sampling steps, as low as a single function evaluation [21]. Due to such characteristics, flow-matching has been adopted widely in various domains of AI, ranging from audio [65, 33] and image generation [58, 54] to forecasting [23, 51], and robotic control and planning [64, 4, 68].

Guidance Methods In many generative tasks, it is essential to control the sampling process to meet certain criteria, such as generating new images given specific text prompts [34]. Guidance methods are effective mechanisms for controlling the sampling process and can be applied in different ways. One way is to apply guidance using a pretrained classifier to condition the sampling process [12, 57, 75, 36]. This approach, however, limits the diversity of samples as the conditioning

is based on a limited set of classes. In addition, such a method requires large-scale retraining if the criteria are changed. For instance, in the context of obstacle avoidance, over 500K training samples are required to capture various goal configurations and obstacle layouts to cover a simple workspace [12].

Training-free guidance methods [55] lower the computational burden by applying and optimizing a cost function directly at inference time, hence eliminating the need to retrain the generative model. These approaches have been widely applied to diffusion policies for robotic control applications in which responding to novel or unexpected situations is paramount [6, 70, 19, 48, 69]. Despite the great success of training-free guidance methods, their direct application to flow-matching is not trivial. Feng et al. [20] propose a comprehensive framework for the translation of diffusion guidance to flow-matching. This approach enables the adaptation of several guidance methods from diffusion to flow-matching and provides a strong baseline for our work. Nevertheless, these adaptations do not account for the training coupling. As we show in the following section, this omission can result in the velocity field being evaluated at points that lie far outside the coupling region, where the model has not been trained.

Base Methods Unlike in diffusion, the literature in pixel-space training-free guidance is still scarce for flow matching, making the number of relevant baselines limited. We chose to study two paradigms, namely Jacobian-based and Jacobian-free guidance by testing our method on ΠGDM and $g^{\text{sim-inv-A}}$ [20], we refer to the latter as ΠGDM_+.

ΠGDM is a direct adaptation of the pseudoinverse-based guidance [56] to the flow-matching setting. In noiseless inverse problems, the guidance takes the form,

$$\Pi\text{GDM}(x, y, H) = -\lambda_t((H^\dagger y - H^\dagger H\hat{x}_1)^T \frac{\partial \hat{x}_1}{\partial x_t})^T$$

where λ_t denotes the guidance schedule, H is the degradation operator (e.g., masking for inpainting, blurring for deblurring, or downsampling for super-resolution), and $y = Hx^*$ is the degraded observation of the target image x^*. The operator $H^\dagger = H^\top(HH^\top)^{-1}$ corresponds to the Moore–Penrose pseudoinverse of H. ΠGDM_+follows that exact formulation but approximates $\frac{\partial \hat{x}_1}{\partial x_t}$ as the identity.

Finally, since no trivial adjoint operator H^T is available for collision avoidance, we instead apply guidance by directly using g^{cov-A} which minimizes an energy function,

$$g_{\text{collision}} = -\lambda_t \nabla_{\widehat{x}_1} ||\widehat{x}_1 - y||$$

with y the target generated by our collision avoidance pipeline. We provide a discussion about our choice of baseline in Appendix B.1.

3 A Short Theory of Coupled Guidance

3.1 Notations

We consider a data space $\mathcal{X} := \mathbb{R}^d$, viewed as an affine space with underlying vector space $\overrightarrow{\mathcal{X}}$. It is equipped with its Borel σ-algebra and the Lebesgue measure λ. We denote measures by Greek letters μ, ν, π. A *distribution* is a probability measure on $\mathcal{X}$, and a *coupling* of two distributions μ and ν is a distribution π on $\mathcal{X} \times \mathcal{X}$ such that its marginals are μ and ν respectively.

3.2 Flow-Matching Training and Inference

A flow-matching model consists of a tuple $(\mu_1, \mu_0, \pi, v^\theta)$ where μ_1 is the data or target distribution, μ_0 is a latent distribution (typically Gaussian), π is a coupling between them, and $v^\theta : \mathcal{X} \times [0,1] \to \overrightarrow{\mathcal{X}}$ is a learnable vector field.

The model is trained using a linear flow-matching objective,

$$\mathcal{L}(\theta) = \mathbb{E}_{(x_0,x_1)\sim\pi} \int_0^1 \|v_t^\theta(x_t) - v_t(x_t)\|^2 \, dt = \mathbb{E} \int_0^1 \|v_t^\theta(x_t) - (x_1 - x_0)\|^2 dt + \text{Cte},$$

where $x_t = (1-t)x_0 + tx_1$ and $v_t(x_t) = \mathbb{E}[x_1 - x_0 \mid x_t]$.

At inference time, one solves the ODE $\dot{x}_t = v^\theta(x_t, t)$, typically via a first-order Euler scheme:

$$x_{t+\delta t} = x_t + v_t^\theta(x_t)\,\delta t, \tag{1}$$

We can extend this Euler step for a one step prediction:

$$\widehat{x}_1 = x_t + (1-t)v^\theta(x_t, t) \tag{2}$$

3.3 From Guided Flow to Coupled Guidance

In practice, guidance takes the form of a task-specific function $\mathcal{G} : \mathcal{X} \to \overrightarrow{\mathcal{X}}$, which acts as a correction to the predicted clean sample. For clarity, we omit conditioning variables (e.g., observation o).

By analogy with universal diffusion guidance [1], each inference step may be interpreted as (i) predicting a denoised endpoint $\widehat{x}_1$, followed by (ii) interpolating toward it,

$$x_{t+\delta t} = \frac{1-t-\delta t}{1-t}\,x_t + \frac{\delta t}{1-t}\,\widehat{x}_1. \tag{3}$$

Naive guidance first modifies the predicted endpoint,

$$\widehat{x}_1^g = \widehat{x}_1 + \alpha_t \mathcal{G}(\widehat{x}_1), \tag{4}$$

$$x_{t+\delta t}^{g,\text{naive}} = \frac{1-t-\delta t}{1-t}x_t + \frac{\delta t}{1-t}\,\widehat{x}_1^g, \tag{5}$$

but this formulation obscures the underlying coupling structure on which flow-matching is trained. We rewrite inference as a coupling update by defining the reconstructed endpoints as

$$\widehat{x}_0 = x_t - t\,v^\theta(x_t, t) \quad \text{and} \quad \widehat{x}_1 = x_t + (1-t)v^\theta(x_t, t) \tag{6}$$

These endpoints are not estimates of the backward flow map, i.e. of the latent that the learned ODE would reach if integrated back to $t = 0$. They are the two endpoints of the training interpolation $x_t = (1-t)x_0 + tx_1$: since this interpolant is linear and $v_t(x_t) = \mathbb{E}[x_1 - x_0 \mid x_t]$, the minimizer v_t gives exactly $\widehat{x}_1 = \mathbb{E}[x_1 \mid x_t]$ and $\widehat{x}_0 = \mathbb{E}[x_0 \mid x_t]$, the posterior means of the training coupling given x_t, which v^θ estimates. No straight-trajectory assumption is involved: both endpoints are recomputed from v^θ at every step, so the curvature of the marginal flow is tracked rather than assumed away.

Then the Euler update (1) is equivalent to

$$x_{t+\delta t} = (1-t-\delta t)\widehat{x}_0 + (t+\delta t)\widehat{x}_1, \tag{7}$$

a linear interpolation between the latent and target endpoints. When applying naive guidance, the target endpoint becomes $\widehat{x}_1^g$, and the latent endpoint must be modified to

$$\widehat{x}_0^{g,\text{naive}} := x_t - t\,\frac{\widehat{x}_1^g - x_t}{1-t}.$$

Thus naive guidance implicitly induces a *latent guidance*

$$\widehat{x}_0^{g,\text{naive}} - \widehat{x}_0 = -\frac{\alpha_t t}{1-t}\,\mathcal{G}(\widehat{x}_1).$$

This expression is a derivation rather than a design choice: at a given x_t, prescribing the target endpoint $\widehat{x}_1^g$ also prescribes its latent partner, and $\widehat{x}_0^{g,\text{naive}}$ is the unique latent consistent with the naive update. Making it explicit exposes how naive guidance breaks the coupling: it silently applies a latent correction directed opposite to the data-side correction. Coupled guidance makes this implicit choice explicit and controllable.

General coupled guidance. Motivated by this observation, we introduce **Coupled Guidance**, the pair $(\mathcal{G}^0, \mathcal{G}^1)$ and the following update rules in the general form:

$$\widehat{x}_0^g = \widehat{x}_0 + \mathcal{G}^0(\widehat{x}_0, \widehat{x}_1, t), \tag{8}$$

$$\widehat{x}_1^g = \widehat{x}_1 + \mathcal{G}^1(\widehat{x}_0, \widehat{x}_1, t), \tag{9}$$

$$x_{t+\delta t}^g = (1-t-\delta t)\widehat{x}_0^g + (t+\delta t)\widehat{x}_1^g. \tag{10}$$

For naive guidance, we have

$$\mathcal{G}^{1,\text{naive}}(\widehat{x}_0, \widehat{x}_1, t) = \alpha_t \mathcal{G}(\widehat{x}_1) \quad \text{and} \quad \mathcal{G}^{0,\text{naive}}(\widehat{x}_0, \widehat{x}_1, t) = -\frac{\alpha_t t}{1-t}\mathcal{G}(\widehat{x}_1). \tag{11}$$

3.4 Training Coupling versus Guided Coupling

A coupled guidance induces a guided interpolant distribution $(\widehat{x}_0^g, \widehat{x}_1^g) \sim (\pi^g \mid o, t)$, which depends on the observation o and timestep t. We continue by characterizing how deviations between π^g and the training coupling π affect flow-matching accuracy, and clarify why naive guidance degrades performance even if ΠGDM mitigates some—but not all—failures.

Assumptions. Following Benton et al. [3], we assume:

- the flows induced by v_t and v_t^θ exist, and are unique and smooth;
- v^θ is differentiable in (t, x) and Lipschitz in x;
- the guided update admits a continuous-time limit $x_{t+\delta t}^g = x_t^g + \delta t\, v_t^g + o(\delta t)$;
- $v_t \mapsto v_t^g$ is Lipschitz with constant L_g.

These hold, in particular, for naive guidance provided $\alpha_t \|\mathcal{G}\| = O(1-t)$ as $t \to 1$. We denote by $\mu_1^{g(o)}$ and $\mu_1^{\theta,g(o)}$ the final guided distribution for the theoretical minimizer v_t of the loss and the trained vector field v_t^θ, respectively.

Theorem 1. *Let $(\mu_1, \mu_0, \pi, v^\theta)$ be a flow-matching model and let g be any coupled guidance. Under the above assumptions,*

$$\mathbb{E}_o\, \mathcal{W}_1\Big(\mu_1^{g(o)},\, \mu_1^{\theta,g(o)}\Big) \le L_g L_\theta\, \mathcal{L}^{1/2}(\theta) \left\| \frac{d\pi^g}{d\pi} \right\|_{L^\infty(\pi)}, \tag{12}$$

where $\mathcal{W}_1$ is the 1-Wasserstein distance, π^g is the guided coupling distribution, $L_\theta = \exp\Big(\int_0^1 L_{\theta,t} dt\Big)$, and $L_{\theta,t}$ is the Lipschitz constant of v_t^θ; See Appendix A for the proof.

Interpretation. The bound highlights the central difficulty of naive guidance: the factor $\frac{d\pi^g}{d\pi}$ measures how far the guided coupling deviates from the training coupling. Under the simplifying assumptions that the couplings are independent, $\pi = \mu_0 \otimes \mu_1$ and $\pi^g = \pi_0^g \otimes \pi_1^g$, the density ratio decomposes into

$$\frac{d\pi^g}{d\pi} \quad = \quad \frac{d\pi_1^g}{d\mu_1} \cdot \frac{d\pi_0^g}{d\mu_0}.$$

The target marginal π_1^g depends on how well the *target guidance* $\mathcal{G}^1$ preserves the data manifold. Under the manifold hypothesis [18], μ_1 is highly anisotropic; ΠGDM improves stability by projecting the guidance direction onto dominant tangent directions of μ_1, thus controlling the data-side marginal density $\frac{d\pi_1^g}{d\mu_1}$. However, computing this projection requires $\frac{d\hat{x}_1}{dx_t}$, which can become ill-behaved when guided trajectories deviate from the training coupling. Using coupled guidance controls the latent-side density, keeping the inference stable.

The latent marginal π_0^g depends on the *latent guidance* $\mathcal{G}^0$. For a Gaussian prior, naive guidance yields

$$\frac{d\pi_0^g}{d\mu_0} \approx \mathbb{E}_{x_1 \sim \mu_1} \int_0^1 \exp\Big(\big\| \tfrac{\alpha_t}{1-t} \mathcal{G}(x_1) \big\|^2 \Big)\, dt,$$

which can explode for moderate guidance strength, causing severe out-of-distribution drift of intermediate x_t^g.

Conclusion. Naive guidance evaluates the learned vector field v^θ at intermediate points x_t^g that may lie far outside the coupling used during training, leading to large density ratios and degraded inference. This motivates designing guidance rules that preserve the latent–data coupling—precisely the principle behind our Coupled Guidance framework.

4 Method

4.1 Coupled Guidance Algorithm

Our method directly instantiates the coupled guidance update rules (8), (9), (10). We use the same target-guidance as standard guided flow, but modify the latent-guidance term to better preserve the

Algorithm 1 Coupled Guidance Sampling

Require: Number of steps T, velocity field $v_t^\theta(x)$, guidance function $\mathcal{G}$, coefficients (α, β), step size δt.

Sample initial latent $x_0^g \sim \mu_0$ and set $t \leftarrow 0$
for $i = 0..T$ **do**
  $v_t \leftarrow v^\theta(x_t^g, t)$
  $\widehat{x}_1 \leftarrow x_t^g + (1-t)v_t$
  $\widehat{x}_0 \leftarrow x_t^g - tv_t$
  $g \leftarrow \mathcal{G}(\widehat{x}_1)$
  $\widehat{x}_1^g \leftarrow \widehat{x}_1 + \alpha g$
  $\widehat{x}_0^g \leftarrow \widehat{x}_0 - \frac{\beta t}{1-t} g$
  $t \leftarrow t + \delta t$
  $x_t^g \leftarrow (1-t)\widehat{x}_0^g + t\widehat{x}_1^g$
end for
return x_t^g

Table 1: Comparison of naive vs. coupled guidance across methods and tasks on ImageNet-128 (minibatch OT).

		Naive		Coupled	
Task	**Metric**	ΠGDM	ΠGDM+	ΠGDM	ΠGDM+
Inpainting	FID↓	18.3	10.3	**8.5**	11.4
	LPIPS↓	0.112	**0.086**	0.088	0.089
	PSNR↑	18.3	20.6	**23.5**	20.3
	SSIM↑	0.903	0.911	**0.916**	0.908
Deblurring	FID↓	70.5	14.3	22.2	**13.2**
	LPIPS↓	0.471	0.197	0.264	**0.184**
	PSNR↑	19.0	23.8	23.5	**24.3**
	SSIM↑	0.449	0.753	0.711	**0.769**
Super-res.	FID↓	49.7	46.0	**8.5**	50.4
	LPIPS↓	0.335	0.396	**0.142**	0.413
	PSNR↑	19.0	17.5	**24.0**	17.3
	SSIM↑	0.645	0.477	**0.816**	0.452

training coupling. Concretely, we define,

$$\mathcal{G}^{1,\alpha}(\widehat{x}_0, \widehat{x}_1, t) = \alpha\, \mathcal{G}(\widehat{x}_1), \tag{13}$$

$$\mathcal{G}^{0,\beta}(\widehat{x}_0, \widehat{x}_1, t) = -\frac{\beta t}{1-t}\, \mathcal{G}(\widehat{x}_1), \tag{14}$$

where (α, β) are hyperparameters controlling data-side and latent-side guidance. The factor $(1-t)^{-1}$ is only evaluated on grid points $t = k/K < 1$, and most guidance schedules decay linearly (or faster) as $t \to 1$, which cancels the singularity; as a safeguard we add a small ϵ to the denominator, and found results insensitive to its value across four orders of magnitude. Algorithm 1 summarizes the full sampling procedure.

This procedure introduces only two hyperparameters and retains the same computational complexity as naively guided flow-matching. Note that $\alpha = 1$ will almost always be the sensible choice, as scaling of data-guidance is in practice already included in the guidance operator itself. Only β needs to be chosen between naive ($\beta = 1$) and coupled guidance ($\beta \in [-1, 0]$). We recommend $\beta \in \{0, -1\}$ as a starting point. Part of the difference between these two values is a guidance-scale effect: at a fixed scale, $\beta = -1$ accumulates roughly $1.9\times$ more guidance than $\beta = 0$ over the trajectory, and once the guidance scale is tuned separately for each β the gap narrows considerably, although a small difference remains. The guidance scale should therefore be tuned per β, as is already standard practice for naive guidance.

Intuitively, β controls both the magnitude and sign of the latent correction relative to the data-side guidance $\mathcal{G}$, acting as an affine parameter along ($\hat{x}_0$,$\hat{x}_0^g$), with a clear geometric interpretation:

- $\beta = 1$: the latent interpolant is $\hat{x}_0^g$, the correction is reversed relative to the data guidance, actively opposing it.
- $\beta = 0$: the latent interpolant remains $\hat{x}_0$ (no latent correction).
- $\beta = -1$: the latent interpolant is set to the symmetric of $\hat{x}_0^g$ with respect to $\hat{x}_0$, i.e. it is pushed in the same direction as the data guidance (since $-\frac{(-1)\,t}{1-t}\mathcal{G} = +\frac{t}{1-t}\mathcal{G}$).

As established in Liu et al. [39, Theorem 3.5], flow-matching training minimizes transport cost, which causes the learned vector field to progressively align latent and data interpolants along trajectories. This means the learned data-latent coupling tends to co-orient $\hat{x}_0$ and $\hat{x}_1$ along the flow. Applying $\beta = 1$ pushes the latent interpolant against the data guidance direction, producing a data-latent coupling that diverges from the learned one. By contrast, $\beta \leq 0$ aligns the latent correction with the data guidance, keeping the guided coupling closer to the learned coupling.

5 Experiments

5.1 Collision Avoidance

Following Chi et al. [9], Clemente et al. [11], we use a 65M-parameter U-Net [52] to parameterize our flow-matching policy, trained on the ManiSkill3 [61] Pick Cube environment and its associated dataset. The action space is 9-dimensional, corresponding to the robot joint positions.

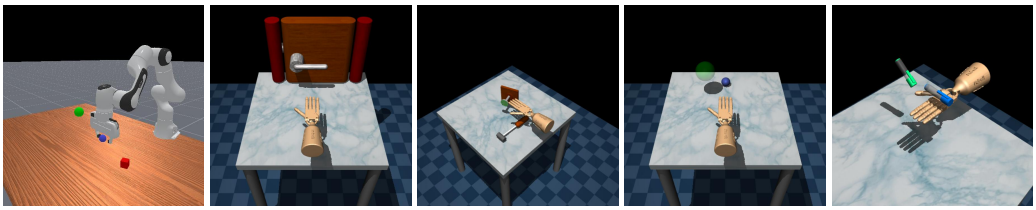


Figure 2: Experimental setups for robotics tasks. From left to right: Maniskill PickCube, Adroit Door, Hammer, Relocate, and Pen.

Table 2: Collision avoidance success rates (%) for all (α, β) couples, over 256 environment seeds.

$\alpha\backslash\beta$	-1	0	1
0	73	21	17
1	**81**	77	36

The model is trained to predict an action horizon of length $h_{\mathcal{A}} = 24$—resulting in an output dimension of $24 \times 9 = 216$—conditioned on an observation horizon of $h_{\mathcal{O}} = 2$ past environment states. On the base environment, the learned policy solves the task almost perfectly, achieving a success rate of 99.8% (see Appendix B.2 for more details). We then evaluate the same policy in a modified environment where an obstacle is introduced between the initial cube position and the target, and apply guidance to enable collision avoidance. The guidance function is defined using a Euclidean signed distance field (ESDF), computed with cuRobo [59], and acts directly on the predicted joint trajectories to penalize collisions. This form of guidance is inherently noisy, as it perturbs each joint state independently across the prediction horizon, resulting in a large deviation between the guided and training couplings, reflected by a high density ratio $d\pi_1^g / d\mu_1$.

Without guidance ($\alpha = 0$, $\beta = 0$), the policy's performance drops to approximately 21%, as trajectories frequently collide with the obstacle. In this setting, ΠGDM is not directly applicable, as it requires access to an adjoint or degradation operator, which is unavailable for this task. We therefore apply guidance by directly minimizing the ESDF-based energy. Under naive guidance ($\alpha = 1$, $\beta = 1$), performance only improves up to 36%. In contrast, coupled guidance consistently yields strong performance despite the noisy nature of the guidance signal, achieving up to 81% success rate with $\alpha = 1$ and $\beta = -1$. Results for different tested (α, β) configurations are reported in Table 2.

5.2 Robotic Control as an Inpainting Task

We emulate a two-level hierarchical motion controller in which a high-level planner provides partial information about future trajectories to a low-level controller, thereby turning control into an inpainting problem. To this end, we train two flow-matching models per task. The high-level planner is a *conditioned* model that, given an observation history, predicts a horizon of $h = 24$ future state–action pairs $(o_t, a_t)_{t=1}^{h}$. The low-level controller is trained on the same dataset but remains fully *unconditioned*: it receives neither observations nor waypoints as inputs, and planner information enters only at inference time, through the guidance term.

At test time, the controller receives sparse supervision from the planner and its task is to infer a complete action sequence consistent with this partial trajectory information. Intuitively, the planner acts as a world-aware module that provides intermittent *waypoints* in the trajectory space, while the controller acts as an action expert that fills in the missing degrees of freedom. We define the guidance density (the sparsity cutoff in Figure 3) as the proportion of entries of the (horizon $\times$ action) trajectory matrix constrained by guidance; guided entries are selected elementwise, uniformly at random. Robustness to low densities matters because hierarchical controllers operate in this regime: a high-level planner typically emits a few waypoints over the horizon rather than dense trajectories, owing to planning frequency, communication budget, or abstraction level. Planner-like protocols (temporal waypoints and temporal blocks) give the same ordering, and the results hold across couplings and priors (Appendices D.1 and D.2).

We evaluate robustness to guidance sparsity under both naive and coupled guidance on the four Adroit dexterous manipulation tasks [50, 22]: Door (latch opening), Pen (reorientation to a target angle), Hammer (nail hammering), and Relocate (object relocation) (Figure 2). Different seeds correspond to variations in task configuration (e.g., door pose, target angle, nail position, and object/goal locations). As ΠGDM and ΠGDM$_+$behave similarly, we present results for ΠGDM in Figure 3 and include additional results in Appendix C.1. For all runs we used 300 seeds.

As shown in Figure 3, coupled guidance is substantially more robust to sparse guidance than naive guidance, without requiring task-specific tuning. Even when sweeping a wide range of guidance scales, naive guidance degrades sharply once 20% or fewer dimensions are guided, whereas coupled guidance maintains near-fully-guided performance down to approximately 5% guided dimensions. Naive guidance can perform well with careful tuning, but requires different guidance scales across sparsity regions, whereas coupled guidance remains stable across all sparsity levels without tuning. When both arms are re-tuned at every density, the optimal scale of coupled guidance does not change, while that of naive guidance varies by a factor of two (Appendix D.2). A classifier two-sample test further shows that coupled guidance keeps guided endpoint pairs much closer to the training coupling than naive guidance (Appendix D.3). Additional results are provided in Appendix C.1.

Since inpainting-based guidance has recently been adopted to improve trajectory consistency in generalist robotic policies [5], methods such as ΠGDM have clear relevance to real-world robotic control. The increased stability provided by coupled guidance directly benefits these applications by reducing sensitivity to sparse or imperfect guidance, enabling more reliable deployment without overhead.

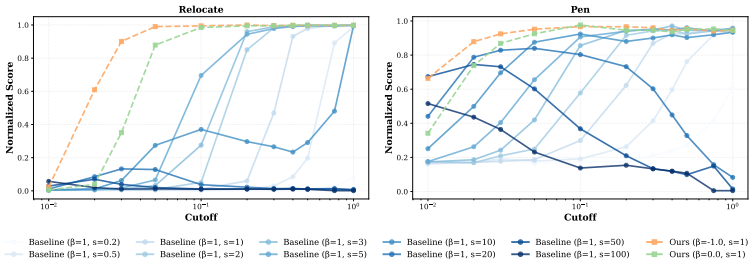


Figure 3: Performance of ΠGDM on Adroit tasks given different sparsity cutoffs. Runs using coupled guidance use a unit guidance scale $s = 1$, without tuning. The blue lines each represent the performance of naive guidance for different guidance scalings s.

5.3 Image Generation

For the image generation tasks, we study the behavior of ΠGDM and ΠGDM$_+$under naive and coupled guidance on box inpainting, deblurring and factor 4 super-resolution. We train a U-Net model from scratch on the CelebA-HQ-256 [40, 30] and ImageNet-128 [13] datasets (see Appendix B.3 for more details). All runs use the same base flow-matching model and are compared via 4 image quality metrics: FID [25], LPIPS [73], PSNR and SSIM [66]. To cover usual settings, we also train a mini-batch OT model with a similar backbone for ImageNet-128. Given that the results are consistent for both datasets and coupling choices, we make general comments in the following section for conciseness. The full tables and plots of the results are provided in Table 1 and Appendix C.2. Results with a class-conditional model, an external baseline (PnP-Flow) and noisy observations are reported in Appendices D.5 and D.6.

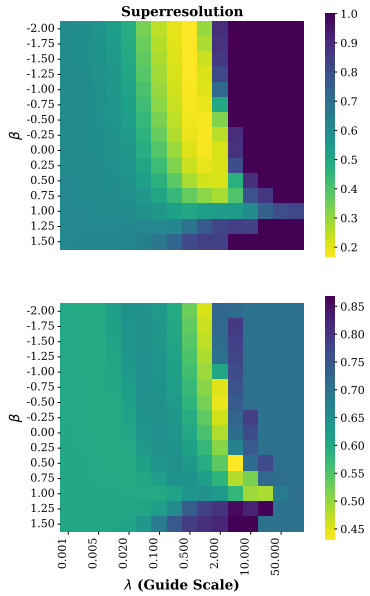


Figure 4: LPIPS scores for ΠGDM , (first row) and ΠGDM$_+$(second row) on super-resolution

We evaluate our method on noiseless linear inverse problems:

$$y = Hx + \sigma_y, \quad \text{with} \quad \sigma_y = 0.$$

Although adding noise may make an accurate reconstruction more difficult, in practice it helps stabilize the guidance process, and $\sigma_y = 0$ is actually the regime in which training-free guidance is most prone to instability. It has been shown in the previous studies [67, 2, 62] that nonzero measurement noise σ_y acts as an implicit regularizer, i.e. it smooths the posterior score so that the linear guidance term dominates for most of the trajectory [2], reduces the effective curvature of the likelihood gradient [62], and in ΠGDM directly Tikhonov-regularizes the covariance inversion via the σ_y^2 I term. When σ_y approaches 0, these stabilizing effects vanish: the

guidance sharpens, instability increases, and approximation errors in the score are no longer masked by a forgiving posterior geometry. This makes the noiseless setting pertinent for our study as it highlights a failure mode where the stabilizing effects of coupled guidance are necessary.

Inpainting In this setting, naive ΠGDM$_+$exhibits stable and reliable performance across all metrics, whereas ΠGDM shows worse performance, producing visible artifacts. This behavior, illustrated in Figure 8 (Appendix C.2), is consistent with the analysis in Section 3, as ΠGDM relies on the Jacobian $\partial\widehat{x}_1/\partial x_t$, which becomes poorly conditioned when guided trajectories deviate from the training coupling. Coupled guidance has little effect on ΠGDM$_+$, whose stability primarily stems from the identity Jacobian approximation. However, it significantly stabilizes ΠGDM by preserving the training coupling. As a result, coupled ΠGDM outperforms naive ΠGDM on all metrics and datasets, and on ImageNet-128 it also outperforms naive ΠGDM$_+$in FID and PSNR; on CelebA-HQ-256, however, naive ΠGDM$_+$remains the strongest method for inpainting on FID, LPIPS and SSIM (Appendix C.2).

Super-resolution In this sparse guidance setting, both ΠGDM$_+$and ΠGDM fail to produce meaningful reconstructions, yielding poor image quality across all evaluation metrics, as shown in Figure 4. Like in the inpainting case, introducing coupled guidance has negligible impact on ΠGDM$_+$, whose performance is fundamentally constrained by the identity Jacobian approximation and thus poorly suited to this task. In contrast, coupled guidance substantially stabilizes ΠGDM by mitigating excessive latent-side drift, enabling reliable sampling throughout the trajectory. This stabilization translates into a clear improvement in sample quality, with the coupled ΠGDM variant consistently outperforming all other methods.

Deblurring In this task, both ΠGDM$_+$ and ΠGDM benefit from coupled guidance. ΠGDM$_+$by default produces high-quality samples under naive guidance, indicating that the identity Jacobian approximation is well suited to this type of degradation. Still, coupled guidance further improves the reconstruction quality across all metrics in this setting. Coupled guidance likewise stabilizes ΠGDM and enables reliable inference; however, even with this stabilization, its performance remains below that of ΠGDM$_+$. Overall, both methods benefit from coupled guidance in this setting even if this time ΠGDM$_+$ emerges as the strongest method. Additional illustrations for image quality metrics are provided in Appendix C.2.

Jacobian Influence Figure 5 illustrates how coupled guidance affects the contribution of the Jacobian-dependent term throughout the inference process. Note that to remain fair to the naive setting, we plot the best run for each β after tuning. The term is initially large during the early steps, reflecting the expected role of the Jacobian in aligning the prediction with the inverse constraint. Crucially, when $\beta = 1$ (naive guidance), the Jacobian contribution remains substantially larger in the stable zone, and the final destabilization occurs earlier, making the final divergence more pronounced. Decreasing β effectively dampens this amplification, extending the stable regime and delaying the time at which the term starts being ill-behaved. This observation provides direct empirical evidence that coupled guidance helps control the magnitude and timing of the Jacobian-induced corrections by preserving the training coupling, thereby preventing instability without degrading the guidance signal.

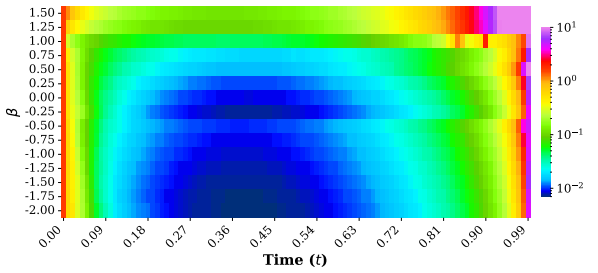


Figure 5: Evolution of $\partial\widehat{x}_1/\partial x_t$'s contribution to the guidance norm for the CelebA-HQ-256 superresolution task through the ΠGDM inference process.

6 Limitations

Simplified theoretical setting. Our analysis is conducted under a simplified formulation using the unweighted quadratic flow-matching loss, linear couplings, and a continuous-time limit. Weighted objectives are the easy case: the weighting only enters at the Cauchy–Schwarz step of

Appendix A, so for any weighting w_t bounded below, the bound holds with $\mathcal{L}^{1/2}(\theta)$ replaced by $(\mathcal{L}_w(\theta)/\inf_t w_t)^{1/2}$; general weighted Bregman divergences, which cover the diversity of flow-matching objectives in the literature [38], would require more than a constant. The proof does not use the linearity of the interpolant, only that the evaluation point is a deterministic function of (x_0, x_1, t), so we expect Theorem 1 to transfer to non-linear interpolants; what is specific to linear couplings is the method, since Eq. (6) and the $t/(1-t)$ factor come from inverting the linear interpolation and would have to be re-derived for other interpolants. Extending the analysis to more sophisticated geometries [7] remains out of scope. Finally, Theorem 1 relies on the existence of a continuous limit, whereas the practical schedules we consider do not have such a limit; we do not control the resulting discretization term, which affects guided and unguided sampling alike.

What the bound explains about β. Under the decomposition of Section 3.4, β only enters the bound through the latent factor $d\pi_0^g/d\mu_0$, and $\beta = 0$ is the unique choice that leaves the latent endpoint unperturbed, hence the unique choice for which this factor keeps its unguided value. However, this factor depends on the latent displacement $-\frac{\beta t}{1-t}\mathcal{G}$ only through its magnitude, so the bound scores $\beta = 1$ and $\beta = -1$ identically, whereas our experiments separate them sharply. Explaining the sign of β requires the law of the evaluation point $x_t^g = (1-t)\widehat{x}_0^g + t\widehat{x}_1^g$, in which $\beta \leq 0$ displaces the latent-side interpolant in alignment with the data-side one (Section 4). The optimal, possibly time-dependent, β_t thus trades off two effects, out-of-distribution drift and direction mismatch, which our theory captures separately but not jointly.

Noiseless image inverse problems. Our work primarily targets noiseless inverse problems, where guidance-induced coupling mismatch is most pronounced and cannot be mitigated by observation noise. In noisy inverse problems, noise implicitly regularizes the posterior and stabilizes Jacobian-based guidance, partially alleviating the issues we address. Moreover, in such settings the effect of coupled guidance interacts with the residual scaling schedule and the noise variance term, making it less straightforward to differentiate improvements due to coupling preservation from those induced by schedule design. In an ablation on ImageNet-128 deblurring with σ_y up to 0.2 (Appendix D.6), Jacobian-based guidance retains a 2.6–3.4 dB advantage when coupled across the whole noise range, whereas the gains of Jacobian-free guidance largely vanish; the benefit of coupled guidance in noisy tasks therefore depends on the guidance operator. Still, since many applications require noiseless inverse guidance (photo editing, content super-resolution, general deblurring), we deem this restriction reasonable.

7 Conclusion

We proposed *coupled guidance*, a training-free mechanism that exploits the geometric structure of flow-matching couplings to stabilize guided generation. By jointly leveraging information from both latent and data spaces, our approach achieves competitive or superior performance across robotics and image inverse problems while requiring minimal hyperparameter tuning and negligible computational overhead. The gains are largest for Jacobian-based guidance such as ΠGDM , whose Jacobian term is most sensitive to evaluations outside the training coupling, and more modest for Jacobian-free guidance such as ΠGDM_+; beyond inverse problems, the collision-avoidance task shows that the mechanism also applies to generic energy-based guidance. Extensive experiments show that coupled guidance improves robustness to strongly out-of-distribution guidance signals and guidance sparsity, while improving the quality of generated samples over all tested dimensionalities and application domains. The observed behaviors validate our theoretical insights and demonstrate that coupled guidance offers a simple, scalable improvement to existing methods. We believe that this approach will extend naturally to more complex coupling structures and that it opens the door for more fine-grained latent control.

References

[1] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *CVPR*, 2023.

[2] Gregory Bellchambers. Exploiting the exact denoising posterior score in training-free guidance of diffusion models, 2025.

[3] Joe Benton, George Deligiannidis, and Arnaud Doucet. Error bounds for flow matching methods. *TMLR*, 2024.

[4] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *CoRL*, 2025.

[5] Kevin Black, Manuel Y Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *NeurIPS*, 2025.

[6] João Carvalho, An T. Le, Piotr Kicki, Dorothea Koert, and Jan Peters. Motion planning diffusion: Learning and adapting robot motion planning with diffusion models. *IEEE Transactions on Robotics*, pages 1–20, 2025.

[7] Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *ICLR*, 2024.

[8] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018.

[9] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[10] Younwoo Choi, Ray Coden Mercurius, Soheil Mohamad Alizadeh Shabestary, and Amir Rasouli. Dice: Diverse diffusion model with scoring for trajectory prediction. In *IV*, 2024.

[11] Mateo Clemente, Leo Brunswic, Rui Heng Yang, Xuan Zhao, Yasser Khalil, Haoyu Lei, Amir Rasouli, and Yinchuan Li. Two-steps diffusion policy for robotic manipulation via genetic denoising. In *NeurIPS*, 2025.

[12] Apan Dastider, Hao Fang, and Mingjie Lin. Apex: Ambidextrous dual-arm robotic manipulation using collision-free generative diffusion models. In *IROS*, 2024.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[14] Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, A. Jha, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. Torchmetrics - measuring reproducibility in pytorch. *J. Open Source Softw.*, 7:4101, 2022.

[15] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021.

[16] Dave Epstein, Allan Jabri, Ben Poole, Alexei Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. In *NeurIPS*, 2023.

[17] Julius Erbach, Dominik Narnhofer, Andreas Dombos, Bernt Schiele, Jan Eric Lenssen, and Konrad Schindler. Solving inverse problems with FLAIR. In *NeurIPS*, 2025.

[18] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[19] Qian Feng, Hang Li, Zhi Zheng, Jianxiang Feng, and Alois Knoll. Language-guided object-centric diffusion policy for collision-aware robotic manipulation. In *ICRA*, 2025.

[20] Ruiqi Feng, Chenglei Yu, Wenhao Deng, Peiyan Hu, and Tailin Wu. On the guidance of flow matching. In *ICML*, 2025.

[21] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *ICLR*, 2025.

[22] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

[23] Yuxiang Fu, Qi Yan, Lele Wang, Ke Li, and Renjie Liao. Moflow: One-step flow matching for human trajectory forecasting via implicit maximum likelihood estimation based distillation. In *CVPR*, 2025.

[24] Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[25] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

[26] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPSW*, 2021.

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[28] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. In *ICML*, 2023.

[29] Adhithyan Kalaivanan, Zheng Zhao, Jens Sjölund, and Fredrik Lindsten. Ess-flow: Training-free guidance of flow-based models as inference in source space. *arXiv preprint arXiv:2510.05849*, 2025.

[30] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018.

[31] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

[32] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *NeurIPS*, 2024.

[33] Taekyung Ki, Dongchan Min, and Gyeongsu Chae. Float: Generative motion latent flow matching for audio-driven talking portrait. In *ICCV*, 2025.

[34] Pengzhi Li, Qiang Nie, Ying Chen, Xi Jiang, Kai Wu, Yuhuan Lin, Yong Liu, Jinlong Peng, Chengjie Wang, and Feng Zheng. Tuning-free image customization with image and text guidance. In *ECCV*, 2024.

[35] Xingyuan Li, Zirui Wang, Yang Zou, Zhixin Chen, Jun Ma, Zhiying Jiang, Long Ma, and Jinyuan Liu. Difiisr: A diffusion model with gradient guidance for infrared image super-resolution. In *CVPR*, 2025.

[36] Haohong Lin, Xin Huang, Tung Phan, David Hayden, Huan Zhang, Ding Zhao, Siddhartha Srinivasa, Eric Wolff, and Hongge Chen. Causal composition diffusion model for closed-loop traffic generation. In *CVPR*, 2025.

[37] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

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

[39] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[40] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[41] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[42] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[43] Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *ICML*, 2023.

[44] Ségolène Tiffany Martin, Anne Gagneux, Paul Hagemann, and Gabriele Steidl. Pnp-flow: Plug-and-play image restoration with flow matching. In *ICLR*, 2025.

[45] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[46] Jonghyun Park and Jong Chul Ye. Flowlps: Langevin-proximal sampling for flow-based inverse problem solvers. *arXiv preprint arXiv:2512.07150*, 2025.

[47] Maitreya Patel, Song Wen, Dimitris N. Metaxas, and Yezhou Yang. Steering rectified flow models in the vector field for controlled image generation. In *ICLRW*, 2025.

[48] Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, and Sam Devlin. Imitating human behaviour with diffusion models. In *ICLR*, 2023.

[49] Javad Rajabi, Soroush Mehraban, Seyedmorteza Sadat, and Babak Taati. Token perturbation guidance for diffusion models. In *NeurIPS*, 2025.

[50] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *RSS*, 2018.

[51] Martin Rohbeck, Charlotte Bunne, Edward De Brouwer, Jan-Christian Huetter, Anne Biton, Kelvin Y. Chen, Aviv Regev, and Romain Lopez. Modeling complex system dynamics with flow matching across time and conditions. In *ICLR*, 2025.

[52] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, 2015.

[53] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

[54] Johannes Schusterbauer, Ming Gui, Frank Fundel, and Björn Ommer. Diff2flow: Training flow matching models via diffusion model alignment. In *CVPR*, 2025.

[55] Yifei Shen, Xinyang Jiang, Yifan Yang, Yezhen Wang, Dongqi Han, and Dongsheng Li. Understanding and improving training-free loss-based diffusion guidance. In *NeurIPS*, 2024.

[56] Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *ICLR*, 2023.

[57] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[58] George Stoica, Vivek Ramanujan, Xiang Fan, Ali Farhadi, Ranjay Krishna, and Judy Hoffman. Contrastive flow matching. In *ICCV*, 2025.

[59] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, et al. curobo: Parallelized collision-free minimum-jerk robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023.

[60] Giuseppe Tagliavini, Stefan Mach, Davide Rossi, Andrea Marongiu, and Luca Benini. A transprecision floating-point platform for ultra-low power computing. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018.

[61] Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

[62] Feng Tian, Yixuan Li, Weili Zeng, Weitian Zhang, Yichao Yan, and Xiaokang Yang. Stabilizing diffusion posterior sampling by noise–frequency continuation, 2026.

[63] Zhengchao Wan, Qingsong Wang, Gal Mishne, and Yusu Wang. Elucidating flow matching ODE dynamics via data geometry and denoisers. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=f5czhqYK3H.

[64] Sen Wang, Le Wang, Sanping Zhou, Jingyi Tian, Jiayi Li, Haowen Sun, and Wei Tang. Flowram: Grounding flow matching policy with region-aware mamba framework for robotic manipulation. In *CVPR*, 2025.

[65] Xihua Wang, Xin Cheng, Yuyue Wang, Ruihua Song, and Yunfeng Wang. Vaflow: Video-to-audio generation with cross-modality flow matching. In *ICCV*, 2025.

[66] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

[67] Yingzhi Xia, Setthakorn Tanomkiattikun, Liangli Zhen, and Zaiwang Gu. Noise-adaptive diffusion sampling for inverse problems without task-specific tuning, 2026.

[68] Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *CVPR*, 2025.

[69] Rui Heng Yang, Xuan Zhao, Leo Maxime Brunswic, Montgomery Alban, Mateo Clemente, Tongtong Cao, Jun Jin, and Amir Rasouli. Cape: Context-aware diffusion policy via proximal mode expansion for collision avoidance. *arXiv preprint arXiv:2511.22773*, 2025.

[70] Xi Ye, Rui Heng Yang, Jun Jin, Yinchuan Li, and Amir Rasouli. Ra-dp: Rapid adaptive diffusion policy for training-free high-frequency robotics replanning. In *IROS*, 2025.

[71] Xin Ye, Burhaneddin Yaman, Sheng Cheng, Feng Tao, Abhirup Mallik, and Liu Ren. Bevdiffuser: Plug-and-play diffusion model for bev denoising with ground-truth guidance. In *CVPR*, 2025.

[72] Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K Terry. Minari, September 2024. URL https://doi.org/10.5281/zenodo.13767625.

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

[74] Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *ICML*, 2023.

[75] Yinan Zheng, Ruiming Liang, Kexin ZHENG, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Diffusion-based planning for autonomous driving with flexible guidance. In *ICLR*, 2025.

A Proof of the OOD bound

Theorem 2. *Let $(\mu_1, \mu_0, \pi, v^\theta)$ be a flow-matching model and let g be any coupled guidance. Under the assumptions of Theorem 1, with L_g the Lipschitz constant of $v \mapsto v^g$, $L_{\theta,t}$ the Lipschitz constant of v_t^θ and $L_\theta = \exp\left(\int_0^1 L_{\theta,t}\, dt\right)$,*

$$\mathbb{E}_o \mathcal{W}_1(\mu_1^{g(o)}, \mu_1^{\theta,g(o)}) \leq L_g L_\theta\, \mathcal{L}^{1/2}(\theta) \left\| \frac{d\pi^g}{d\pi} \right\|_{L^\infty(\pi)} \tag{15}$$

where π^g is the distribution of coupling (x_0^g, x_1^g) during inference.

Proof. We follow the lines of the proof of Theorem 1 of Benton et al. [3]. Let $X_{s,t}^x$ and $Y_{s,t}^x$ be the flows of the guided fields v^g and $v^{g,\theta}$, respectively: for all $x \in \mathbb{R}^d$ and all $0 \leq s \leq t \leq 1$

$$X_{s,t}^x = x + \int_s^t v^g(X_{s,r}^x, r) dr\ ; \tag{16}$$

$$Y_{s,t}^x = x + \int_s^t v^{g,\theta}(Y_{s,r}^x, r) dr. \tag{17}$$

By the Alekseev–Gröbner Theorem we have

$$X_{s,t}^x - Y_{s,t}^x = \int_s^t \left(\nabla_x Y_{r,t}^{X_{s,r}^x}\right) (v^g(X_{s,r}^x, r) - v^{g,\theta}(X_{s,r}^x, r)) dr. \tag{18}$$

Furthermore,

$$\nabla_x Y_{s,t}^x = I + \int_s^t \left(\nabla_x v^{g,\theta}(Y_{s,r}^x, r)\right) \nabla_x Y_{s,r}^x dr. \tag{19}$$

It follows that by the Gronwall Lemma,

$$\|\nabla_x Y_{s,t}^x\|_{\mathrm{op}} \leq \exp\left(\int_0^1 L_{\theta,r}\, dr\right) = L_\theta. \tag{20}$$

Hence, we get:

$$\begin{aligned} \|X_{s,t}^x - Y_{s,t}^x\| &\leq \int_s^t \left\|\nabla_x Y_{r,t}^{X_{s,r}^x}\right\|_{\mathrm{op}} \\ &\qquad \times \|v^g(X_{s,r}^x, r) - v^{g,\theta}(X_{s,r}^x, r)\| dr. \\ &\leq L_\theta \int_s^t \|v^g(X_{s,r}^x, r) - v^{g,\theta}(X_{s,r}^x, r)\| dr. \end{aligned} \tag{21}$$

Taking the expectation for $x \sim \mu_0$ and observation o, and writing $I_t := (1-t)x_0 + tx_1$ for the interpolant, we have

$$
\begin{aligned}
\mathbb{E}_o \mathbb{E}_x \|X^x_{0,1} - Y^x_{0,1}\| &\leq L_\theta \mathbb{E}_o \mathbb{E}_x \int_0^1 \left\| v^g(X^x_{0,r}, r) - v^{g,\theta}(X^x_{0,r}, r) \right\| dr \\
\mathbb{E}_o \mathcal{W}_1(\mu_1^{g(o)}, \mu_1^{g(o),\theta}) &\leq L_\theta \mathbb{E}_o \mathbb{E}_{t\sim\mathcal{U},(x_0,x_1)\sim(\pi^g|t,o)} \left\| v^g(I_t,t) - v^{g,\theta}(I_t,t) \right\| \\
&\leq L_\theta \mathbb{E}_o \mathbb{E}_{t\sim\mathcal{U},(x_0,x_1)\sim(\pi|t,o)} \frac{d(\pi^g|t,o)}{d(\pi|t,o)} \left\| v^g(I_t,t) - v^{g,\theta}(I_t,t) \right\| \\
&\leq L_\theta \mathbb{E}_o \mathbb{E}_{t\sim\mathcal{U},(x_0,x_1)\sim\pi} \frac{d(\pi^g|t,o)}{d\pi} \left\| v^g(I_t,t) - v^{g,\theta}(I_t,t) \right\| \\
&\leq L_g L_\theta \mathbb{E}_{t\sim\mathcal{U},(x_0,x_1)\sim\pi} \left[\mathbb{E}_o \frac{d(\pi^g|t,o)}{d\pi} \right] \left\| v(I_t,t) - v^\theta(I_t,t) \right\| \\
&\leq L_g L_\theta \mathbb{E}_{t\sim\mathcal{U},(x_0,x_1)\sim\pi} \frac{d(\pi^g|t)}{d\pi} \left\| v(I_t,t) - v^\theta(I_t,t) \right\| \\
&\leq L_g L_\theta \left(\mathbb{E}_{t\sim\mathcal{U},(x_0,x_1)\sim\pi} \left(\frac{d(\pi^g|t)}{d\pi} \right)^2 \right)^{1/2} \\
&\qquad \times \left(\mathbb{E}_{t\sim\mathcal{U},(x_0,x_1)\sim\pi} \left\| v(I_t,t) - v^\theta(I_t,t) \right\|^2 \right)^{1/2} \\
&\leq L_g L_\theta \left\| \frac{d\pi^g}{d\pi} \right\|_{L^\infty(\pi)} \sqrt{\mathcal{L}(\theta)}
\end{aligned}
$$

□

B Experimental details

B.1 Baseline selection

As stated in the main text, the baselines for training-free guidance are scarce. We took Feng et al. [20]'s five variants as our starting point and considered each carefully.

First, g^{MC} requires a large number of parallel samples per step, which makes it prohibitively expensive for our image experiments, an issue which the original paper also faces. On the lower-dimensional Adroit tasks it is tractable, and we report it in Appendix D.4. For most of our experiments (super-resolution and inpainting and our guidance sparsity experiments), g^{cov-G} and g^{cov-A} reduce analytically to ΠGDM and $g^{\text{sim-inv-A}}$ (which we refer to as ΠGDM+) respectively. We hence choose to benchmark our method on these two baselines as they cover all 4 tractable methods from Feng et al. [20].

This setup also covers FlowChef [47], which gives proportional guidance to g^{cov-A}, and in turn ΠGDM+, with a factor that is absorbed during our tuning. We additionally report PnP-Flow [44] as an external reference point (Appendices D.4 and D.5). The remaining training-free guidance works [17, 46, 29] operate exclusively in latent space, rely on noise-level terms incompatible with noiseless tasks, or do not scale to high dimension tasks; none of them are directly applicable here.

For the baseline choice of collision avoidance, we cannot directly use ΠGDM , as it would require finding an *adjoint operator*. Concretely, this means that we would need to find a function h such that, with p the collision avoidance pipeline, $p \circ h \circ p(x) = p(x)$. This means that given the environment and the value of the Euclidean Signed Distance Function, we need to be able to find a set of joint positions that solves to the same ESDF. This is highly non trivial, and if implemented would have a prohibitive cost to use in a robotics setting. Since basic energy minimization does not have such a constraint on the guidance operator, we use it in its simplest form :

$$g_{\text{collision}} = -\lambda_t \nabla_{\widehat{x}_1} ||\widehat{x}_1 - y||$$

B.2 Robotics

For the robotics tasks we use the same UNet architecture as in Chi et al. [9], with 3 levels of depth respectively 256, 512 and 1024. They are trained with a time resolution of $1/128$ for 1000 epochs

on the "expert-v2" datasets available in the Minari python library [72]. The reported scores use 300 environment seeds.

B.3 Image Generation

For the image generation tasks, a 2 billion parameter UNet is used as defined in the code provided by Feng et al. [20], with minor fixes to the training code to make it more stable. They are trained using a batch of size 128, base learning rate 10^{-4} with AdamW [42], warm-up then cosine annealing [41]. For both training and inference we tested and found negligible performance impacts for running the model in mixed precision: the training is done using torch automatic mixed precision, and all methods use BF16 [60] for the reported scores. We noticed an improvement in base model performance when normalizing data which was not done in Feng et al. [20]'s code. The models are hence trained with normalized data, but we noticed that both ΠGDM and $g^{\text{sim-inv-A}}$ suffered from doing guidance in normalized space with guidance scale varying across two orders of magnitude from task to task. To fairly evaluate the baselines, the model is called in normalized space but the velocity field is then unnormalized before applying guidance:

$$v_t = \sigma v_\theta \left(\frac{x_{raw} - \mu}{\sigma}, t \right) \tag{22}$$

where μ and σ are the dataset's mean and standard deviation. Finally, all image metrics are calculated using torchmetrics [14].

C Additional results

C.1 Robotics

In the robotics setting, our method improves robustness to sparsity for ΠGDM and ΠGDM$_+$in a similar way, for all 4 tasks.

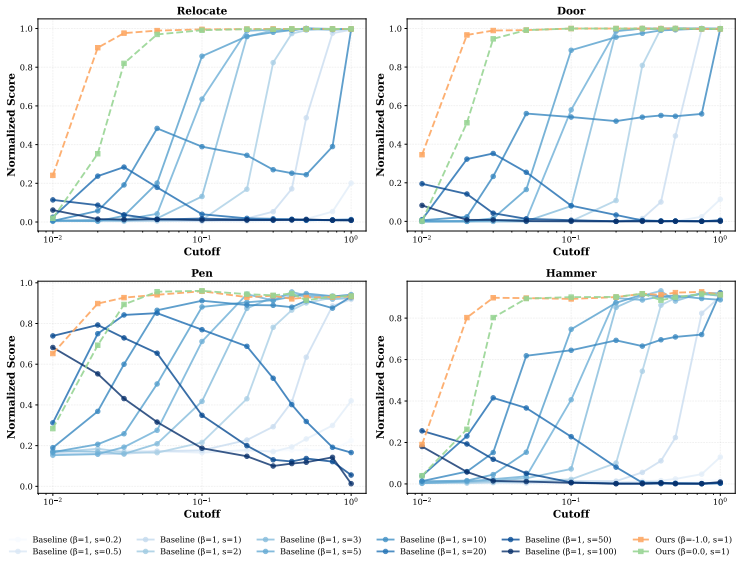


Figure 6: Sparsity tests for all Adroit tasks using ΠGDM$_+$: the cutoff represents the proportion of guided coordinates. Different shades of blue represent different tunings of the guidance scale for the naive guidance runs ($\beta = 1$)

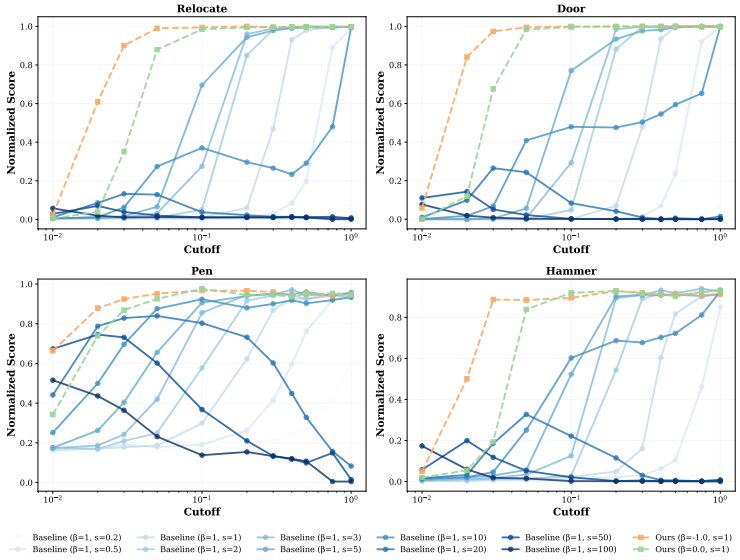


Figure 7: Sparsity tests for all Adroit tasks using ΠGDM : the cutoffs represent the proportion of guided coordinates. Different shades of blue represent different tunings of the guidance scale for the naive guidance runs ($\beta = 1$)

C.2 Images

All image to image scores (LPIPS, PSNR, SSIM) behave in similar ways for different tunings and β values. The plots look slightly different for the FID scores which can be expected: very strong guidances lead to divergence and artifacts, so the strong guidance cliff is the same. However in the low guidance regions, the model is basically not guided, leading to high quality samples, hence giving a decent FID score.

Table 3: Comparison of naive vs. coupled guidance across methods and tasks on CelebA-HQ-256.

		Naive		**Coupled**	
Task	**Metric**	ΠGDM	ΠGDM+	ΠGDM	ΠGDM+
Inpainting	FID↓	54.3	**7.5**	20.7	**7.5**
	LPIPS↓	0.184	**0.041**	0.100	**0.041**
	PSNR↑	20.5	29.3	**30.1**	29.1
	SSIM↑	0.873	**0.953**	0.927	**0.953**
Deblurring	FID↓	54.0	**22.7**	33.0	**22.7**
	LPIPS↓	0.462	0.153	0.209	**0.148**
	PSNR↑	20.1	30.0	28.6	**30.6**
	SSIM↑	0.472	0.830	0.784	**0.840**
Super-res.	FID↓	55.9	76.9	**25.6**	77.0
	LPIPS↓	0.502	0.486	**0.166**	0.431
	PSNR↑	18.1	19.6	**28.8**	19.3
	SSIM↑	0.449	0.443	**0.833**	0.459

Table 4: Comparison of naive vs. coupled guidance across methods and tasks on ImageNet (CFM).

		Naive		**Coupled**	
Task	**Metric**	ΠGDM	ΠGDM+	ΠGDM	ΠGDM+
Inpainting	FID↓	13.0	10.5	**8.8**	11.3
	LPIPS↓	0.106	**0.086**	0.092	0.088
	PSNR↑	20.4	20.5	**23.4**	20.6
	SSIM↑	0.907	**0.911**	0.908	0.909
Deblurring	FID↓	58.6	1.6	16.6	**1.2**
	LPIPS↓	0.452	0.029	0.232	**0.021**
	PSNR↑	19.4	32.9	22.9	**34.0**
	SSIM↑	0.474	0.969	0.749	**0.976**
Super-res.	FID↓	20.7	49.3	**8.9**	50.6
	LPIPS↓	0.230	0.411	**0.150**	0.418
	PSNR↑	21.8	17.2	**23.9**	17.2
	SSIM↑	0.741	0.458	**0.811**	0.445

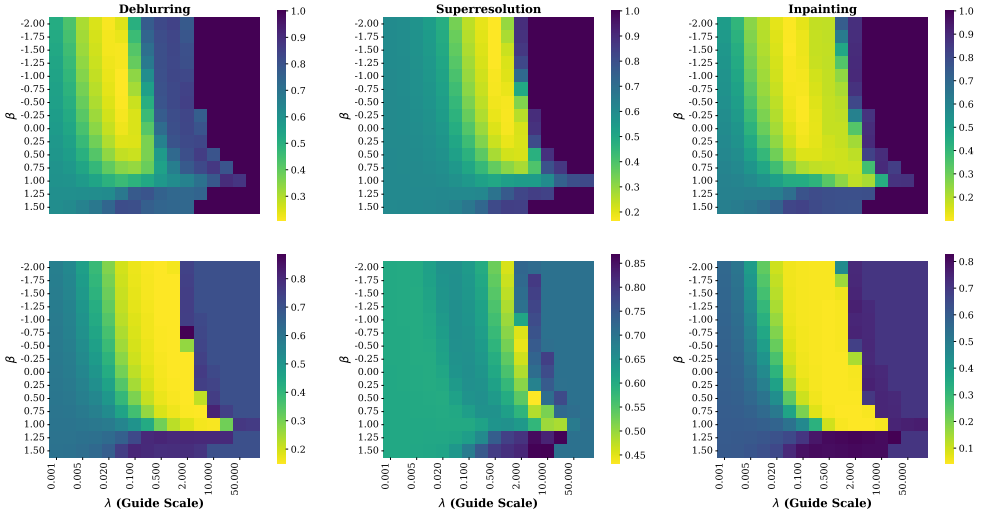


Figure 8: Sample fidelity using different tuning scales and β values for deblurring, superresolution and inpainting (left to right). First row shows the LPIPS scores for ΠGDM , second row shows the performance of ΠGDM_+.

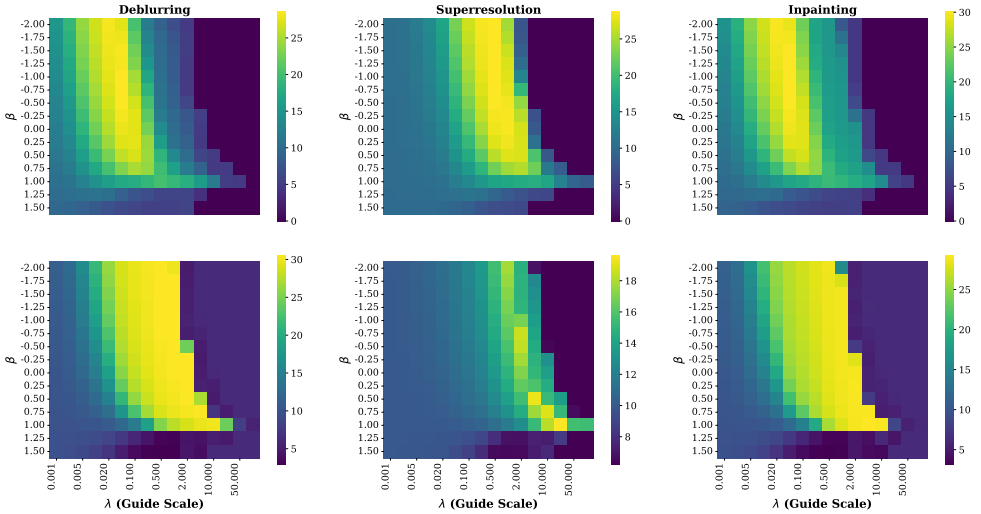


Figure 9: Sample fidelity using different tuning scales and β values for deblurring, superresolution and inpainting (left to right). First row shows the PSNR scores for ΠGDM , second row shows the performance of ΠGDM_+.

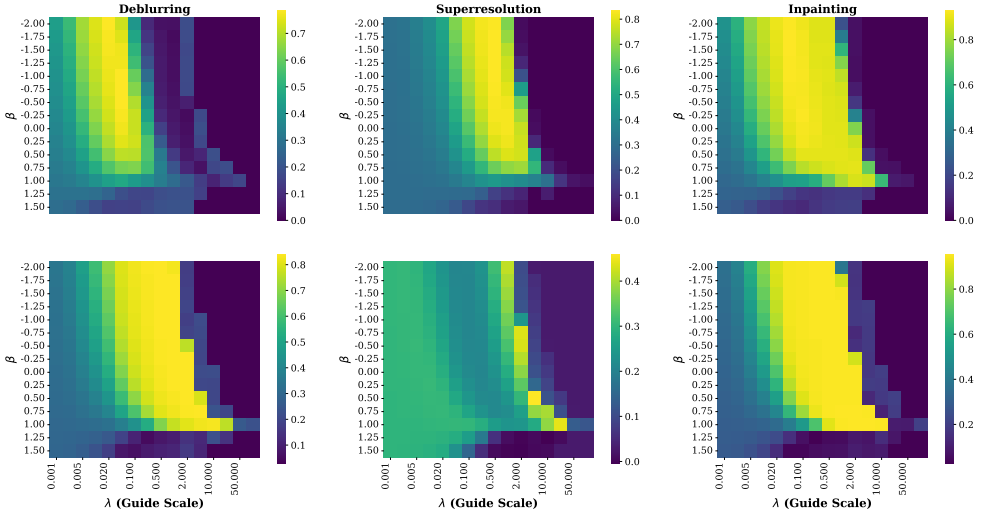


Figure 10: Sample fidelity using different tuning scales and β values for deblurring, superresolution and inpainting (left to right). First row shows the SSIM scores for ΠGDM , second row shows the performance of ΠGDM$_+$.

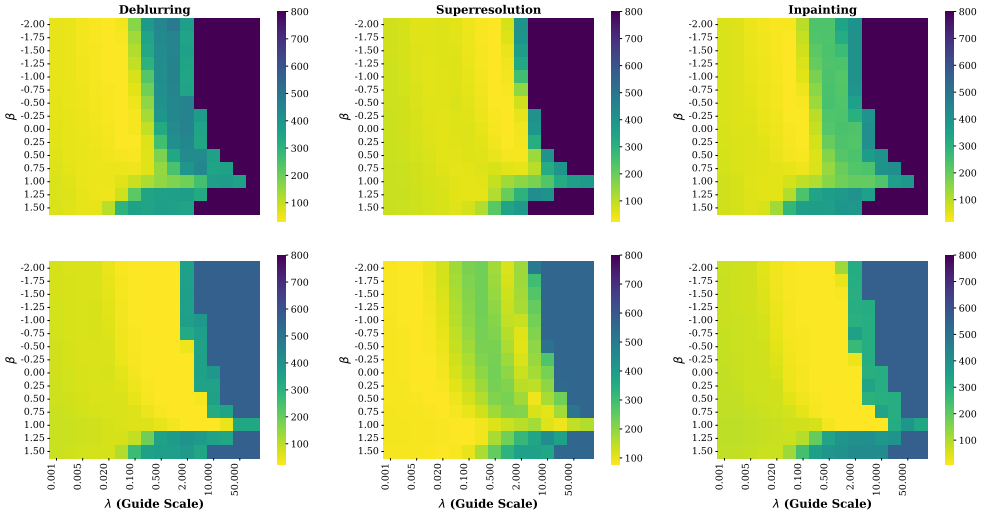


Figure 11: Sample fidelity using different tuning scales and β values for deblurring, superresolution and inpainting (left to right). First row shows the FID scores for ΠGDM , second row shows the performance of ΠGDM$_+$.

We also provide visualizations for both $\widehat{x}_1$ through inference and final samples for all tasks and a range of beta values. These samples show the difference in convergence between naive and coupled guidance and help understand where the artifacts created by naive guidance come from.



Figure 12: $\widehat{x}_1$ predictions using ΠGDM for the first 10 steps of deblurring showing the convergence of the guided process for $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from top to bottom)



Figure 13: $\widehat{x}_1$ predictions using ΠGDM for the first 10 steps of inpainting showing the convergence of the guided process for $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from top to bottom)



Figure 14: $\widehat{x}_1$ predictions using ΠGDM for the first 10 steps of superresolution showing the convergence of the guided process for $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from top to bottom)



Figure 15: $\widehat{x}_1$ predictions using ΠGDM$_+$ for the first 10 steps of deblurring showing the convergence of the guided process for $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from top to bottom)



Figure 16: $\widehat{x}_1$ predictions using ΠGDM$_+$ for the first 10 steps of inpainting showing the convergence of the guided process for $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from top to bottom)

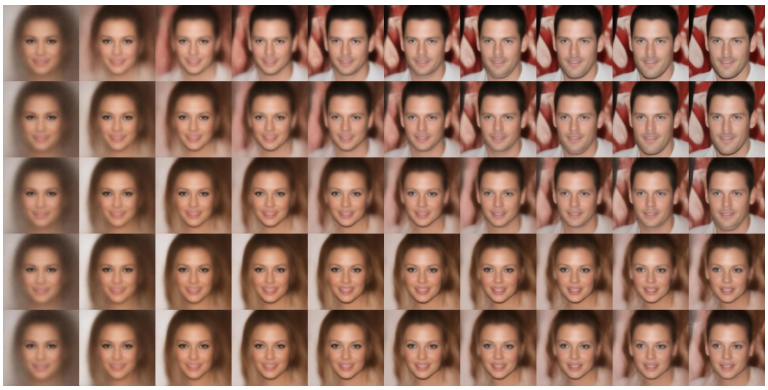


Figure 17: $\widehat{x}_1$ predictions using ΠGDM$_+$ for the first 10 steps of superresolution showing the convergence of the guided process for $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from top to bottom)



Figure 18: Final samples for deblurring using ΠGDM with $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from left to right)



Figure 19: Final samples for inpainting using ΠGDM with $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from left to right)

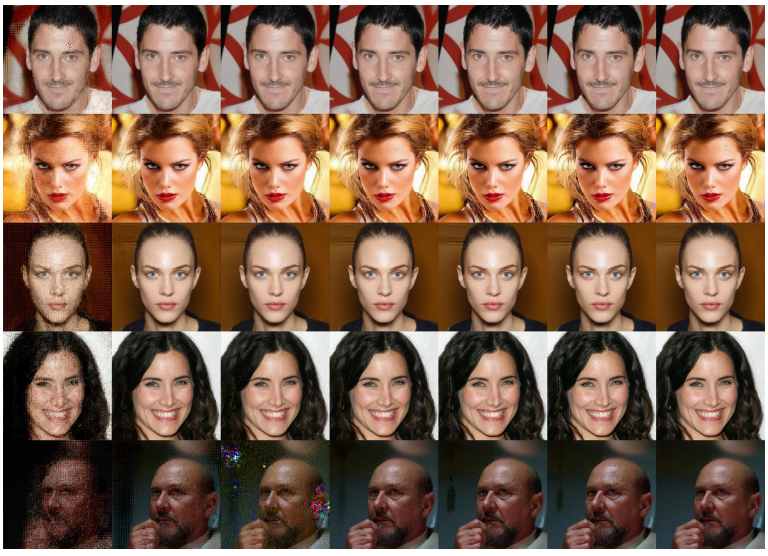


Figure 20: Final samples for superresolution using ΠGDM with $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from left to right)



Figure 21: Final samples for deblurring using ΠGDM$_+$with $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from left to right)



Figure 22: Final samples for inpainting using ΠGDM$_+$with $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from left to right)



Figure 23: Final samples for superresolution using ΠGDM$_+$with $\beta \in [1, 0.75, 0.5, 0.25, 0]$ (from left to right)

D Additional experiments

This appendix reports experiments carried out during the review process. In all tables, "tuned" arms are evaluated at their best guidance scale, and W–L counts the paired cells (identical seeds and noise draws) won and lost by coupled guidance against tuned naive guidance.

D.1 Coupling and prior variants

Coupled guidance does not rely on minibatch OT couplings. We retrain the Adroit models with different couplings and priors (3 training seeds per variant) and compare tuned naive guidance with coupled guidance for $\beta \in \{0, -1\}$ (Table 5). The ordering $\beta \in \{0, -1\} \geq \beta = 1$ holds for every variant, and no variant requires a different default for β. The guidance scale is tuned per arm: the arms accumulate guidance differently over the trajectory, so their optimal scales differ by construction.

Table 5: Adroit dense reward under different couplings and priors, with paired win–loss counts. The Gaussian-mixture prior is evaluated at its own tuned guidance scale over the 10–25% densities where it functions (at 5% both arms are at floor).

Coupling / prior	Tuned naive	Coupled $\beta = 0$ (W–L)	Coupled $\beta = -1$ (W–L)
Gaussian, independent	31.7	32.8 (121–35)	**33.4** (114–42)
Gaussian, minibatch OT	35.9	35.6 (32–16)	**36.9** (33–15)
Uniform, independent	35.2	35.2 (31–17)	**36.6** (32–16)
Models of Section 5	40.1	42.3 (77–19)	**42.6** (86–23)
Gaussian mixture, 8 modes	18.5	**21.0** (20–4)	20.6 (18–6)

D.2 Sparsity protocols and guidance-scale robustness

In Section 5, guided entries are selected elementwise and uniformly at random. To mirror hierarchical planners more closely, we also evaluate temporal protocols on the sparse cells where sparsity is at stake (elementwise density $\leq 10\%$, $m \leq 2$ waypoints out of 16, blocks at 5% density), with both arms at their tuned scales (Table 6). Across the five protocols, coupled guidance improves dense reward by 7–11% over fully tuned naive guidance and wins 78–88% of the paired cells.

Table 6: Adroit dense reward for different sparsity protocols.

Sparsity protocol	Tuned naive	Coupled $\beta = -1$	W–L
Elementwise, resampled (Section 5)	38.4	**42.7**	7–1
Elementwise, fixed mask	27.9	**29.8**	70–19
Periodic temporal waypoints	34.8	**38.0**	7–1
Random temporal waypoints	33.8	**37.2**	16–4
Contiguous temporal blocks	38.2	**42.5**	21–3

Robustness to sparsity also shows in the tuning. Re-tuning both arms at every density, the optimal guidance scale g^* of coupled guidance does not move, whereas that of naive guidance changes with density (Table 7): naive guidance must be re-tuned per sparsity level, while coupled guidance works at a single scale and wins where guidance is sparse.

Table 7: Optimal guidance scale g^* per guidance density, and dense-reward margin of coupled over naive guidance when both are at their optimum.

Guidance density	5%	10%	25%
Naive g^*	50	25	25
Coupled g^*	2	2	2
Margin at both optima	$+3.5$	$+4.0$	0.0

D.3 Distance to the training coupling

To measure directly how far guidance moves the coupling, we train a binary classifier to distinguish the endpoint pairs $(\widehat{x}_0, \widehat{x}_1)$ produced under guidance from pairs drawn from the model's training coupling on Adroit. Its test AUC measures the deviation of the guided coupling from the training coupling (0.5: indistinguishable); an unguided control gives 0.5005. At 5% guidance density with $k = 100$ sampling steps (24 cells per arm), coupled guidance stays close to this floor (AUC 0.527 for $\beta = -1$ and 0.542 for $\beta = 0$), whereas naive guidance reaches 0.685: measured as excess over chance, naive guidance distorts the coupling 4–$7\times$ more than coupled guidance. Consistently, the endpoint inconsistency of the reconstructed latent $\widehat{x}_0$, a latent-side symptom of a growing density ratio $d\pi^g/d\pi$, fluctuates about twice as much under naive guidance as under coupled guidance at 5–10% density and matched scale (4.97×10^{-4} vs. 2.3–2.7×10^{-4}).

D.4 Monte-Carlo guidance on Adroit

While g^{MC} is too costly for our image experiments, it is tractable on the Adroit tasks ($d = 1152$). We tune it on both of its hyperparameters, the energy scale (up to 10^5) and its internal step size, and report the best setting over the sample counts we tried. We also include PnP-Flow [44] with its published settings (Table 8). Once tuned, g^{MC} is a genuine baseline, and it also benefits from the coupled update; coupled ΠGDM remains the strongest arm on three of the four tasks.

Table 8: Adroit dense reward per task at 5% guidance density and $k = 100$ sampling steps (approximate task ceilings: 20 / 92 / 53 / 31). Guidance arms are scale-tuned.

Arm	Door	Hammer	Pen	Relocate
Unguided	-0.2	0.1	7.7	0.1
PnP-Flow (published settings)	-0.2	-1.1	6.0	0.1
g^{MC}, tuned	5.7	35.7	21.9	0.8
g^{MC} + coupled update	7.2	43.6	20.8	2.1
Naive ΠGDM ($\beta = 1$), tuned	**12.4**	48.1	33.4	5.8
Coupled ΠGDM ($\beta = -1$)	12.0	**54.1**	**40.8**	**6.8**

D.5 Class-conditional ImageNet-128

We train a class-conditional version of the ImageNet-128 model, with the same architecture and training recipe as in Appendix B.3 and an independent coupling, and evaluate the three inverse problems with 10,240 images per configuration. Table 9 reports paired differences between coupled and naive guidance: coupled guidance improves on every task and method, except for ΠGDM$_+$ on inpainting, where both give very similar results. Table 10 compares each arm at its best cell of the same 16-scale ladder, together with PnP-Flow as an external reference point; FID is computed at the set level, the other metrics are per-image means. PnP-Flow collapses on all three tasks.

Table 9: Class-conditional ImageNet-128: paired differences (coupled − naive), 10,240 images per configuration. Win rate: fraction of images where coupled guidance has the higher PSNR.

Task	Method	ΔPSNR (dB ± SE)	ΔLPIPS	PSNR win rate (%)
Deblurring	ΠGDM	$+4.00 \pm 0.02$	-0.187	98.1
	ΠGDM+	$+0.72 \pm 0.01$	-0.021	94.6
Inpainting	ΠGDM	$+1.36 \pm 0.06$	-0.035	61.9
	ΠGDM+	-0.25 ± 0.02	$+0.003$	54.5
Super-res.	ΠGDM	$+4.47 \pm 0.04$	-0.120	98.8
	ΠGDM+	$+0.14 \pm 0.00$	-0.004	73.5

Text-guided attribute control. We also apply CLIP-ensemble guidance towards the attributes "in the snow" and "red" on the class-conditional model (10,000 images per cell, scored by a held-out CLIP ViT-L/14). At matched attribute score, coupled guidance lowers FID relative to tuned naive guidance by 0.9 on average for "in the snow" (24.7 at the best point) and by 1.1 on average for

Table 10: Class-conditional ImageNet-128: each arm at its best cell (10,240 images per cell).

Task	Arm (best cell)	FID↓	LPIPS↓	PSNR↑	SSIM↑
Deblurring	PnP-Flow	179.2	0.673	11.0	0.104
	Naive ΠGDM+	10.3	0.202	23.4	0.738
	Coupled ΠGDM+	**9.0**	**0.181**	**24.1**	**0.763**
Inpainting	PnP-Flow	32.1	0.128	19.7	0.903
	Naive ΠGDM+	**6.2**	0.083	20.8	0.912
	Coupled ΠGDM	6.7	**0.082**	**22.1**	**0.919**
Super-res.	PnP-Flow	92.2	0.570	15.0	0.206
	Naive ΠGDM	24.5	0.269	19.9	0.697
	Coupled ΠGDM	**5.4**	**0.148**	**23.9**	**0.811**

"red" (partial scale ladder). These gains are modest, consistent with the smaller effect of coupling on Jacobian-free guidance.

D.6 Noisy observations

We repeat ImageNet-128 deblurring with observation noise $\sigma_y \in \{0.05, 0.1, 0.2\}$, paired on identical images and noise draws, with per-arm tuned guidance scales, and report g^{cov-G} and g^{cov-A} alongside ΠGDM and ΠGDM$_+$(Table 11). The split follows whether the guidance uses the Jacobian $\partial\widehat{x}_1/\partial x_t$, not the method family: Jacobian-based guidance retains a $+2.6$ to $+3.4$ dB advantage when coupled across the whole noise range, while the gains of Jacobian-free guidance shrink to between -0.9 and $+0.6$ dB once $\sigma_y > 0$. For the ΠGDM rows we swept the guidance scale but not the residual schedule r_t, so these rows are indicative; $\sigma_y = 0.2$ was not run for them.

Table 11: ImageNet-128 deblurring with observation noise: ΔPSNR (dB, coupled − naive), per-arm tuned scales.

Guidance	Uses $\partial\widehat{x}_1/\partial x_t$	$\sigma_y = 0$	0.05	0.1	0.2
ΠGDM	yes	$+3.40$	$+2.99$	$+2.69$	—
ΠGDM+	no	$+0.60$	-0.87	-0.78	—
g^{cov-G}	yes	$+2.99$	$+2.96$	$+2.88$	$+2.58$
g^{cov-A}	no	$+1.25$	$+0.58$	$+0.63$	$+0.24$

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims in the abstract and introduction are both supported by the theoretical results in appendix A and section 4 and the empirical validation in section 5.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We add a dedicated limitations section, and also discuss limitations directly in the experiment section.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: All assumptions are directly given in the main text, while the proof is provided in appendix A.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We provide the training hyperparameters, inference hyperparameters, architecture choices along with a pseudo-code translating exactly the code used in our experiments.

Guidelines:

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

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

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

   Answer: [No]

   Justification: While we provide the information to reproduce our experiments and only use open-source datasets, we do not provide the experiment code directly.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All our experiments are standard tasks and / or based on open source code from previous work, and any changes or additional experiments are fully described in the text.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: We do not report error bars in the main tables and figures. We report the number of evaluation seeds for each experiment (256 environment seeds for collision avoidance, 300 seeds for the Adroit tasks); robotics metrics are success rates or rewards averaged over these randomized environments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

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

8. **Experiments compute resources**

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

   Answer: [N/A]

   Justification: Our method is inference based and adds no overhead in terms of compute, keeping the generation the same cost as announced by the previous work chosen for the baselines. For the additional robotics experiments, the compute requirements are provided in the experimental details.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: No human subjects or potential harms are treated in our work.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [No]

Justification: We do not include a broader-impacts section. Our contribution is a sampling-time correction that improves the stability of existing guidance methods; it does not introduce new capabilities or risks beyond those of the underlying generative models and guidance methods (e.g., image restoration techniques can in general be misused for image manipulation).

Guidelines:

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

11. **Safeguards**

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

Answer: [N/A].

Justification: As part of this work we are not releasing any new code or data. Our approach is merely in place for improving existing methods and it does not directly impose any risks.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The datasets, robotics environments and the paper from which we took the code to implement our method are all duly cited.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: No new assets are introduced.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No crowdsourcing or human subjects are part of this work.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: No crowdsourcing or human subjects are part of this work.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are not an important part of the method or anywhere in the work.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.