Conditional Diffusion with Less Explicit Guidance via Model Predictive Control

Anonymous Author(s)
Affiliation
Address
email

Abstract

How much explicit guidance is necessary for conditional diffusion? We consider the problem of conditional sampling using an unconditional diffusion model and limited explicit guidance (e.g., a noised classifier, or a conditional diffusion model) that is restricted to a small number of time steps. We explore a model predictive control (MPC)-like approach to approximate guidance by simulating unconditional diffusion forward, and backpropagating explicit guidance feedback. MPC-approximated guides have high cosine similarity to real guides, even over large simulation distances. Adding MPC steps improves generative quality when explicit guidance is limited to five time steps.

Five time steps with explicit guidance | + MPC | Reference

1 Introduction

Diffusion models are a class of generative models that have achieved remarkable sample quality, particularly for text-to-image generation (1), where diffusion has been guided using *classifier guidance* or *classifier-free guidance* to sample images $\mathbf{x} \sim p(\mathbf{x}|\mathbf{c})$ for a conditioning variable $\mathbf{c}$ (e.g., text) (2; 3). Controlling generative models is important for applications such as text generation and drug discovery, where multiple distinct conditional variables $\mathbf{c}_1, \mathbf{c}_2, ...\mathbf{c}_n$ can be important: e.g., drug activity and permeability (4).

For each new conditioning information source $\mathbf{c}$ of interest, classifier guidance and classifier-free guidance require training a new explicit guidance model over all diffusion time steps $t \in [0, ..., T]$ (often, T =100 to 1,000), and sample using the explicit guide at each generative time step (often, 25-100) (2; 3). Here, we explore whether conditional sampling is achievable without explicit guidance at every generative step, and if it is achievable with very few steps. This line of inquiry may make it easier to condition on new variables by reducing the training burden of new explicit guidance models.

Rejection sampling and Langevin "churning" have been explored for image editing, inpainting, and conditional sampling on new variables without training a new model over diffusion time steps, but lack general applicability (5; 1; 6; 7; 8; 9; 10): churning appears limited to "local" edits, while rejection sampling is inefficient for rare events. Separately, scheduler advances have reduced sampling steps from 100-1000 to 25-50 while retaining high sample quality (11; 12). This work aims to be generally applicable and synergistic with scheduler improvements.

Under review at the NeurIPS 2022 Workshop on Score-Based Methods. Do not distribute.

Diffusion models. Diffusion models are trained on noise-corrupted data, and learn an iterative denoising process to generate samples. We give a non-precise introduction following (13), and refer interested readers to (11) for a precise description. A diffusion model $\hat{\mathbf{x}}_\theta$ is trained to optimize:

$$\mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},t}[w_t\|\hat{\mathbf{x}}_\theta(\alpha_t\mathbf{x}+\sigma_t\boldsymbol{\epsilon},\mathbf{c})-\mathbf{x}\|_2^2] \tag{1}$$

where $(\mathbf{x},\mathbf{c})$ are data-conditioning pairs, $t\sim\mathcal{U}([0,1])$, $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$, and α_t,σ_t, and w_t are time-varying weights that influence sample quality. In the $\boldsymbol{\epsilon}$-prediction parameterization, $\hat{\mathbf{x}}_\theta(\mathbf{z}_t,\mathbf{c})=(\mathbf{z}_t-\sigma_t\boldsymbol{\epsilon}_\theta(\mathbf{z}_t,\mathbf{c}))/\alpha_t$ where $\boldsymbol{\epsilon}_\theta$ is the learned function. Notably, this training procedure has an expectation over t, which can be hundreds to thousands of time steps.

To sample, a simple scheduler starts at $\mathbf{z}_T\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ and iteratively generates $\mathbf{z}_{t-1}=(\mathbf{z}_t-\sigma\tilde{\boldsymbol{\epsilon}}_\theta)/\alpha_t$ where the choice of $\tilde{\boldsymbol{\epsilon}}_\theta$ distinguishes sampling strategies. In general, schedulers can jump to $\mathbf{z}_{t-\Delta}$ as a function of starting time t, jump size Δ, latent $\mathbf{z}_t$, and predicted noise $\tilde{\boldsymbol{\epsilon}}_\theta$.

Diffusion guidance. Classifier guidance (2) requires training a *noised classifier* $p_t(\mathbf{c}|\mathbf{z}_t)$ over T time steps, and uses $\tilde{\boldsymbol{\epsilon}}_\theta=\boldsymbol{\epsilon}(\mathbf{z}_t,\mathbf{c})-\nabla_{\mathbf{z}_t}\log p_t(\mathbf{c}|\mathbf{z}_t)$. Notably, pre-trained *clean-data classifiers* cannot be directly used for guidance. Classifier-free guidance (3) learns both a conditional and unconditional diffusion model by setting $\mathbf{c}=\mathbf{0}$ with 10% probability during training; $\tilde{\boldsymbol{\epsilon}}_\theta=\boldsymbol{\epsilon}_\theta(\mathbf{z}_t):=\boldsymbol{\epsilon}_\theta(\mathbf{z}_t,\mathbf{c}=\mathbf{0})$ achieves unconditional sampling. Classifier-free guidance with weight w uses

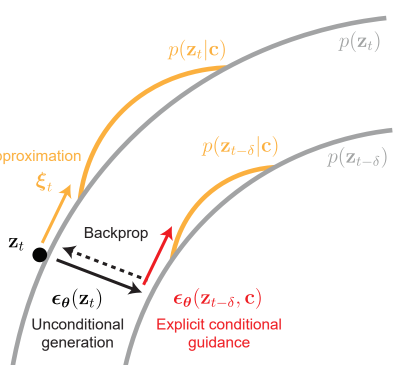
$$\tilde{\boldsymbol{\epsilon}}_\theta=(1+w)\boldsymbol{\epsilon}(\mathbf{z}_t,\mathbf{c})-w\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t). \tag{2}$$

Model predictive control (MPC). Model predictive control aims at controlling a time-evolving system in an optimized manner, by using a predictive dynamics model of the system and solving an optimization problem online to obtain a sequence of *control actions*. Typically, the first control action is applied at the current time, then the optimization problem is solved again to act at the next time step (14). The general formalized MPC problem is:

$$\arg\min_{\mathbf{s}_{1:T},\mathbf{a}_{1:T}}\sum_{t=1}^{T}\ell_t(\mathbf{s}_t,\mathbf{a}_t)\text{ subject to }\mathbf{s}_{t+1}=f(\mathbf{s}_t,\mathbf{a}_t);\mathbf{s}_1=\mathbf{s}_{\texttt{init}} \tag{3}$$

where $\mathbf{s}_t,\mathbf{a}_t$ are the state and control action at time t, ℓ_t is a cost function, f is a dynamics model, and $\mathbf{s}_{\texttt{init}}$ is the initial state of the system. MPC can be solved with gradient methods (15; 16).

2 Approximate conditional guidance via model predictive control



Our problem is performing conditional diffusion on a latent $\mathbf{z}_t$ with only access to an unconditional diffusion model. In particular, we do not have an explicit conditional guide $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t,\mathbf{c})$ at time t; instead, we can evaluate it only at $t-\delta$. Our method, MPC guidance, optimizes an approximation $\boldsymbol{\xi}_t\approx\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t,\mathbf{c})$, which is used in classifier-free guidance (eq. 2) to apply one generative step on $\mathbf{z}_t$ to obtain $\mathbf{z}_{t-\Delta}$. This can be applied repeatedly to reach $\mathbf{z}_0$. In terms of MPC, we view $\mathbf{z}_t$ as states, control actions as $\tilde{\boldsymbol{\epsilon}}_\theta$, the dynamics model f as the diffusion generative process given $\mathbf{z}_t$ and $\tilde{\boldsymbol{\epsilon}}_\theta$, and define loss ℓ at time $t-\delta$ using the explicit guide (Fig. 2).

Noised classifier. With a noised classifier $p_{t-\delta}(\mathbf{c}|\mathbf{z}_t)$, the explicit guide $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_{t-\delta},\mathbf{c})=\nabla_{\mathbf{z}_{t-\delta}}\log p_{t-\delta}(\mathbf{c}|\mathbf{z}_{t-\delta})$. We propose to unconditionally generate $\mathbf{z}_{t-\delta}$ from $\mathbf{z}_t$ and evaluate $\log p_{t-\delta}(\mathbf{c}|\mathbf{z}_{t-\delta})$ which we treat as "inverse loss". Our MPC guide at time t is a first-order, one-step optimization of this loss:

$$\boldsymbol{\xi}_t=-\nabla_{\mathbf{z}_t}\ell(\mathbf{z}_{t-\delta})=-\nabla_{\mathbf{z}_t}\log p_{t-\delta}(\mathbf{c}|\mathbf{z}_{t-\delta}) \tag{4}$$

Conditional diffusion model. When the explicit guide is a conditional diffusion model $\boldsymbol{\epsilon}_\theta(\mathbf{z}_{t-\delta},\mathbf{c})$, we denoise $\mathbf{z}_t$ to $\mathbf{z}_{t-\delta}$ and construct the MPC guide as:

$$\boldsymbol{\xi}_t=-\nabla_{\mathbf{z}_t}\ell(\mathbf{z}_{t-\delta})=-\nabla_{\mathbf{z}_t}\|\mathbf{z}_{t-\delta}-\mathbf{z}^*\|^2 \tag{5}$$

where gradients with respect to $\mathbf{z}_t$ are blocked for the target $\mathbf{z}^*:=\mathbf{z}_{t-\delta}+\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_{t-\delta},\mathbf{c})$.

Algorithm 1: Approximate guide with noised classifier

```
def approx_guide(zt, t, dt, noised_classifier):
    z = denoise(zt, t, dt)     # differentiable; denoise zt to time t-dt
    return autograd(noised_classifier(z), zt)    # grad wrt zt
```

Algorithm 2: Approximate guide with conditional diffusion model

```
def approx_guide(zt, t, dt, cond_score):
    z = denoise(zt, t, dt)     # differentiable; denoise zt to time t-dt
    with no_grad():
        target = z + cond_score(z, t-dt)
    loss = (z - target)**2
    return autograd(loss, zt)    # grad wrt zt
```

Backpropagation through diffusion. To compute gradients with respect to $\mathbf{z}_t$, we must backpropagate through unconditional diffusion. This incurs memory cost linear in the number of denoising steps used. In practice, five to ten denoising steps enabled good performance without memory issues.

3 Experiments

We perform experiments on Stable Diffusion (1), an open-source text-to-image latent diffusion model trained on LAION-5B (17) with a pre-trained text conditional and unconditional model. Latent diffusion occurs over 1000 time steps: $\mathbf{z}_0 \rightleftharpoons \mathbf{z}_{1000}$, and an adversarially-trained autoencoder encodes and decodes $\mathbf{x} \rightleftharpoons \mathbf{z}_0$. We treat the conditional diffusion model as the explicit guide. We use the pseudo linear multi-step (PLMS) scheduler (12) which is deterministic.

Approximate guides have high accuracy. In figure 1, we compare our approximated guide $\boldsymbol{\xi}_t$ to Stable Diffusion's conditional guide $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t, \mathbf{c})$ using the cosine similarity between the two gradients (see appendix for full details). Approximate guides obtained by denoising $\mathbf{z}_t$ to $\mathbf{z}_{t-\delta}$ are very similar to Stable Diffusion's guide, with cosine similarity above 0.99 even as δ increases to 500 time steps out of 1000 total diffusion steps. At $\delta = 900$, similarity is maintained above 0.80.

In contrast, approximate guides formed by denoising and decoding $\mathbf{z}_t$ to images $\mathbf{x}$, applying CLIP (18) spherical loss, and backpropagating back to $\mathbf{z}_t$ are essentially orthogonal to $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t, \mathbf{c})$, with mean similarity around 0.01. This is consistent with observations that the manifold of natural images is complex in pixel space, and gradients on images are difficult to use for optimizing latents (19). This highlights the challenge of conditional diffusion sampling using only clean-data classifiers.

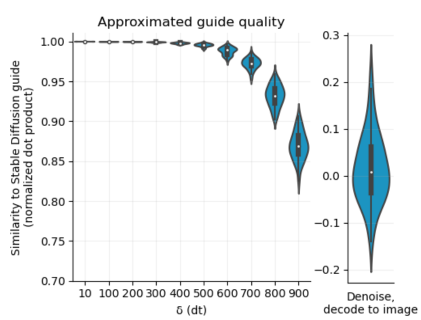


Figure 1: MPC guides have high cosine similarity to real guides

Approximate guides improve robustness to sample quality damage with reduced explicit guidance. We evaluated conditional sampling with explicit guidance restricted to just $n = 5$ time steps, with classifier weight $w = 2$. We compare to using $k = 3$ additional MPC-guided generative steps (with a total of $n + k = 8$ steps), and a *reference* with full explicit guidance on $n + k$ steps - if MPC is accurate, then samples should look similar to the reference. We also generated *gold standard* samples with 50 explicit guidance steps. We evaluated PLMS baselines with both n and $n + k$ generative steps (see appendix). Each approach was initialized with identical z_T; as each approach is deterministic, quality can be judged by similarity to the reference and gold standard.

On random MS-COCO prompts, adding MPC generative steps significantly improved visual sample quality over the baseline (Fig. 2) and improve FID to the reference and gold standard (Table 3). MPC samples are more visually similar to the reference than the baseline, and intriguingly, in some cases seem to outcompete the reference in visual similarity to the gold standard.

FID (↓)	Reference	Gold standard
Baseline ($n = 5$)	400.0	443.28
+ MPC ($k = 3$)	**282.4**	**312.84**

Figure 2: Comparison of samples (Stable Diffusion, pseudo linear multi-step scheduler, guidance weight $w = 2$)

4 Discussion

We described a method for approximating guidance for conditionally sampling from diffusion models with model predictive control, and showed preliminary evidence that approximated guidance improves sample quality when access to a conditional guide is severely restricted to just five time steps.

Looking forward, future work may be interested in addressing instabilities and divergence. In some settings, we found that approximate guides tended to cause divergence to reference latent trajectories over time. We found this issue to be particularly problematic with larger classifier guidance weights w: even if $\boldsymbol{\xi}_t$ is very similar to $\boldsymbol{\epsilon}(\mathbf{z}_t, \mathbf{c})$ (e.g., 0.9999), and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t)$ is identical, the adjusted prediction $\tilde{\boldsymbol{\epsilon}}(\mathbf{z}_t, \mathbf{c}) = (1 + w)\boldsymbol{\xi}_t - w\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t)$ can have significantly lower similarity (e.g., 0.992). We also observed that divergence increased with the number of approximate guidance steps.

Our results suggest the possibility of conditional diffusion using explicit guidance (e.g., a conditional diffusion model) trained on a small number of time steps. Future work can explore this by restricting conditional training; here, we only restricted the time steps at which we queried the ground-truth guide which was trained on all time steps.

References

[1] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[2] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[3] Jonathan Ho and Tim Salimans. Classifier free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[4] Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating bayesian optimization for biological sequence design with denoising autoencoders. In *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022.

[5] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

[6] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *ICCV*, 2021. doi: 10.48550/ARXIV.2108.02938. URL https://arxiv.org/abs/2108.02938.

[7] Vedant Singh, Surgan Jandial, Ayush Chopra, Siddharth Ramesh, Balaji Krishnamurthy, and Vineeth N. Balasubramanian. On conditioning the input noise for controlled image generation with diffusion models, 2022. URL https://arxiv.org/abs/2205.03859.

[8] Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *CVPR*, 2022.

[9] Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. D2c: Diffusion-decoding models for few-shot conditional generation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12533–12548. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/682e0e796084e163c5ca053dd8573b0c-Paper.pdf.

[10] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022. URL https://arxiv.org/abs/2201.09865.

[11] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022. URL https://arxiv.org/abs/2206.00364.

[12] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PlKWVd2yBkY.

[13] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. URL https://arxiv.org/abs/2205.11487.

[14] Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and J. Zico Kolter. Differentiable mpc for end-to-end planning and control. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/ba6d843eb4251a4526ce65d1807a9309-Paper.pdf.

[15] Stephen Piche, James Keeler, Greg Martin, Gene Boe, Doug Johnson, and Mark Gerules. Neural network based model predictive control. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper/1999/file/db957c626a8cd7a27231adfbf51e20eb-Paper.pdf.

[16] Homanga Bharadhwaj, Kevin Xie, and Florian Shkurti. Model-predictive control via cross-entropy and gradient-based optimization. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 277–286. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/bharadhwaj20a.html.

[17] Christoph Schuhmann, Romain Beaumont, Cade W Gordon, Ross Wightman, mehdi cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Mitchell Wortsman, Richard Vencu, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=M3Y74vmsMcY.

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[19] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1laeJrKDB.

[20] Matthias Bühlmann. Stable diffusion based image compression, 2022. URL https://matthias-buehlmann.medium.com/stable-diffusion-based-image-compresssion-6f1f0a399202.

A Appendix

A.1 Experiments

We used an Nvidia A100 with 80 GB memory for our experiments. Backpropagating through diffusion requires backpropagating through Stable Diffusion's U-Net several times. We found that roughly 10 or more denoising steps exceeded the memory of our A100, but that five denoising steps was sufficient for performance.

We used classifier-free guidance weight $w = 2$, following (3). In practice, we scale our approximate guide $\boldsymbol{\xi}_t$ at time t to match the norm of the unconditional score $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t)$.

We will release our code in a future version.

Details on Stable Diffusion. Stable Diffusion was trained with classifier-free guidance, conditioning on CLIP-embedded text prompts (18), with 1000 diffusion time steps. An adversarially-trained autoencoder encodes and decodes images , which is an $8\times$ down-sampled latent space. Latents z_0 were very weakly regularized (10^{-6} weight) towards a unit Gaussian. Despite this, when visualized as images, latents $\mathbf{z}_0$ appear as fuzzy versions of the decoded image $\mathbf{x} = \texttt{decoder}(\mathbf{z}_0)$ (20).

Similarity study. At each starting time t, we initialized z_t by unconditionally denoising from the prior z_T. We obtained an approximate guide for various δ, also called dt. Stable diffusion has 1000 total diffusion timesteps, so we varied t in $[200, 400, 600, 800, 1000]$. We varied δ in increments of 100, and performed 10 replicates for each experimental condition. We used the following text prompts, some of which were from the Stable Diffusion paper (1): 'a photo of a cat', 'a photo of an astronaut riding a horse on mars', 'a street sign that reads latent diffusion', 'a zombie in the style of picasso', 'a watercolor painting of a chair that looks like an octopus', 'an illustration of a slightly conscious neural network'. We observed similar results for all prompts. The plot depicts data for $t = 1000$, for varying δ on the x-axis, across prompts and replicates: there are 60 datapoints for each violin plot, which is smoothed with kernel density estimation using seaborn.

Restricted explicit guidance experiments. Our approach used an eight-step schedule evenly divided from t =1000 to 0: [875, 750, 625, 500, 375, 250, 125, 0], with explicit guidance at times [750, 500, 250, 125, 0] and MPC at [875, 625, 375]. We compare to a *reference* with the same eight-step schedule with full explicit guidance. Our PLMS *baseline* uses the five-step schedule [800, 600, 400, 200, 0] with explicit guidance. We also tried another baseline using the eight-step schedule, explicit guidance at five time steps, and unconditional steps at times [875, 625, 375], but found that this baseline ignored prompts.

Wall-clock time (for one sample). 50 generative steps takes about 12 seconds. 10 generative steps takes about 2.5 seconds. We find that with 5 generative steps, adding 3 MPC steps adds negligible runtime, with all runs finishing in 1-3 seconds. In a separate unreported experimental setting, our method, with 25 total generative denoising steps, guidance at 10 time steps, 10 unconditional denoising steps for approximating the guide, and churning, takes about 34 seconds. The same setting, without churning, takes about 18 seconds.