Accelerating Diffusion Models for Adaptive Motion Planning in Dynamic Environments

Matthias Van Eysendeyk*, Edward Sandra*, Dries Dirckx*, Lander Vanroye* Wilm Decré*
*Department of Mechanical Engineering
KU Leuven and Flanders Make@KU Leuven
3001 Heverlee, Belgium
Email: matthias.vaneysendeyk@kuleuven.be

***Abstract*—Diffusion models have recently emerged as a powerful approach for robot motion planning, capable of generating multimodal, high-quality trajectories. Their main limitation is slow sampling, which hinders real-time replanning in dynamic environments. This work studies several acceleration strategies for diffusion-based motion planning and leverages the resulting speedups to enable replanning at up to 100 Hz in a dynamic benchmark with randomly moving obstacles. A cost-based trajectory selection mechanism is used to exploit multimodality during replanning by balancing motion time, predicted collision risk, and smoothness. Comparisons against warm-start diffusion and MPC show that accelerated diffusion models achieve the best performance in dynamic settings.**

***Index Terms*—Motion Planning, Diffusion Models, Dynamic Environments**

I. Introduction

Robot motion planning is a fundamental problem in robotics, involving the computation of collision-free trajectories from a start state to a goal state within a given environment. Classical approaches such as sampling-based planners and optimization-based methods (e.g., model predictive control (MPC)) have shown strong performance, but often struggle with either multimodality or generalization to complex or dynamic environments.

Diffusion models [1], [2] have recently emerged as a promising alternative. They iteratively denoise Gaussian noise into feasible trajectories, enabling diverse solutions and strong generalization when conditioned on environmental context. Their main limitation is slow sampling, since generating a trajectory requires many sequential network evaluations, making real-time replanning difficult.

This work shows that accelerated diffusion sampling enables high-frequency replanning in dynamic environments, and that explicit selection over sampled trajectories is key to exploiting multimodality during execution.

II. Preliminaries & Related Work

A. Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models (DDPMs) [2], are a class of generative models that learn to reverse a gradual noising process. The idea is to transform data into noise through a forward diffusion process and then learn a reverse process to recover the original data. The forward process is defined as

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\, \beta_t \mathbf{I}\right), \tag{1}$$

where β_t is a variance schedule. After T steps, $\mathbf{x}_T$ approximates an isotropic Gaussian $\mathcal{N}(0, \mathbf{I})$. A closed-form expression allows direct sampling of $\mathbf{x}_t$ from $\mathbf{x}_0 \sim q(\mathbf{x}_0)$:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\, (1-\bar{\alpha}_t)\mathbf{I}\right), \tag{2}$$

with $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$.

The reverse process aims to denoise $\mathbf{x}_t$ back to $\mathbf{x}_0$ and is learned using a neural network with weights θ:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t),\, \mathbf{\Sigma}_\theta(\mathbf{x}_t, t)\right), \tag{3}$$

Typically, only the mean μ is learned and the covariance is set to the lower bound of the reverse process entropy $\mathbf{\Sigma} = \sigma_t^2 \mathbf{I}$, with $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. Additionally, the total added noise ϵ is learned instead of the mean μ by minimizing the following loss:

$$\mathcal{L} = \mathbb{E}_{\mathbf{x}_0, \epsilon, t}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2\right]. \tag{4}$$

Training involves sampling a timestep t, generating $\mathbf{x}_T$ using (2), and updating the network parameters via the loss above. At inference, sampling starts from $\mathbf{x}_t \sim \mathcal{N}(0, I)$ and iteratively applies the learned reverse steps, optionally conditioned on environment information, to produce a valid trajectory.

B. Diffusion Models in Motion Planning

Diffusion models have recently gained attention for robot motion planning due to their ability to generate multimodal, long-horizon trajectories. Existing approaches can be broadly divided based on how task objectives and constraints are incorporated during sampling. Some methods guide the reverse diffusion process using an additional term, such as gradients from a reward or constraint predictor (classifier guidance), to enforce feasibility and goal-directed behavior [3]–[5]. In contrast, other approaches avoid auxiliary guidance at inference time and instead condition the diffusion model on contextual information through architectural mechanisms, such as encoded task descriptions or environment features (classifier-free guidance) [6], [7]. These context-aware models

This work was supported by Flanders Make through SBO project LearnOpTra (Learning meets optimisation for robust and multimodal trajectory planning and control), and by KU Leuven's Special Research Fund (BOF), grant STG/25/004 (Advanced programming and control of smart and high-performance machines, vehicles, and robotic systems).

inject conditioning directly into the U-Net backbone, enabling generalization across tasks and environments while maintaining efficient sampling.

Diffusion models are attractive for motion planning due to their ability to represent complex, multimodal trajectory distributions. However, a major limitation of most diffusion-based planners is their slow inference: generating a plan typically requires T sequential denoising steps (often 20-100), making real-time replanning impractical in dynamic settings. To mitigate this, a *warm-start* strategy can be used [3], [5]. Instead of denoising from pure Gaussian noise, replanning is performed by perturbing a previously generated trajectory with small noise and refining it over $T_r \ll T$ steps. While effective for improving latency, warm-starting alters the sampling behavior: initializing from a near-valid solution significantly reduces exploration and suppresses multimodality. As a result, the planner may converge to a single locally refined solution, undermining one of the central advantages of diffusion models and potentially leading to suboptimal behavior in environments where diverse trajectories are essential.

C. Accelerated Sampling

Reducing diffusion inference latency is a major topic in image generation, but remains relatively underexplored in motion planning. The main computational bottleneck is that sampling requires T sequential neural network evaluations. Existing acceleration methods typically follow two directions: reducing the cost of each diffusion step, e.g., through architectural improvements or latent diffusion [8], and reducing the number of diffusion steps required at inference. This work focuses on the latter. The most relevant approaches are improved sampling schemes, that can simply replace the DDPM sampling scheme without retraining the model, or knowledge distillation using the well-known student-teacher principle.

1) Alternative Sampling Schemes: These methods replace the standard DDPM sampler (Sec. II-A) with more efficient algorithms that require fewer steps while preserving sample quality.

A first important example is Denoising Diffusion Implicit Models (DDIM) [9], which replaces the Markovian reverse process of DDPM with a deterministic, non-Markovian one. Because the training objective depends only on the marginals $q(\mathbf{x}_t \mid \mathbf{x}_0)$, the same model can be sampled without retraining. In practice, DDIM predicts $\mathbf{x}_0$ from the estimated noise,

$$\tilde{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}, \tag{5}$$

and uses this estimate to take larger, deterministic steps during sampling $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \tilde{\mathbf{x}}_0)$.

A complementary view formulates diffusion in continuous time as a stochastic differential equation (SDE) [10], [11]. While the reverse-time SDE remains stochastic and therefore still requires small step sizes, it admits an equivalent deterministic probability flow ODE (PF-ODE) with the same marginals. This ODE perspective enables the use of dedicated numerical solvers with larger steps.

Based on this formulation, DPM-Solver++ [12] was proposed as specialized solvers for the PF-ODE. By exploiting its semi-linear structure, they reduce discretization error more effectively than generic ODE solvers. DDIM can in fact be interpreted as a first-order instance of this family, while DPM-Solver++ further improves stability and efficiency through reparameterization and multistep higher-order updates.

2) Knowledge Distillation: Unlike alternative samplers, which can be used as drop-in replacements, knowledge distillation trains a separate student model to mimic a pretrained diffusion model in fewer steps. The student is typically trained to approximate multiple denoising steps of the teacher with a single update. Progressive Distillation [13] does this iteratively by merging two teacher steps into one student step. Consistency Distillation [14] instead enforces that points along the same ODE trajectory map to the same denoised sample $\mathbf{x}_0$. More recently, Simple and Fast Distillation (SFD) [15] restricts training to the timesteps actually used during inference, resulting in a particularly efficient distillation procedure.

III. Acceleration Strategies for Diffusion-Based Motion Planning

Several strategies for accelerating diffusion-based motion planning are considered, with standard DDPM sampling serving as the baseline. The considered methods are:

- **DDIM** [9]: a first-order deterministic sampler.
- **DPM-Solver++** [12]: a higher-order multistep ODE solver.
- **Distillation**: a lightweight single-stage distillation procedure producing a reduced-step student model.

A. Sampling Schedules and Stability Considerations

Reducing the number of diffusion steps requires an appropriate sampling schedule. For DDIM, a subset of timesteps $S = \{S_0, S_1, \ldots, S_K\}$ is selected from $\{0, 1, \ldots, T\}$, spaced as uniformly as possible:

$$S = \left\{ \text{round}\left(k \cdot \frac{T}{K}\right) \,\middle|\, k = 0, 1, \ldots, K \right\}. \tag{6}$$

Rounding is required because both the diffusion model and the noise schedule are defined on discrete timesteps.

For DPM-Solver++, the schedule is instead chosen to be uniform in half-log signal-to-noise ratio, $\lambda = \log(\sqrt{\bar{\alpha}_t}/\sqrt{1-\bar{\alpha}_t})$:

$$S = \left\{ t_\lambda\left(\lambda(T) + k \cdot \frac{\lambda(0) - \lambda(T)}{K}\right) \,\middle|\, k = 0, 1, \ldots, K \right\}. \tag{7}$$

Unlike DDIM, DPM-Solver++ is not restricted to discrete timesteps. The discrete cosine noise schedule $\bar{\alpha}_t$ is therefore interpolated to obtain a continuous $\bar{\alpha}(t)$, which in practice generalizes well to intermediate times.

To improve numerical stability, static thresholding of $\tilde{\mathbf{x}}_0$ is used for DDIM, while DPM-Solver++ uses dynamic thresholding [16]. Instabilities are most pronounced near $t = T$, where very small $\bar{\alpha}_t$ can amplify noise-prediction errors and produce unrealistic denoised values. Although v-parametrization

has been proposed as a remedy [13], [17], better empirical results were obtained by starting the denoising process from $t = T - m$ rather than $t = T$, with m chosen from a lower bound on $\lambda(t)$.

B. Distillation Training Procedure

The distillation procedure follows the main idea of SFD: the student model is trained on the timesteps used by the teacher during inference. This keeps training short, but ties the student to a specific sampling scheme and reduced step count K.

The student noise predictor ϵ_ψ uses the same architecture as the teacher and is initialized from the teacher weights θ. Training proceeds as follows:

1. Noise is added to clean data using (2), yielding $\mathbf{x}_{S_K}$.
2. The teacher denoises $\mathbf{x}_{S_k} \rightarrow \mathbf{x}_{S_{k-1}}$ using L DPM-Solver++ steps conditioned on encoded context c_θ, while the student predicts the same transition in a single large DDIM step conditioned on c_ψ.
3. The student is optimized to match the teacher output using mean squared error.
4. The teacher output is then used as input for the next step in both models, and the process is repeated until full denoising.

This allows the student to approximate the teacher using only K denoising steps. In the implementation used here, $L = 6$, the sampling schedule is log-SNR uniform as in (7), and static thresholding is applied.

To further reduce function evaluations, the *Analytical First Step* (AFS) trick from SFD is adopted. Since the first denoising input is pure noise, the corresponding noise estimate can be approximated as $\epsilon(\mathbf{x}_{S_K}, S_K) \approx \mathbf{x}_{S_K}$, eliminating one network evaluation. This is particularly beneficial for small K.

C. Distillation of Guided Diffusion

Many diffusion models use classifier-free guidance (CFG) [18], where the noise prediction is computed as

$$\tilde{\epsilon}_\theta(\mathbf{x}_t, t, \mathbf{c}) = \omega\, \epsilon_\theta(\mathbf{x}_t, t, \mathbf{c}) + (1 - \omega)\, \epsilon_\theta(\mathbf{x}_t, t, \emptyset), \quad (8)$$

with guidance scale ω, conditioning context $\mathbf{c}$, and $\emptyset$ denoting the absence of conditioning.

Distilling guided diffusion introduces an additional design choice: whether to preserve flexibility in ω. Prior work addresses this either by conditioning the student on ω [19] or by training over multiple guidance values [20]. Here, a single value of ω is fixed before training and used throughout teacher denoising. This simplifies student training and removes the second model evaluation otherwise required by CFG during inference, as the student implicitly learns the guided model for the chosen ω. Although this sacrifices flexibility at inference time, retraining with a different ω remains practical due to the short training time.

IV. Adaptive Replanning in Dynamic, Uncertain Environments

We consider motion planning in environments with dynamic, uncertain obstacles. At time τ_i, the robot is in configuration q^0 and must reach a goal q^M while avoiding a set of obstacles $\{\mathcal{O}_n\}_{n=1}^N$ whose future motion is stochastic. A trajectory is denoted by $\pi = \{q^0, q^1, \ldots, q^M\}$, where q^h is the configuration at time $\tau_{i+h} = \tau_i + h\Delta\tau$. At each replanning step, we seek a trajectory that balances progress to the goal, collision avoidance under predicted obstacle motion, and consistency with past decisions.

The acceleration strategies described above make diffusion-based online replanning feasible in such settings. Crucially, once sampling is sufficiently fast, a batch of trajectories can be sampled in each replanning step and we can exploit the multimodal trajectory distribution at every decision step rather than relying on a single warm-started refinement [3], [5]. As the diffusion model is trained on static scenes and conditioned on a single obstacle layout at inference time, dynamic environments are handled through repeated replanning with updated static observations.

A. Adaptive Replanning

An online replanning mechanism reduces the dynamic problem to a sequence of locally static planning problems. At fixed intervals $\Delta\tau$, the robot executes the previously selected trajectory while generating a new batch for the next time instant $\tau_{i+1} = \tau_i + \Delta\tau$. The start state for this batch is the predicted robot configuration at τ_{i+1}.

Since trajectory generation for step τ_{i+1} is already initiated at time τ_i, the obstacle observations at τ_{i+1} are not yet available and must be estimated. To this end, we perform a short-horizon prediction by linearly extrapolating obstacle motion from recent observations. Concretely, the pose of obstacle n at step h is denoted by $\hat{\mathcal{O}}_n^h$ and obtained by extrapolating its average velocity over the last four observations. These predicted obstacle poses are used as conditional context for the diffusion model, allowing each replanning step to operate on a static proxy of the otherwise dynamic scene.

B. Optimal Trajectory Selection

At each replanning step, the diffusion model generates a batch $\mathcal{B}$ of candidate trajectories π. The trajectory selected for execution minimizes a composite cost that trades off goal progress, predicted collision risk, and smoothness relative to the previously executed trajectory. This score, referred to as the *ReplanScore*, enables switching between modes in response to moving obstacles.

Let $\phi(q, \mathcal{O}_n)$ denote the signed distance to obstacle n (negative in collision) and define the penetration depth $\delta(q, \mathcal{O}_n) := \max\{0, -\phi(q, \mathcal{O}_n)\}$.

1) Duration cost: Preference is given to trajectories with low remaining motion time from the current state to the goal:

$$\mathcal{C}_{\mathrm{dur}}(\pi) \;=\; \tau_{c \to g}(\pi). \quad (9)$$

2) Predictive collision cost: To account for delayed collisions, a short horizon H is evaluated using predicted obstacle poses. An exponential decay $\gamma \in (0, 1]$ reflects increasing uncertainty:

$$\mathcal{C}_{\mathrm{coll}}(\pi) \;=\; \sum_{h=0}^{H} \gamma^{\,h} \sum_{n=1}^{N} \left[\delta(q^h,\, \hat{\mathcal{O}}_n^h)\right]^2. \quad (10)$$

TABLE I
RESULTS ON THE STATIC FRANKASPHERES DATASET.

Sampler	S(%)↑	FTR(%)↑	Jerk(rad/s^3)↓	Var↑
DDPM	96.6	67.5	9.32	2.88
DDIM	94.1	48.7	9.05	2.56
DPMSolver++	93.7	48.0	9.09	2.44
d-CAMPD	96.4	57.4	9.04	4.08

3) Smoothness cost: To discourage erratic switching, a smoothness penalty is imposed with respect to the previously selected trajectory π_{prev}, shifted by one step:

$$\mathcal{C}_{\text{smooth}}(\pi) \;=\; \sum_{h=0}^{H} \eta^{\,h}\, \max\{0,\, \|q^h - q_{\text{prev}}^{h+1}\|_2^2 - \kappa\}. \quad (11)$$

The selected trajectory minimizes the *ReplanScore*:

$$\mathcal{C}(\pi) = \mathcal{C}_{\text{dur}}(\pi) + w_{\text{c}}\, \mathcal{C}_{\text{coll}}(\pi) + w_{\text{s}}\, \mathcal{C}_{\text{smooth}}(\pi), \quad (12)$$

$$\pi^\star = \arg\min_{\pi \in \mathcal{B}}\; \mathcal{C}(\pi) \quad (13)$$

where $\mathcal{B}$ is the sampled batch of trajectories and $w_{\text{c}}, w_{\text{s}} > 0$ are scalar weights.

C. Comparing the Acceleration Techniques

All acceleration methods are evaluated using CAMPD [7] as backbone model, with standard DDPM sampling ($T = 50$) as baseline. Evaluation is performed on the static FrankaSpheres benchmark for a 7-DoF Franka Emika Panda manipulator planning between random start and goal states in environments with randomly placed spherical obstacles inside the workspace $\mathcal{B}_F = \{[x, y, z] \mid x, y \in [-1, 1],\, z \in [0, 1.3]\}$. The test set contains 3000 unseen environments with 5 to 10 obstacles. Each sampler generates batches of 100 trajectories, and accelerated samplers use $K = 3$ diffusion steps. Table I reports success rate (**S**), feasible trajectory rate (**FTR**), mean RMS jerk, and mean inter-trajectory variance (**Var**) as a proxy for multimodality.

The accelerated samplers retain success rates close to DDPM while reducing inference time from 132 ms to 8 ms. The main degradation appears in feasible trajectory rate, suggesting that context-awareness rather than denoising quality becomes limiting at very low step counts. Among the accelerated methods, the distilled CAMPD model (d-CAMPD) best preserves both success and multimodality, and is therefore used in the dynamic replanning experiments with $K = 3$.

D. Evaluation in Dynamic Environments

Dynamic evaluation is performed on a new test set of 500 unseen FrankaSpheres-like environments. Each environment contains 1 to 5 spherical obstacles whose motion follows an Ornstein–Uhlenbeck process with obstacle-specific long-term mean velocities, introducing stochastic and uncertain dynamics. Obstacle motion is constrained to the workspace $\mathcal{B}_F$ through boundary reflections, and a safety zone is enforced around the robot base.

TABLE II
RESULT ON THE DYNAMIC FRANKASPHERES TEST SET.

Method	S(%)↑	MS	Jerk(rad/s^3)↓	$\tau_{s\to g}(s)$ ↓
warm-start	74.8	0.0	38.3	1.85
MPC	63.4	0.1	15.8	2.37
d-CAMPD	81.2	3.8	56.6	2.14
w/o *ReplanScore*	72.4	0.6	53.0	1.85

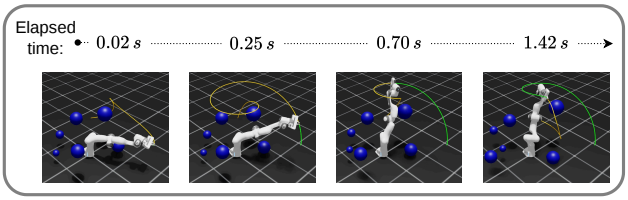

Fig. 1. Example of dynamic replanning at 100 Hz using d-CAMPD with *ReplanScore*. As the dynamic spherical obstacles move through the workspace, the planned trajectory (yellow line) switches between modes to maintain feasibility.

The proposed approach is compared against two baselines: warm-start diffusion and model predictive control (MPC) implemented with CuRobo [21]. The warm-start baseline replans at 100 Hz by denoising a perturbed previous trajectory with DDPM using $T_r = 3$ steps per cycle. The MPC baseline also runs at 100 Hz using predicted obstacle positions at time τ_{i+1}.

To isolate the effect of multimodal trajectory selection, d-CAMPD is evaluated both with and without *ReplanScore*. Without *ReplanScore*, the shortest trajectory in the sampled batch is selected. Table II reports overall success rate (**S**), mean number of mode switches (**MS**), mean RMS jerk, and mean motion time.

The results show that d-CAMPD combined with *ReplanScore* achieves the highest success rate, indicating that fast batched sampling can effectively exploit multimodality during replanning. Selecting only the shortest trajectory greatly reduces mode switching and lowers performance, showing that the gain comes from explicit selection across modes rather than from fast replanning alone. This improved adaptability comes at the cost of increased jerk due to more frequent mode changes.

V. CONCLUSION

This work demonstrates that diffusion models, traditionally applied to static planning tasks, can enable real-time, multimodal motion planning when combined with acceleration techniques and an adaptive replanning scheme, exploiting the multimodality for which diffusion models are known. Experiments show an improvement in success rate compared to MPC and warm-start-based diffusion, underscoring the potential of accelerated diffusion models for dynamic environments. Future research will focus on mitigating jerk caused by frequent mode switches, reducing training and distillation data requirements, deployment on physical systems, and strengthening safety guarantees.

References

[1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 2256–2265.

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, 2020, pp. 6840–6851.

[3] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *Proceedings of the 39th International Conference on Machine Learning*, 2022, pp. 9902–9915.

[4] J. Carvalho, A. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 1916–1923.

[5] S. Zhou, Y. Du, S. Zhang, M. Xu, Y. Shen, W. Xiao, D.-Y. Yeung, and C. Gan, "Adaptive online replanning with diffusion models," in *Advances in Neural Information Processing Systems*, 2023, pp. 44 000–44 016.

[6] F. Ni, J. Hao, Y. Mu, Y. Yuan, Y. Zheng, B. Wang, and Z. Liang, "MetaDiffuser: Diffusion model as conditional planner for offline meta-RL," in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 26 087–26 105.

[7] E. Sandra, L. Vanroye, D. Dirckx, R. Cartuyvels, J. Swevers, and W. Decré, "Accelerated multi-modal motion planning using context-conditioned diffusion models," *arXiv preprint arXiv:2510.14615*, 2025.

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 684–10 695.

[9] J. Song, C. Meng, and S. Ermon, "Denoising Diffusion Implicit Models," *arXiv e-prints*, p. arXiv:2010.02502, 2020.

[10] D. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," in *Advances in Neural Information Processing Systems*, 2021, pp. 21 696–21 707.

[11] Y. Song, J. Sohl-Dickstein, D. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-Based Generative Modeling through Stochastic Differential Equations," *arXiv e-prints*, p. arXiv:2011.13456, 2020.

[12] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "DPM-Solver++: Fast Solver for Guided Sampling of Diffusion Probabilistic Models," *arXiv e-prints*, p. arXiv:2211.01095, 2022.

[13] T. Salimans and J. Ho, "Progressive Distillation for Fast Sampling of Diffusion Models," *arXiv e-prints*, p. arXiv:2202.00512, 2022.

[14] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," *arXiv e-prints*, p. arXiv:2303.01469, 2023.

[15] Z. Zhou, D. Chen, C. Wang, C. Chen, and S. Lyu, "Simple and fast distillation of diffusion models," in *Advances in Neural Information Processing Systems*, 2024, pp. 40 831–40 860.

[16] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, "Photorealistic text-to-image diffusion models with deep language understanding," in *Advances in Neural Information Processing Systems*, 2022, pp. 36 479–36 494.

[17] S. Lin, B. Liu, J. Li, and X. Yang, "Common diffusion noise schedules and sample steps are flawed," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 5404–5411.

[18] J. Ho and T. Salimans, "Classifier-Free Diffusion Guidance," *arXiv e-prints*, p. arXiv:2207.12598, 2022.

[19] C. Meng, R. Rombach, R. Gao, D. Kingma, S. Ermon, J. Ho, and T. Salimans, "On distillation of guided diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 14 297–14 306.

[20] Y. Li, H. Wang, Q. Jin, J. Hu, P. Chemerys, Y. Fu, Y. Wang, S. Tulyakov, and J. Ren, "Snapfusion: Text-to-image diffusion model on mobile devices within two seconds," in *Advances in Neural Information Processing Systems*, 2023, pp. 20 662–20 678.

[21] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, N. Ratliff, and D. Fox, "Curobo: Parallelized collision-free robot motion generation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 8112–8119.