Sum of Costs Diffusion with Dynamic Guidance for Motion Planning

Aysu Aylin Kaplan, Özgür Erkent
Computer Engineering Department, Hacettepe University, Ankara, Turkey
{aysukaplan, ozgurerkent}@hacettepe.edu.tr

***Abstract*—The motion planning problem for robotic manipulation can be addressed through classical or deep learning approaches. Existing methods face significant challenges in generalizing to diverse settings. In this study, we present a method with high generalization capability that generates collision-free trajectories using diffusion models where the denoising process is guided by the gradient of the total collision cost. We are also presenting a dynamic approach for choosing start step of the gradient guidance. Experimental results demonstrate that guiding the diffusion model dynamically with the sum of collision costs offers more robust performance by overcoming the generalization issues faced by competing methods. The proposed model demonstrates its effectiveness by achieving the highest performance on diverse test settings in Mπnets dataset among the compared methods.**

I. Introduction and Related Works

Motion Planning problem can be defined as finding a continuous and collision free trajectory given a start and a goal point. This trajectory for a robotic manipulation task can be found using classical approaches [1–3]. Classical approaches can be divided as optimization or sampling based [3]. Optimization based approaches require extensive finetuning of cost functions and are highly dependent on initialization, whereas although being probabilistically complete, sampling based approaches often require significant computation time to find a solution. The motion planning problem for robotic manipulation task can also be solved using deep learning methods [4, 5]. These methods are highly data dependent and although they achieve high success rates in the settings where they were trained, they fail to generalize to new settings.

In recent years, diffusion models have demonstrated significant capability in image generation [6–8]. Following their success in image generation tasks, recent literature have explored to solve motion planning problem using diffusion models [9–13]. In one of the early studies, Motion Policy Diffusion used gradient of a single cost function to guide the generation process of the diffusion model [12]. EDMP showed that using only one type of a cost function for guidance is not enough for generalization [10]. They performed gradient guidance using different costs for each parallel denoising batch which resulted in improved performance. Depending on the success of EDMP, GPD used diffusion in polynomial parametric space of trajectories [9]. They employed a single cost for guiding the generation and performed stitching on the generated trajectories. GPD claimed that using polynomials as geometric priors improve computation time while maintaining collision avoidance performance. More recently, hierarchical diffusion frameworks have been extended from image synthesis to robotic control and planning [14]. Subsequently, DPPO applied this structural logic to reinforcement learning by formulating a two-layer Diffusion Policy MDP [15]. Expanding this hierarchical approach to the motion planning, Cascaded Diffusion proposed a framework that unifies global and local prediction, feeding higher level model output into lower level diffusion model [16].

In this study, instead of using one cost function to guide the denoising process, we are calculating multiple costs and using the sum of these costs for the gradient guidance. By using the sum of costs, our approach achieves high success rates on both trained and new test settings while demonstrating better generalization than prior methods.

Another important difference in our gradient based guidance method is that, in contrast to prior methods [9, 10, 12], which use intermediate noisy trajectory for the guidance, we compute the gradient using the predicted final trajectory. By utilizing the predicted final trajectory instead of the noisy trajectory for cost calculation, our approach achieves more accurate gradient based guidance performance.

The proposed method utilizes parallel denoising with gradient guidance, calculating the sum of costs via the final trajectory prediction. Furthermore, contrary to prior approaches that initiate gradient guidance at a fixed denoising step, our framework adaptively determines the start step for each problem by evaluating the weight uniformity across parallel processes weighted by their respective sum-of-costs.

Our contributions can be listed as follows:

- We propose a guidance method that uses the gradient of the **sum-of-costs** during the denoising process of the diffusion model.
- Unlike prior approaches that compute gradients on noisy intermediate states, we calculate the gradient on the **predicted final trajectory**.
- We introduce a **dynamic** gradient guidance initiation method based on the weight uniformity of the costs.
- Performance has been improved compared to all other methods tested on the different settings of Mπnets [5] dataset.

II. Approach

Motion Planning Problem: Motion planning for robotic manipulation tasks aim to find a collision-free trajectory

Algorithm 1: Sum-of-Costs Diffusion with Dynamic Guidance

Input: Trained denoising process p_θ, diffusion timestep T, K parallel processes, gradient guidance scale w, weight normalization scale λ.

Output: Trajectory $\boldsymbol{\tau}^{best}$.

Initialization:;
Sample K trajectories: $\boldsymbol{\tau}_T^{(1\ldots K)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
Set guidance active flag $G \leftarrow$ False; $\tilde{U}_{T+1} \leftarrow 0$;
for $t = T, T-1, \ldots, 1$ **do**
  foreach $k \in \{1, \ldots, K\}$ **do**
    Predict final trajectory: $\hat{\boldsymbol{\tau}}_0^{(k)}$ (Eq. 3)
    Compute total cost: $J(\hat{\boldsymbol{\tau}}_0^{(k)})$ (Eq. 4);
  if *G is False* **then**
    Compute the weights w_t (Eq. 5)
    Compute the weight uniformity U_t (Eq. 6)
    Compute $\tilde{U}_t = \gamma \tilde{U}_{t+1} + (1-\gamma) U_t$
    $\nabla_t = |\tilde{U}_t - \tilde{U}_{t+1}|$
    if $\nabla_t < \epsilon$ **then**
      $G \leftarrow$ True;
  foreach $k \in \{1, \ldots, K\}$ **do**
    if *G is True* **then**
      Update via gradient:
      $\tau_t^{(k)} \leftarrow \tau_t^{(k)} - w \nabla J(\hat{\tau}_0^{(k)})$;
    Sample denoised trajectory:
    $\boldsymbol{\tau}_{t-1}^{(k)} \sim p_\theta(\boldsymbol{\tau}_{t-1}^{(k)} \mid \boldsymbol{\tau}_t^{(k)})$;
    Fix boundaries: $\mathbf{q}_0^{(k)} \leftarrow \mathbf{q}_{start}, \mathbf{q}_L^{(k)} \leftarrow \mathbf{q}_{goal}$;
Final Selection:;
Sort $\{\boldsymbol{\tau}_0^{(1\ldots K)}\}$ by $J(\boldsymbol{\tau}_0)$ and select the first collision-free trajectory as $\boldsymbol{\tau}^{best}$;

between a start configuration $\mathbf{q}_s$ and a goal configuration $\mathbf{q}_g$. A trajectory comprising L waypoints is defined as $\tau = \{\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_L\}$. The objective is to obtain a trajectory τ that satisfies the boundary conditions $\mathbf{q}_0 = \mathbf{q}_{start}$ and $\mathbf{q}_L = \mathbf{q}_{goal}$ while avoiding collisions with workspace obstacles and self-collisions.

Trajectory Generation using a Diffusion Model: Forward diffusion process is formally defined as a Markov chain that incrementally adds noise to a given data sample for t timesteps, ultimately transforming the data into a standard normal distribution using β_t, the variance schedule at timestep t and the identity matrix I:

$$q(\tau_t \mid \tau_{t-1}) = \mathcal{N}(\tau_t; \sqrt{1-\beta_t}\tau_{t-1}, \beta_t I) \tag{1}$$

The proposed trajectory generation framework follows an iterative denoising scheme that transforms a Gaussian noise to a kinematically valid and collision-free path in the configuration space, as detailed in Algorithm 1. This process is formulated as a reverse diffusion sequence where, at each timestep t, the model predicts and removes noise to refine the trajectory. This iterative denoising process, parameterized by θ, is defined as:

$$p_\theta(\boldsymbol{\tau}_{t-1} \mid \boldsymbol{\tau}_t) = \mathcal{N}(\boldsymbol{\tau}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{\tau}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{\tau}_t, t)) \tag{2}$$

where $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ are the mean and covariance matrix of the Gaussian distribution computed given $\boldsymbol{\tau}_t$ and t.

The final trajectory estimate $\hat{\boldsymbol{\tau}}_0$ can be computed from the current noisy trajectory $\boldsymbol{\tau}_t$, where $\epsilon_\theta(\boldsymbol{\tau}_t)$ is the noise component predicted by the model [17, 18]:

$$\hat{\boldsymbol{\tau}}_0 = \frac{\boldsymbol{\tau}_t - \sqrt{1-\alpha_t}\,\epsilon_\theta(\boldsymbol{\tau}_t)}{\sqrt{\alpha_t}} \tag{3}$$

Sum-of-Costs: We implement parallel denoising processes $k \in \{1, \ldots, K\}$, and for each process at the timestep t of the denoising, we predict final trajectory $\hat{\boldsymbol{\tau}}_0^{(k)}$. The total cost function of each parallel process $J(\hat{\boldsymbol{\tau}}_0^{(k)})$ is computed using the predicted final trajectory $\hat{\boldsymbol{\tau}}_0^{(k)}$ as the sum of seven swept volume (SV) and five intersection volume (IV) costs [10]:

$$J(\hat{\boldsymbol{\tau}}_0^{(k)}) = \sum_{j=1}^{7} J_{SV,j}(\hat{\boldsymbol{\tau}}_0^{(k)}) + \sum_{l=1}^{5} J_{IV,l}(\hat{\boldsymbol{\tau}}_0^{(k)}) \tag{4}$$

The intersection volume cost detects collisions by calculating penetration between robot and obstacle bounding boxes. The swept volume cost computes the volume swept between consecutive waypoints to prevent collisions between adjacent links. Each individual cost utilizes specific parameters, such as obstacle expansion and clearance distances. Summing these distinct cost functions enhances collision detection robustness by leveraging their different parameters. While a single cost may fail to register a specific contact due to its specific parameterization, the sum-of-costs approach ensures that collisions missed by one cost are captured by the other. By summing these individual volumes into one cost function, the total represents a more comprehensive approximation of the collisions. This sum-of-costs function guides the diffusion model's generation process, enabling the production of trajectories with enhanced generalization capability and safety guarantees.

Dynamic Start Step for Gradient Guidance: Fig. 1 illustrates the empirical distribution of the initiation timestep across evaluated environments. To decide the start step for guidance, we first compute the normalized weights $w_t^{(k)}$ of

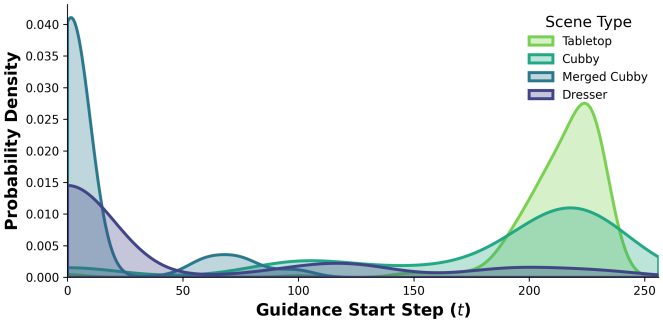

Fig. 1. Probability Density of Gradient Guidance Initiation Timesteps Across Diverse Scenes on Both Test Set.

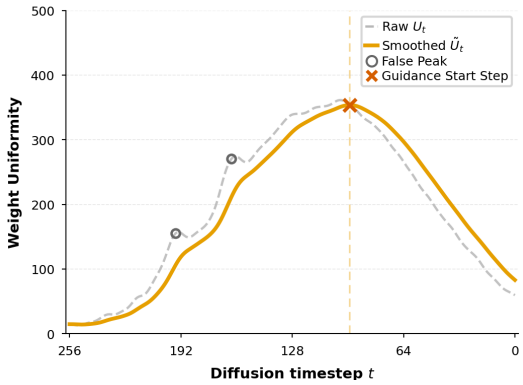

Fig. 2. Effect of EMA Smoothing on Guidance Activation.

each parallel process k, using softmax normalization with scale λ, where the parameter λ acts as a temperature variable that regulates the sharpness of the distribution:

$$w_t^{(k)} = \frac{\exp(-J(\hat{\tau}_0^{(k)})/\lambda)}{\sum_{i=1}^{K} \exp(-J(\hat{\tau}_0^{(i)})/\lambda)} \tag{5}$$

We then compute the weight uniformity (U), which quantifies how evenly the costs are distributed across parallel processes:

$$\mathrm{U}_t = \left[\sum_{i=1}^{K} (\hat{w}_t^i)^2\right]^{-1} \tag{6}$$

Gradient guidance is dynamically initiated at the timestep t where the smoothed weight uniformity $\tilde{U}_t$ reaches its global maximum as illustrated in Fig. 2. This value is computed by applying an exponential moving average to the instantaneous weight uniformity U_t, defined as $\tilde{U}_t = \gamma \tilde{U}_{t+1} + (1-\gamma)U_t$ where $\gamma \in [0, 1]$ is a smoothing coefficient. Mathematically, the initiation point is detected when the gradient $\nabla_t = |\tilde{U}_t - \tilde{U}_{t+1}|$ falls below a threshold ϵ. This peak signifies the point of maximal cost uniformity across parallel processes, identifying the precise transition where stochastic noise is sufficiently reduced for accurate cost evaluation before the model collapses into specific local minima. Initiating guidance at this exact moment ensures that gradients steer the model based on valid structural predictions.

Once initiated, the denoising process is guided by the gradient of the cost function computed on the predicted final trajectory $\hat{\boldsymbol{\tau}}_0$, steering the model toward collision-free regions where the guidance scale w controls the influence of the cost gradient on the denoising process:

$$\boldsymbol{\tau}_t^{(k)} = \boldsymbol{\tau}_t^{(k)} - w \nabla J(\hat{\boldsymbol{\tau}}_0^{(k)}) \tag{7}$$

We propose calculating the cost gradient from the predicted clean trajectory $\hat{\boldsymbol{\tau}}_0$, in contrast to previous studies that compute guidance costs on the noisy trajectories $\boldsymbol{\tau}_t$. This formulation mitigates the effects of stochastic noise present in $\boldsymbol{\tau}_t$, which frequently yields unreliable cost estimations. Calculated on the denoised estimate $\hat{\boldsymbol{\tau}}_0$, the gradient represents the intended output more accurately. This alignment is critical because the primary objective is the generation of a collision free final trajectory; evaluating the cost on the predicted clean state ensures that guidance is directed toward the feasible trajectory space rather than being misdirected by intermediate noise present in $\boldsymbol{\tau}_t$.

Final Trajectory Selection: After denoising, the K generated trajectories are sorted by their sum-of-costs in ascending order. The first collision-free trajectory in this sorted list is selected as the final trajectory $\boldsymbol{\tau}^{best}$.

III. Experiments

We are performing our experiments on 7 DoF Franka Emika Panda robot using PyBullet Physics simulator [19].

A. Experimental Settings

Dataset: Mπnets [5], is a robotic motion planning dataset created to enable robotic manipulators to move through different environments without colliding with obstacles. Mπnets has a hybrid training dataset consisting of 3.27 million collision-free trajectories on four type of scenes: tabletop, cubby, merged cubby and dresser. Hybrid training dataset is generated using a hybrid planner that combined AIT* [20] and Geometric Fabrics [21]. We evaluate our approach on the Hybrid, Both, and Global test sets of Mπnets [5]. Each test set includes 1800 problems across the same four scene types as the training dataset. The Hybrid test set contains problems solvable only by the hybrid planner, while the Global test set consists of problems solvable exclusively by the global planner, AIT* [20]. The Both test set comprises problems solvable by both planners. Each problem consists of a start configuration, a goal position and obstacle configurations.

Model and Training Details: We train a temporal U-NET model similar to [10] [9] on the Mπnets [5] Hybrid training dataset. We train the model for 7 hours on an RTX A6000 GPU for 2500 steps using a batch size of 16384 and a learning rate of 0.008 with SGD optimizer. We set the diffusion timestep to T=256 and the model outputs joint configurations for 50 waypoints along a trajectory.

Baselines: In the experiments, we compare our approach against optimization based planners G.Fabrics [21], CHOMP [1] and STORM [2], deep learning methods MPNets [4] and Mπnets [5], diffusion based methods EDMP [10], GPD [9] and Cascaded Diffusion [16]. For all deep learning and diffusion-based baselines, we report the performance of models trained on the Hybrid training set of Mπnets [5].

Implementation Details: We perform K parallel denoising process where we set K as 500. The selection of an ensemble size $K = 500$ ensures that a large number of parallel processes explore the trajectory space, which facilitates the generation of more diverse and generalizable solutions. To implement gradient cost guidance, an empirically determined guidance scale $\omega = 0.1$ is utilized. Additionally, softmax cost normalization is performed using a scale factor $\lambda = 0.1$ which is selected to ensure that the weight distribution across parallel denoising processes is sufficiently sensitive to produce a distinct and identifiable peak using the uniformity metric U_t.

TABLE I
COMPARISON OF SUCCESS RATES(%). THE BEST RESULTS ARE INDICATED IN **BOLD**, AND THE SECOND-BEST RESULTS ARE UNDERLINED.

Method	Global	Both	Hybrid
G.Fabrics	38.44	60.06	59.33
Storm	50.22	76.00	74.50
CHOMP	26.67	32.20	31.61
MPNets	65.28	67.67	41.33
MπNets	75.78	95.06	95.33
EDMP	75.93	85.06	86.13
GDP	87.00	92.61	92.83
Cascaded Diffusion	85.13	-	**98.00**
Ours	**94.22**	**96.88**	96.66

TABLE II
ABLATION OF GRADIENT GUIDANCE START STEP ON SUCCESS RATE (%).

Start Step	32	64	128	256	Dynamic(Ours)
Success Rate	90.88	94.33	95.44	96.22	**96.88**

TABLE III
ABLATION STUDY ON GRADIENT GUIDANCE USING PREDICTED TRAJECTORIES, SHOWING SUCCESS RATES (%).

Guidance Trajectory	Global	Both	Hybrid
Current Trajectory(τ_t)	89.44	91.44	91.94
Predicted Trajectory($\hat{\tau}_0$)	**94.22**	**96.88**	**96.66**

At each timestep, the denoising process may generate start and goal configurations that deviate from the problem specifications. Therefore, we enforce the boundary conditions by resetting the first and last waypoints of the trajectory to the target problem configurations after each denoising step, following [9, 10, 12].

Robot and obstacles are modeled as AABBs (Axis-Aligned Bounding Boxes) in the Pybullet simulation for cost calculation. After the diffusion model generation, a final collision check is performed using robot and obstacle meshes. A problem is considered successful if the generated trajectory is collision-free, including self-collisions. Success rate is the percentage of successful problems relative to the total number of problems as defined in Mπnets [5].

B. Experimental Results

The proposed approach demonstrates competitive performance across the evaluated benchmarks as presented in Table I. Specifically, the method achieves the highest results on the Global and Both datasets among the tested baselines. In the Hybrid dataset, our performance is comparable to Cascaded Diffusion [16], by a 1.34% margin. This slight performance gap can be attributed to the hierarchical architecture of the cascaded approach, which sequentially feeds the output of a global model as a conditioned input into a local diffusion model. This multi-stage approach provides a more informed starting point compared to our method, which generates trajectories starting from Gaussian noise. Furthermore, while Cascaded Diffusion incorporates an explicit plan refinement stage on the final output, which likely contributes to its slightly higher success rate in the hybrid test set, our approach eliminates the need for such post-processing. Notably, unlike methods that rely on trajectory stitching [9, 16] or trajectory optimization [13], our framework generates valid paths directly through the guided denoising process.

The proposed method demonstrates strong generalization capabilities; while trained on the Hybrid dataset, it achieves its highest performance on the Both dataset. These results suggest that integrating sum-of-costs guidance with a dynamic start step enhances performance by adaptively initiating cost steering.

C. Ablations

Ablation on Dynamic Start Step: In Table II, we explored the effect of the start step to start gradient guidance on the success rate on the Both test set. We compared the fixed timestep guidance baseline and our proposed dynamic gradient guidance method. The results in Table II indicate that initiating guidance at earlier stages of the denoising process (e.g., T=256) leads to higher success rates compared to late stage initiation (T=32), with performance increasing from 90.88% to 96.22%. This trend suggests that earlier intervention provides more iterations for cost guidance to effectively influence the trajectory before it reaches final convergence. While this trend suggests that early intervention provides more iterations for cost guidance to effectively influence the trajectory before convergence, our dynamic initiation method achieves the highest overall performance. As illustrated in Fig. 1, the dynamically selected start step varies significantly across scene types, demonstrating that adaptively determining the guidance start step for each problem optimizes the transition from generative sampling to cost guidance better than using a fixed start step.

Ablation on Gradient Guidance on the Predicted Trajectory: We compared the effect of the chosen trajectory for the cost calculation on the gradient guidance in Table III. Calculating gradient guidance using the predicted final trajectory rather than the noisy intermediate trajectory yielded an approximately 5% increase in success rate across all test sets. This improvement suggests that the predicted trajectory provides a more accurate representation for cost evaluation.

IV. CONCLUSION

In this study, a trajectory generation approach is presented that steers the denoising process of a diffusion model by utilizing sum-of-costs guidance. By integrating a diffusion prior with an adaptive guidance start step, the method seeks to generate valid paths directly within the denoising phase. Experimental results demonstrate strong generalization; although trained on a Hybrid dataset, our model outperforms all baselines on the Global and Both sets with the dynamic start mechanism. Ablation studies confirm that our dynamic guidance approach is more effective than starting guidance at a fixed start step using the predicted final trajectory for the gradient guidance.

References

[1] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 489–494.

[2] M. Bhardwaj, B. Sundaralingam, A. Mousavian, N. D. Ratliff, D. Fox, F. Ramos, and B. Boots, "Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 750–759.

[3] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[4] A. H. Qureshi, A. Simeonov, M. J. Bency, and M. C. Yip, "Motion planning networks," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 2118–2124. [Online]. Available: http://arxiv.org/abs/1806.05767

[5] A. Fishman, A. Murali, C. Eppner, B. Peele, B. Boots, and D. Fox, "Motion policy networks," in *Conference on Robot Learning*. PMLR, 2023, pp. 967–977.

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

[7] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[8] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," in *International Conference on Learning Representations*, 2024, pp. 1862–1874.

[9] A. Srikanth, P. Mahajan, K. Saha, V. Mandadi, P. Paul, P. Wadhwani, B. Bhowmick, A. Singh, and M. Krishna, "GPD: Guided polynomial diffusion for motion planning," in *IEEE International Conference on Automation Science and Engineering*, 2025, pp. 2758–2765.

[10] K. Saha, V. Mandadi, J. Reddy, A. Srikanth, A. Agarwal, B. Sen, A. Singh, and M. Krishna, "Edmp: Ensemble-of-costs-guided diffusion for motion planning," in *IEEE International Conference on Robotics and Automation*, 2024, p. 10351–10358. [Online]. Available: http://dx.doi.org/10.1109/ICRA57147.2024.10610519

[11] Y. Luo, C. Sun, J. B. Tenenbaum, and Y. Du, "Potential based diffusion motion planning," in *International Conference on Machine Learning*, 2024. [Online]. Available: https://energy-based-model.github.io/potential-

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

[13] H. Huang, B. Sundaralingam, A. Mousavian, A. Murali, K. Goldberg, and D. Fox, "Diffusionseeder: Seeding motion optimization with diffusion for rapid motion planning," 2024. [Online]. Available: https://arxiv.org/abs/2410.16727

[14] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, "Cascaded diffusion models for high fidelity image generation," 2021. [Online]. Available: https://arxiv.org/abs/2106.15282

[15] A. Ren, J. Lidard, L. Ankile, A. Simeonov, P. Agrawal, A. Majumdar, B. Burchfiel, H. Dai, and M. Simchowitz, "Diffusion policy policy optimization," in *International Conference on Learning Representations*, 2025, pp. 77 288–77 329.

[16] M. Sharma, A. Fishman, V. Kumar, C. Paxton, and O. Kroemer, "Cascaded diffusion models for neural motion planning," in *International Conference on Robotics and Automation*, 2025. [Online]. Available: https://arxiv.org/pdf/2505.15157

[17] H. Chung, J. Kim, M. T. Mccann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems," *arXiv preprint arXiv:2209.14687*, 2022.

[18] B. Efron, "Tweedie's formula and selection bias," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1602–1614, 2011.

[19] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[20] M. P. Strub and J. D. Gammell, "Adaptively informed trees (ait): Fast asymptotically optimal path planning through adaptive heuristics," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 3191–3198.

[21] M. Xie, K. Van Wyk, A. Li, M. A. Rana, Q. Wan, D. Fox, B. Boots, and N. Ratliff, "Geometric fabrics for the acceleration-based design of robotic motion," *arXiv preprint arXiv:2010.14750*, 2020.