Accelerated Multi-Modal Motion Planning Using Context-Conditioned Diffusion Models

Edward Sandra[1,3] Lander Vanroye[1,3] Dries Dirckx[1,3] Ruben Cartuyvels[2] Jan Swevers[1,3] Wilm Decré[1,3]

***Abstract*— Classical methods in robot motion planning, such as sampling-based and optimization-based methods, often struggle with scalability towards higher-dimensional state spaces and complex environments. Diffusion models, known for their capability to learn complex, high-dimensional and multi-modal data distributions, provide a promising alternative when applied to motion planning problems and have already shown interesting results. However, most of the current approaches train their model for a single environment, limiting their generalization to unseen environments. The techniques that do train a model for multiple environments rely on a specific camera to provide the model with the necessary environmental information. To effectively adapt to diverse scenarios without the need for retraining, this research proposes Context-Aware Motion Planning Diffusion (CAMPD). CAMPD leverages a classifier-free denoising probabilistic diffusion model, conditioned on sensor-agnostic contextual information. An attention mechanism, integrated in the well-known U-Net architecture, conditions the model on an arbitrary number of contextual parameters. CAMPD is evaluated on a 7-DoF robot manipulator and benchmarked against state-of-the-art approaches on real-world tasks, demonstrating its adaptability to changing and unseen environments while generating high-quality, multi-modal trajectories at a fraction of the time required by existing methods.**

I. INTRODUCTION

Robot motion planning is a fundamental problem in robotics, involving the task of finding a collision-free trajectory for a robot between a start and goal configuration. Solving this problem can be challenging, particularly in high-dimensional and complex environments. Moreover, motion planners are frequently required to consider constraints beyond avoiding collisions, such as dynamic constraints to ensure the feasibility of the motion on the robot. Additionally, the motion planner should be highly efficient to respond effectively to disturbances in the environment within a limited time frame during online deployment. Classical approaches fall into sampling-based and optimization-based methods. Sampling-based methods [1], [2] are asymptotically complete but often yield non-smooth trajectories and scale poorly with problem size. Optimization-based methods [3], [4] can handle constraints and objectives but are sensitive to initialization and prone to local minima. Recent learning-based techniques offer an alternative to classical motion planning, broadly divided into learning from demonstration and reinforcement learning (RL) [5]. Learning from demonstration trains networks on expert trajectories, either generating collision-free paths directly [6] or guiding classical planners [7]. RL, by contrast, optimizes a reward through trial-and-error, requiring no expert data. These methods show promise for high-dimensional problems but face challenges in generalization, data efficiency, and reward design. Diffusion models [8] have achieved success in image, video, and speech generation [9] and are increasingly applied in robotics for both imitation [7] and policy learning [10]. However, important research challenges remain [11], particularly in enhancing the generalization and robustness of these deep learning-based methods. Existing research typically proposes methods to train a diffusion model for a single environment, limiting the model's adaptability to different environments [6], [10], [12]. The techniques that do train a model for multiple environments [7] rely on a camera to provide the necessary environmental information, requiring an additional processing step and increasing the algorithm's complexity. This work addresses the generalization challenge by introducing Context-Aware Motion Planning Diffusion (CAMPD), a classifier-free diffusion probabilistic model (DPM) for robot motion planning, capable of rapidly generating multi-modal solutions and adapting to unseen environments and tasks. CAMPD is a planning-as-inference approach [13] that generates collision-free, near-optimal trajectories by sampling from a learned conditional distribution. A U-Net [14] incorporating attention mechanisms [15] conditions the model on environmental context, and classifier-free guidance steers trajectories toward safe regions. The key contributions of this work are: (1) a context-aware diffusion-based motion planner that integrates sensor-agnostic contextual information directly into the model; (2) significantly improved generalization to novel environments compared to state-of-the-art planners; and (3) real-time generation of high-quality, dynamically feasible trajectories, enabling rapid planning and replanning in dynamic settings. Evaluated on diverse problems, the approach demonstrates strong generalization, captures multi-modal solutions, and delivers efficient performance at a fraction of existing methods.

II. RELATED WORK

Diffusion probabilistic models [8] have shown promise in generating multi-modal plans for high-dimensional problems. Janner et al. [6] proposed a method where sampling and planning coincide, but it requires retraining for each environment, limiting adaptability. Cost-guided approaches such as Motion Planning Diffusion (MPD) [12] generalize to

[1]Department of Mechanical Engineering, KU Leuven, 3001 Heverlee, Belgium
[2]Department of Computer Science, KU Leuven, 3001 Heverlee, Belgium
[3]Flanders Make@KU Leuven, Belgium
Correspondence to: Edward Sandra edward.sandra@kuleuven.be

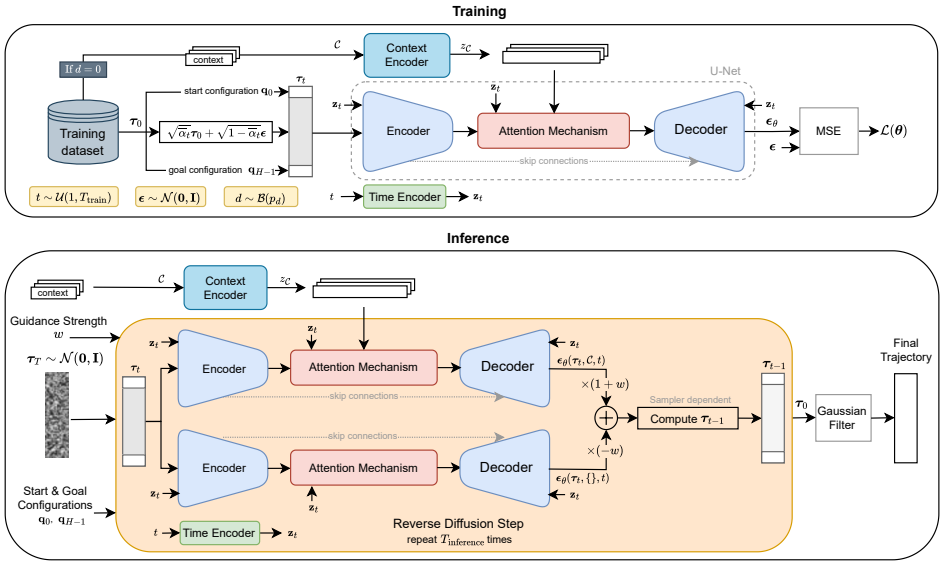


Fig. 1: Overview of CAMPD: During training, trajectories are perturbed with Gaussian noise at a random diffusion step, while start and goal states remain fixed for conditioning. Context is encoded and randomly dropped (Bernoulli p_d) for conditional/unconditional training. A U-Net with attention predicts the noise using MSE loss. During inference, noise is iteratively denoised via reverse diffusion with classifier-free guidance (w), keeping start and goal fixed. Finally, a Gaussian filter is applied to the resulting trajectory to reduce jerk.

unseen environments but rely on hand-crafted cost functions, which demand tuning and increase computation time. Other methods condition directly on environmental input: SceneDiffuser [16] uses point clouds, while DiffusionSeeder [7] employs depth images encoded by a Vision Transformer [17] and integrates cuRobo [3] to blend sampling with optimization. These methods achieve strong results but depend on complex sensory representations, even when simpler structured descriptions (e.g., object dimensions and locations) are available. CAMPD instead leverages a classifier-free DPM conditioned on sensor-agnostic contextual information, enabling flexible environment descriptions. In contrast to Decision Diffuser [18], it supports an arbitrary number of contextual elements such as variable obstacles.

III. METHODOLOGY

This work studies the robot motion planning problem, where the goal is to rapidly generate a fast, collision-free trajectory to bring a robot manipulator from an initial configuration to a goal configuration.

A. Terminology

A trajectory $\boldsymbol{\tau} \triangleq (\mathbf{q}_0, \ldots, \mathbf{q}_{H-1}) \in \mathbb{R}^{d_q \times H}$ is a sequence of robot configurations over horizon H in a d_q-dimensional space. The context $\mathcal{C} = \mathbf{c}_{k,l}$ denotes structured environmental and task parameters, organized into K types with L_k instances per type k (e.g., spheres $\mathbf{c}_{1,l} = [x_l, y_l, z_l, r_l]$). The start and goal configurations $\mathbf{q}_0$ and $\mathbf{q}_{H-1}$ are excluded from the context.

B. Neural Network Architecture

As in MPD [12], CAMPD employs planning-as-inference via a diffusion probabilistic model (DPM) [8], [19]. A DPM consists of two processes: a forward and reverse diffusion process. The noise $\boldsymbol{\epsilon}$ added in the forward process is learned using the mean squared error (MSE) loss:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{t,\boldsymbol{\epsilon},\boldsymbol{\tau}_0,\mathcal{C}} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{\tau}_t, \mathcal{C}, t)\|^2 \right], \tag{1}$$

with $\boldsymbol{\theta}$ the network parameters and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ the learned estimate of $\boldsymbol{\epsilon}$. The estimated mean $\boldsymbol{\mu}_\theta$ of the reverse process is then given by [8]

$$\boldsymbol{\mu}_\theta(\boldsymbol{\tau}_t, \mathcal{C}, t) = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{\tau}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{\tau}_t, \mathcal{C}, t) \right). \tag{2}$$

In CAMPD, the diffusion model estimates the added noise $\boldsymbol{\epsilon}$ in both conditional and unconditional modes, such that classifier-free guidance [20] can be employed:

$$\boldsymbol{\epsilon}_{\boldsymbol{\theta}}{}'(\boldsymbol{\tau}_t, \mathcal{C}, t) = (1+w)\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{\tau}_t, \mathcal{C}, t) - w\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{\tau}_t, \{\}, t) \tag{3}$$

with w the guidance strength. The network architecture consists of a time encoder, context encoder and a U-Net, as illustrated in Figure 1.

a) Time Encoder: The time encoder applies sinusoidal positional encoding [15] followed by a single-hidden-layer multi-layer perceptron (MLP), yielding a latent representation $\mathbf{z}_t = \text{MLP}_t(\text{PE}(t)) \in \mathbb{R}^{d_z}$.

b) Context Encoder: The context encoder maps $\mathcal{C}$ to latent representations $z_\mathcal{C} = \mathbf{z}_{\mathbf{c}_{k,l}} \in \mathbb{R}^{d_z}$, where each instance $\mathbf{c}_{k,l}$ of type k is processed by a single-hidden-layer MLP_k, i.e., $\mathbf{z}_{\mathbf{c}_{k,l}} = \text{MLP}_k(\mathbf{c}_{k,l})$. The resulting representations are used in the U-Net attention mechanism.

c) U-Net: The U-Net $\boldsymbol{\epsilon}_{\boldsymbol{\theta},\text{U}}(\boldsymbol{\tau}_t, z_\mathcal{C}, \mathbf{z}_t)$ predicts the added noise $\boldsymbol{\epsilon}$ from the noisy trajectory $\boldsymbol{\tau}_t$, conditioned on latent time $\mathbf{z}_t$ and context $\mathbf{z}_{\mathbf{c}_{k,l}}$. The encoder–decoder structure follows prior diffusion-based planning work [6], with an

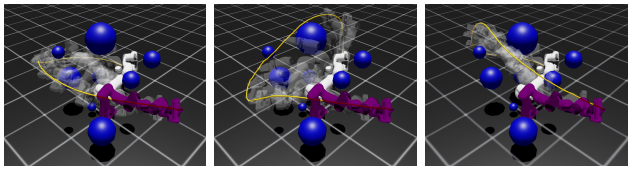
attention module between them. This module applies self-attention and cross-attention [15] over $\mathbf{z}_t$ and $\mathbf{z}_{\mathbf{c}_{k,l}}$, allowing conditioning on arbitrary numbers of context instances (e.g., varying obstacle configurations).

C. Training

Given a noisy trajectory $\boldsymbol{\tau}_t$, time step t, and context $\mathcal{C}$, the model is optimized to predict the added noise $\boldsymbol{\epsilon}$ using the MSE loss (Equation (1)). Classifier-free guidance is enabled by randomly omitting context with probability p_d, while start and goal configurations remain fixed. The training procedure is illustrated in Figure 1.

D. Inference

Trajectories are obtained by iteratively denoising Gaussian noise via a reverse diffusion process, implemented using samplers such as DDPM [8], DDIM [21], or DPM-Solver++ [22]. Classifier-free guidance [20] is applied to enhance contextual conditioning (Equation (3)). The first and last states of the noisy sequence $\boldsymbol{\tau}_t$ are fixed to the start and goal configurations. Finally, a Gaussian filter is applied to the resulting trajectory to reduce jerk. The full inference procedure is illustrated in Figure 1.



(a) (b) (c)

Fig. 2: Example of multi-modal trajectories generated by CAMPD in an unseen environment with spherical obstacles (blue). The white and purple arms indicate start and goal configurations, respectively. The yellow line shows the end-effector trajectory, with (c) highlighting the best trajectory of a batch of 100 samples.

TABLE I: Evaluation on sphere environments. For each unseen configuration (environment, start, goal), 100 trajectories are generated. Metrics: Time (T) = batch generation time; Success (S) = at least one feasible trajectory; Feasible Trajectory Rate (FTR) = % collision-free; Best Smoothness Difference (BSD) = smoothness gap to baseline; Variance (Var) = summed joint distance variances.

Experiment	Method		T (s) ↓	S (%) ↑	FTR (%) ↑	BSD (%) ↓	Var ↑
Random Spheres	RRTC + Fatrop		16.49 ± 14.70*	86.6 ± 34.1	–	–	0.5 ± 1.4
	CAMPD	$w = 1$	0.066 ± 0.0	97.5 ± 15.6	79.4 ± 31.0	2.2 ± 7.0	1.7 ± 3.3
		$w = 1.5$	0.066 ± 0.0	97.4 ± 15.9	82.3 ± 30.4	3.9 ± 10.7	1.9 ± 3.7
		$w = 2$	0.066 ± 0.0	97.0 ± 17.1	82.8 ± 30.7	5.0 ± 12.4	2.2 ± 4.5
		$w = 5$	0.066 ± 0.0	97.0 ± 17.1	67.9 ± 31.5	9.7 ± 45.2	3.7 ± 8.3
Partially Random Spheres	MPD		3.165 ± 0.8	80.7 ± 39.5	59.6 ± 43.4	–	4.8 ± 6.8
	CAMPD	$w = 1.5$	0.066 ± 0.0	90.7 ± 29.1	73.3 ± 37.5	-36.0 ± 14.5	1.7 ± 3.1

*Under ideal parallel CPU execution.

IV. EXPERIMENTAL RESULTS

Experiments were conducted on a 7-DoF Franka Emika Panda robot, comparing CAMPD to a classical motion planner (RRT-Connect [1] + Fatrop [4]), MPD [12], and, on the MπNet [23] dataset of real-world tasks, to cuRobo (v0.6.2) [3] and DiffusionSeeder [7]. All methods use the same architecture and hyperparameters; details are provided in Appendix V-A.

TABLE II: Evaluation on the MπNet test set.

	$N_{\text{batch}} = 12$							$N_{\text{batch}} = 64$				
	cuRobo	DiffusionSeeder		IK cuRobo + **CAMPD**				cuRobo	IK cuRobo + **CAMPD**			
				DDPM	DPMSolver++				DDPM	DPMSolver++		
		$N_{\text{iters}} = 25$	50	$w = 1$	1	2	5		$w = 1$	1	2	5
Plan Time (ms)	26	15	17	7+20	8+5	8+5	7+5	111	8+52	8+7	8+7	8+7
Success Rate	92.0%	85.1%	85.8%	90.7%	86.9%	87.6%	78.9%	93.8%	98.3%	96.2%	97.2%	93.3%
Jerk (rad/s^3)	36.5	108.8	103.6	26.1	25.9	29.1	41.6	62.2	37.1	37.2	42.0	75.6
Motion Time (s)	1.12	1.26	1.26	1.38	1.36	1.37	1.47	1.01	1.13	1.14	1.14	1.22
Translation Err (mm)	0.60	0.98	0.95	0.004	0.003	0.003	0.003	0.87	0.003	0.003	0.003	0.003
Quaternion Err (°)	0.13	0.93	1.44	0.005	0.004	0.004	0.004	0.16	0.004	0.004	0.004	0.004

A. Evaluation in Sphere-Based Environments

The task requires navigating from a start to a goal configuration through environments containing one to ten randomly placed spheres, spaced to ensure feasible paths. Each sphere is represented by $\mathbf{c}_{\text{sphere},l} = [x_l, y_l, z_l, r_l]$, encoding position and radius. The training set, generated with a hybrid planner combining RRT-Connect [1] and Fatrop [4], comprises 113 469 smooth trajectories across 2400 environments. The reverse sampling process was performed using DDPM [8].

Figure 2 shows an example of multi-modal solutions generated by CAMPD in an unseen environment. Table I summarizes the results. On 1000 unseen environments, CAMPD outperforms the hybrid planner in success rate and variance. Although CAMPD's best trajectories are slightly less smooth on average, it can find more locally optimal solutions by identifying modes missed by the hybrid planner, thereby occasionally surpassing the method used to generate its dataset. On 600 environments with partially fixed obstacles, CAMPD also outperforms MPD in success rate, feasible trajectory rate, and smoothness, despite not having seen parts of these environments during training. MPD explores more local minima due to it's balanced optimization costs, producing more diverse but less smooth solutions. CAMPD's performance is sensitive to guidance strength w, which increases success and feasible trajectory rates but may reduce smoothness. CAMPD is computationally efficient and deterministic, taking only $0.066\,\text{s}$ per batch of 100 trajectories, compared to $(16.49 \pm 14.73)\,\text{s}$ for the hybrid planner and $(3.165 \pm 0.008)\,\text{s}$ for MPD, whose slowness stems from cost-gradient computations or low RRT-Connect success rates.

B. Real-World Task Evaluation

To evaluate CAMPD in real-world scenarios, a model was trained on the MπNet simulation test set [23], where each task specifies an initial configuration and target end-effector pose. During inference, the cuRobo [3] IK solver generates feasible joint-space goals. The task context includes cuboids and spheres, represented by $\mathbf{c}_{\text{cuboid},l} = [x_l, y_l, z_l, w_l, h_l, d_l, q_{w,l}, q_{x,l}, q_{y,l}, q_{z,l}]$ and $\mathbf{c}_{\text{sphere},l} = [x_l, y_l, z_l, r_l, h_l, q_{w,l}, q_{x,l}, q_{y,l}, q_{z,l}]$, where positions, dimensions/radii, and orientations are encoded. The training set, generated with cuRobo [3], contains 1 million trajectories across 200 000 environments. CAMPD is

evaluated against two baselines: cuRobo [3] and DiffusionSeeder [7]. Table II reports results on the MπNet test set, with baseline metrics of DiffusionSeeder taken from [7]. Note that DiffusionSeeder uses a depth camera for obstacle detection, not exact object poses; ESDF computation time is excluded for fair comparison. CAMPD achieves faster planning times than both cuRobo and DiffusionSeeder. Moreover, at higher batch sizes, CAMPD surpasses cuRobo in terms of success rate. However, cuRobo consistently produces trajectories with the shortest motion time.

V. CONCLUSIONS

This work proposes Context-Aware Motion Planning Diffusion (CAMPD), a diffusion-based motion planning method capable of incorporating contextual information. CAMPD utilizes a diffusion model to estimate the added noise in trajectories, conditioned on contextual factors. Experiments conducted in simulation demonstrate that CAMPD excels in generalizing to unseen environments, outperforming the existing method MPD in terms of success rate and solution optimality. Additionally, CAMPD showcases exceptional computational efficiency, making it suitable for online applications. Its ability to generate high-quality, executable trajectories directly on the robot without the need for post-processing further highlights its potential for practical deployment in complex motion planning tasks. Future work will focus on enhancing the data generation process, improving the model's generalization capabilities, reducing its computational complexity even more, and expanding its applicability to sensor-based environments.

APPENDIX

A. Model Parameters

Diffusion steps were $T_{\text{train}} = 25$ with latent dimension $d_z = 64$. The U-Net had depth 4, 4 attention heads, and input dimension 64 per head. Training used batch size 128, learning rate 1×10^{-4}, and unconditional probability $p_d = 0.33$. Models ran on an NVIDIA© GeForce RTX™ 4090 GPU. For DDPM, $T_{\text{inf}} = T_{\text{train}}$; for DPMSolver++, $T_{\text{inf}} = 3$.

ACKNOWLEDGMENT

This work was supported 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) and by Flanders Make through SBO project LearnOpTra (Learning meets optimization for robust and multimodal trajectory planning and control). Flanders Make is the Flemish strategic research center for the manufacturing industry.

REFERENCES

[1] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000, pp. 995–1001 vol.2.

[2] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[3] 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.

[4] L. Vanroye, A. Sathya, J. De Schutter, and W. Decré, "Fatrop : A fast constrained optimal control problem solver for robot trajectory optimization and control," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 10 036–10 043.

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[6] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," 2022. [Online]. Available: https://arxiv.org/abs/2205.09991

[7] 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

[8] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[9] H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P.-A. Heng, and S. Z. Li, "A survey on generative diffusion models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 2814–2830, 2024.

[10] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, 2024.

[11] T. Ubukata, J. Li, and K. Tei, "Diffusion model for planning: A systematic literature review," 2024. [Online]. Available: https://arxiv.org/abs/2408.10266

[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," *arXiv.org*, 2023.

[13] B. Matthew and T. Marc, "Planning as inference," *Trends in Cognitive Sciences*, vol. 16, no. 10, pp. 485–488, 2012.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[16] S. Huang, Z. Wang, P. Li, B. Jia, T. Liu, Y. Zhu, W. Liang, and S.-C. Zhu, "Diffusion-based generation, optimization, and planning in 3d scenes," 2023. [Online]. Available: https://arxiv.org/abs/2301.06015

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021. [Online]. Available: https://arxiv.org/abs/2010.11929

[18] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision-making?" 2023. [Online]. Available: https://arxiv.org/abs/2211.15657

[19] 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*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 2256–2265.

[20] J. Ho and T. Salimans, "Classifier-free diffusion guidance," 2022. [Online]. Available: https://arxiv.org/abs/2207.12598

[21] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2022. [Online]. Available: https://arxiv.org/abs/2010.02502

[22] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models," 2023. [Online]. Available: https://arxiv.org/abs/2211.01095

[23] A. Fishman, A. Murali, C. Eppner, B. Peele, B. Boots, and D. Fox, "Motion policy networks," 2022. [Online]. Available: https://arxiv.org/abs/2210.12209