Steerable Scene Generation with Post Training and Inference-Time Search

Nicholas Pfaff[1], **Hongkai Dai**[2], **Sergey Zakharov**[2], **Shun Iwase**[2,3], **Russ Tedrake**[1,2]
[1]Massachusetts Institute of Technology, [2]Toyota Research Institute, [3]Carnegie Mellon University

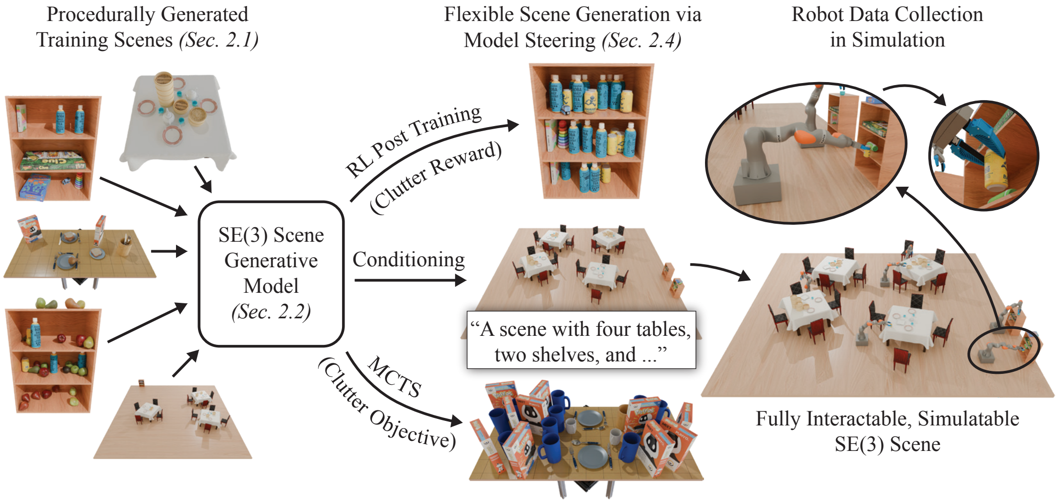

Procedurally Generated Training Scenes *(Sec. 2.1)*
SE(3) Scene Generative Model *(Sec. 2.2)*
Flexible Scene Generation via Model Steering *(Sec. 2.4)*
RL Post Training (Clutter Reward)
Conditioning
MCTS (Clutter Objective)
"A scene with four tables, two shelves, and ..."
Robot Data Collection in Simulation
Fully Interactable, Simulatable SE(3) Scene

Figure 1: **Overview of our approach.** We train a diffusion-based generative model on SE(3) scenes generated by procedural models, then adapt it to downstream objectives via reinforcement learning-based post training, conditional generation, or inference-time search. The resulting scenes are physically feasible and fully interactable. We demonstrate teleoperated interaction in a subset of generated scenes using a mobile KUKA iiwa robot.

Abstract:

Training robots in simulation requires diverse 3D scenes that reflect the specific challenges of downstream tasks. However, scenes that satisfy strict task requirements, such as high-clutter environments with plausible spatial arrangement, are rare and costly to curate manually. Instead, we generate large-scale scene data using procedural models that approximate realistic environments for robotic manipulation, and adapt it to task-specific goals. We do this by training a unified diffusion-based generative model that predicts which objects to place from a fixed asset library, along with their SE(3) poses. This model serves as a flexible scene prior that can be adapted using reinforcement learning-based post training, conditional generation, or inference-time search, steering generation toward downstream objectives even when they differ from the original data distribution. Our method enables goal-directed scene synthesis that respects physical feasibility and scales across scene types. We introduce a novel MCTS-based inference-time search strategy for diffusion models, enforce feasibility via projection and simulation, and release a dataset of over 44 million SE(3) scenes spanning five diverse environments. Website with videos, code, data, and model weights: https://steerable-scene-generation.github.io/

Keywords: Scene Generation, Simulation, Diffusion, MCTS

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

1 Introduction

Robots increasingly rely on data-intensive learning methods, making simulation a promising strategy for scalable training and evaluation [1, 2, 3, 4, 5]. As robotics shifts toward foundation models, it is encouraging that the demand for large and diverse training datasets will only increase [6, 7, 8]. However, acquiring scenes that meaningfully challenge robot capabilities or reflect human teleoperator preferences remains difficult, as such scenes are rare, expensive to curate, and task-specific. For example, a robot may need to operate in highly cluttered environments or interact with specific object categories. Instead of manually authoring such scenes, we propose training a unified generative model on large-scale procedurally generated data and adapting it to downstream objectives using reinforcement learning-based post training, conditional generation, and inference-time search.

Recent work has advanced automatic scene creation at both the object [9, 10] and scene level [11, 12]. We focus on the latter, where the task is to select objects from a fixed library and place them at continuous SE(3) poses. Classical approaches to scene synthesis rely on procedural modeling, where object relationships are encoded as rule sets or grammars [13, 14, 15, 16, 17, 18, 19]. Recent works incorporate priors from large language models (LLMs) or vision-language models (VLMs) [20, 21, 22, 23]. Others aim to extract 3D scenes directly from 2D images [24, 25, 26], moving toward generating large-scale 3D datasets from internet-scale image corpora. A separate line of work trains generative models that learn object relationships directly from scene data, without relying on handcrafted rules or LLMs [11, 27, 28, 29, 30, 31]. These models typically operate in SE(2), assuming floor-aligned layouts composed of large furniture items. We combine the strengths of both directions by treating procedural and image-to-3D pipelines as data sources for training a generative scene model. Rather than using these pipelines at inference time, we distill their output into a flexible prior that can be adapted to downstream tasks. Our framework is agnostic to the (object ID, SE(3) pose) data source and can be augmented with real-world scenes when available.

Prior generative models often represent scenes as floor-aligned SE(2) layouts and focus on static furniture arrangements [27, 28, 29, 30, 31]. In contrast, we target cluttered SE(3) scenes composed of small, manipulable objects relevant to robotic manipulation. Many such scenes require vertical translation (e.g., placing an object on a shelf) and full 3D rotation (e.g., standing cutlery in a utensil crock), which SE(2) cannot represent. These manipulation-ready settings demand physically feasible placements, including non-penetration and static equilibrium. PhyScene [11] encourages such feasibility through classifier-based guidance, but may still produce invalid samples. In contrast, we guarantee physical correctness via a nonlinear programming projection and simulation.

In practice, the distribution of available training data often does not reflect downstream objectives, such as maximizing robot performance or aligning with human preferences. While diffusion models are typically trained to maximize likelihood under the training distribution, this is insufficient when the data does not cover task-relevant domains. We study three complementary strategies for steering a pretrained scene model toward downstream goals. First, we adopt reinforcement learning-based post training, which has been applied in NLP and vision to optimize for user preferences [32, 33, 34, 35], but remains unexplored for scene generation. Second, we explore conditional generation, widely used in SE(2) scene models, in the SE(3) setting. Third, we introduce an inference-time Monte Carlo Tree Search (MCTS) procedure over partial scenes. Together, these tools enable goal-directed scene generation beyond the support of the original training distribution.

We evaluate our generative model pipeline on five scene types ranging from tabletop to room-scale environments, compare against SE(2)-based baselines extended to SE(3), and show that the generated scenes can be used directly for robot data generation. We demonstrate post training and inference-time search using physical feasibility and high-clutter objectives relevant to robotics [36].

Summary of contributions. Our main contribution is showing that a scene generative model trained on broad procedural data can be steered toward task-specific objectives, such as increasing clutter. Specifically: (1) we demonstrate how this can be achieved through reinforcement learning-based post training, conditional generation, and a novel MCTS-based inference-time search strategy for diffusion models; (2) we release our code, data, and model weights; and (3) we present a dataset comprising over 44 million unique SE(3) scenes spanning five distinct scene types, each featuring numerous small, movable objects. Individual scenes include up to 125 objects, supporting diverse and complex interaction scenarios relevant to robotic tasks, and providing a valuable benchmark for future work on SE(3) scene generation.

2 SE(3) Scene Generation and Steering

We learn a generative model over scenes, where each object is selected from a known library and placed at a continuous SE(3) pose. Our method begins with data from a procedural generator (Section 2.1), trains a diffusion model (Section 2.2), and applies post processing to ensure physical feasibility (Section 2.3). To steer the pretrained model toward downstream goals, we explore reinforcement learning post training, conditional generation, and inference-time search (Section 2.4).

2.1 Data Generation

We train on procedurally generated scenes, but our method is agnostic to the scene generator and supports any source that outputs (object, pose) tuples. This includes future procedural pipelines as well as real-world scene data. While large-scale real-world SE(3) datasets remain scarce, building them from internet-scale image or video corpora is a promising direction [24, 25, 26]. In this work, we use a single procedural model [16] to generate training data. Distilling that data into a generative model yields a compact, unified, and differentiable scene prior that enables post training, conditional generation, and inference-time search—capabilities not easily supported by procedural systems. We provide additional data generation details in the appendix.

2.2 SE(3) Scene Diffusion

Scene Representation. We represent a scene as an unordered object set $\mathcal{X} = \{\mathbf{o}_i \mid i \in \{1, \ldots, N\}\}$, where N is an upper bound on the number of objects [30]. Each object $\mathbf{o}_i$ consists of an SE(3) pose, parameterized by a translation $\mathbf{p} \in \mathbb{R}^3$ and a rotation, represented as a 9D vector $\mathbf{R} \in \mathbb{R}^9$. While this rotation representation is used during training and diffusion, we project it onto SO(3) at sample time as in [37]. Each object also includes a one-hot vector $\mathbf{c} \in \{v \in \{0,1\}^C \mid \sum_i v_i = 1\}$, indexing a specific object asset from a fixed library $\mathcal{S}$ of C assets. Following [30], we include an empty object in $\mathcal{S}$ to support variable-sized scenes. Our generative model learns distributions over such object sets $\mathcal{X}$.
Training Objective. We adopt the mixed discrete-continuous diffusion framework from [31]. Specifically, we apply continuous diffusion [38] to $\mathbf{p}$ and $\mathbf{R}$ and discrete diffusion [39] to $\mathbf{c}$, conditioning each on the other during generation.
Model Architecture. Since we represent scenes as object sets $\mathcal{X}$, the denoising model f should be object-order equivariant [30]: for any permutation $\sigma(\cdot)$, it should satisfy $f(\sigma(\mathcal{X})) = \sigma(f(\mathcal{X}))$ [40]. Standard Transformers satisfy this property when positional encodings are omitted [41]. We adopt the Flux architecture [42], using its image branch without positional encodings to preserve equivariance. Flux offers efficient training and strong performance across domains such as images and music [42, 43]. For mixed diffusion, we add input/output MLP projections following [31].

2.3 Physical Feasibility Post Processing

Even when trained on feasible data, generative models may produce SE(3) scenes that violate physical constraints, such as non-penetration or static equilibrium. These issues often arise from small numerical errors, e.g., due to mixed precision or slight misalignments. To enforce physical feasibility, we first resolve inter-object collisions by projecting continuous object translations to the nearest collision-free configuration while keeping orientations fixed to help preserve static equilibrium (see the appendix for details). We then simulate the scene in Drake [44], allowing unstable objects to settle under gravity. While projection removes penetrations, simulation corrects unstable configurations by adjusting full object poses, ensuring scenes are physically plausible and ready for downstream use. We apply simulation only after projection, as deep penetrations can cause large contact forces under rigid-body models, leading to explosive behavior.

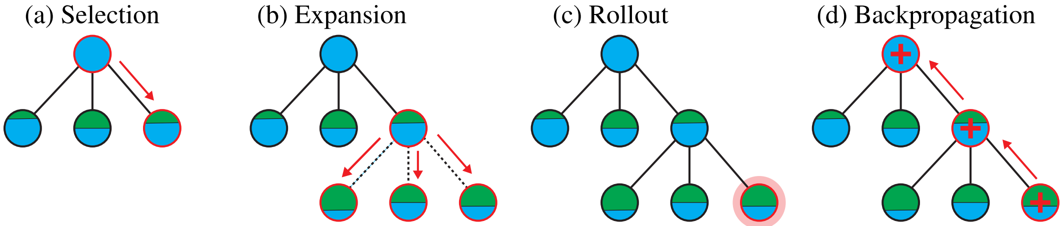


Figure 2: **Our MCTS inference-time search.** The root node is fully masked (blue), and child nodes represent partially inpainted scenes (blue-green). The rollout node is highlighted with a red halo.

2.4 Steering Scene Generative Models Toward Downstream Objectives

A key capability of scene generative models is their potential for steering generation toward downstream goals, even beyond the training distribution. We explore three complementary strategies: reinforcement learning-based post training (Section 2.4.1), conditional generation (Section 2.4.2), and inference-time search (Section 2.4.3). The appendix contains additional details and a comparison of different steering methods.

2.4.1 Post Training with Reinforcement Learning

Distilling scene datasets into a differentiable model enables reinforcement learning (RL)–based post training. We adopt DDPO [34] to fine-tune a continuous DDPM-based scene model [38] using task-specific rewards and apply the regularization from [35] to stabilize training. We use object count (clutter) as a downstream reward to test whether RL-based post training can adapt the model beyond the training distribution. To enable compatibility with existing fine-tuning methods, we use a fully continuous diffusion model, representing object categories and poses as continuous variables as in [30]. Our aim is not to propose a new RL algorithm but to demonstrate the *feasibility and utility* of post training for scene models.

2.4.2 Conditional Generation

Learned generative models support flexible conditioning, unlike most procedural systems [16, 17, 18]. Our models can be conditioned on language, partial scenes, or other modalities. We explore two strategies: (1) conditional training and (2) test-time inpainting using an unconditional model.

Text-conditioned generation. We encode prompts using BERT [45] and inject the resulting embeddings into the conditional branch of our Flux-based architecture. To enable a single model to support both conditional and unconditional generation, we randomly mask the conditioning information during training. This allows us to apply classifier-free guidance (CFG) [46] at inference time by interpolating between the conditional and unconditional outputs. Our prompts are procedurally generated and can describe object counts, object identities, or spatial relationships between objects.

Scene completion and re-arrangement. We perform inpainting directly in the structured scene representation. Given a binary inpainting mask indicating which parts of the scene to synthesize, we generate missing content while clamping the rest to their fixed values during the reverse diffusion process [47]. For example, we can rearrange scenes by regenerating the continuous poses while keeping the object categories fixed. For scene completion, we synthesize both categories and poses for empty objects. This enables consistent, plausible generation from partial inputs.

2.4.3 Inference-Time Search via MCTS

Generative scene models can be steered toward downstream objectives at inference time. We demonstrate this via a Monte Carlo Tree Search (MCTS) procedure that incrementally constructs a scene through conditional inpainting. At each step, an inpainting mask identifies which objects to regenerate, such as unstable ones or empty slots, and a reward function evaluates the resulting scene. As a running example, we consider the objective of maximizing the number of physically feasible objects, i.e., objects that are non-penetrating and in static equilibrium. Each node in the MCTS tree represents a partially completed scene and a corresponding inpainting mask.

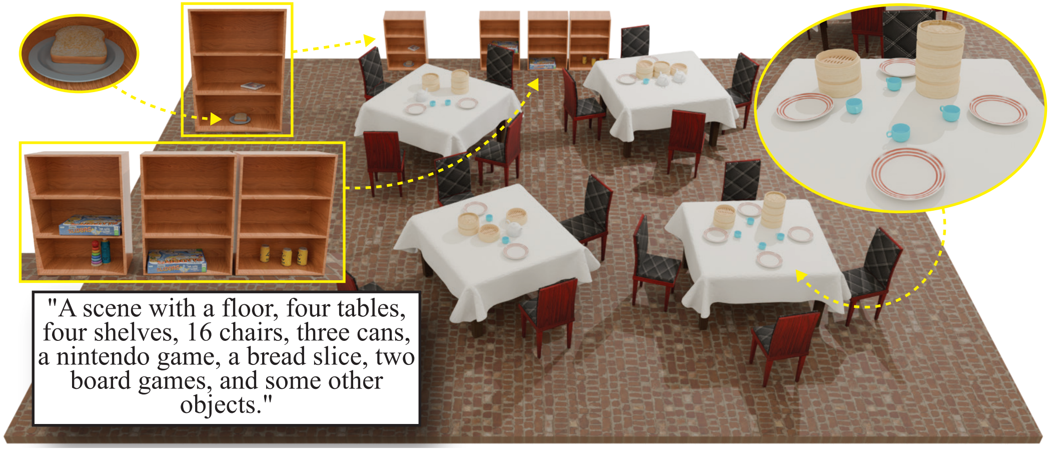

"A scene with a floor, four tables, four shelves, 16 chairs, three cans, a nintendo game, a bread slice, two board games, and some other objects."

Figure 3: **Text-conditioned scene generation.** A model trained on the Restaurant (High-Clutter) dataset is queried with the shown text prompt. The generated scene matches both the large-scale layout and fine-grained object details.

The search proceeds through the standard MCTS phases [48] (shown in Figure 2):
(a) Selection. We traverse the tree from root to leaf, selecting at each step the child with the highest UCT [49] value: $\mathrm{UCT}(j) = \bar{r}_j + c \cdot \sqrt{\frac{2 \ln n_{\text{parent}}(j)}{n_j}}$, where $\bar{r}_j$ is the average reward of child j, $n_{\text{parent}}(j)$ and n_j are the visit counts of parent and child, and c is an exploration constant.
(b) Expansion. At a leaf, we sample B completions, where B is the branching factor, by inpainting the masked objects with different noise initializations. Each resulting scene is evaluated to identify remaining invalid or incomplete objects, producing a new inpainting mask (e.g., flagging newly unstable or empty objects). These (scene, mask) pairs form new child nodes.
(c) Rollout. One of the new children is selected randomly and scored using a task-specific reward, in our example, the number of physically feasible objects. Since each node corresponds to a complete scene (when discarding masked objects), rollout in our setting reduces to directly reading the reward, rather than “rolling out“ to a terminal state.
(d) Backpropagation. The reward is propagated up the tree, updating average reward estimates and visit counts along the way.

We run the search for a fixed number of iterations or until a scene with no masked objects is found. If no such fully valid scene is produced, we return the best partial scene encountered, discarding any objects that remain masked.

Controlling the Objective. Our framework adapts to diverse downstream goals through two modular components: the *mask generator*, which determines which objects to inpaint, and the *reward function*, which evaluates scene quality. These components can be defined independently. For example, one might mask all physically invalid or empty objects, but optimize for a more targeted, yet aligned reward, such as the number of edible objects or the degree of prompt alignment. This decoupling enables flexible and modular search strategies across various downstream objectives.
Connection to Prior Work. When the branching factor $B = \infty$, our method reduces to *Random Search* from Ma et al. [50], repeatedly sampling new scenes without building on previous ones.

3 Experimental Evaluation

3.1 Evaluation Setup

Metrics. We evaluate generative quality using image-based metrics adapted to SE(3) scenes. Following prior work on SE(2) scenes [27, 28, 30, 31], we compute Fréchet Inception Distance (FID) and classifier accuracy (CA, in %) based on semantic renderings. A CA near 50% indicates realistic generation, while a CA near 100% indicates clear separability. We render each scene from a manu-

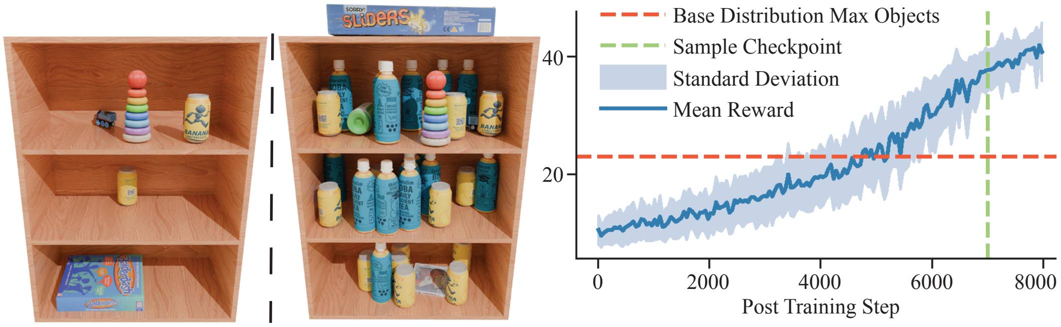


Figure 4: **RL post training with an object count reward.** We post-train a model originally trained on the Living Room Shelf dataset. *Left*: Sample before post training. *Middle*: Sample after post training. *Right*: Reward curve. The red line marks the maximum number of objects seen during pre-training (23). Before post training, we increase the maximum number of objects allowed by the scene representation by 20 to enable higher object counts. The green line indicates the checkpoint used for sampling (step 7000), chosen to avoid overoptimization.

ally defined informative viewpoint specific to the scene type. We also report KL divergence between object category distributions, prompt-following accuracy (APF) for count and object-type prompts, and median total penetration (MTP) to assess physical feasibility. MTP is computed before applying our post processing. Full metric definitions are provided in the appendix.

Baselines. We compare our proposed approach against two state-of-the-art diffusion-based scene synthesis methods: (1) DiffuScene [30], which uses a 1D U-Net with a continuous DDPM objective, and (2) MiDiffusion [31], a Transformer-based model that employs a mixed discrete-continuous diffusion objective. We apply minimal modifications to both implementations to support our scene representation $\mathcal{X}$. For MiDiffusion, we replace floor plan conditioning with text conditioning. All models, including ours, use the same BERT text encoder [45].

Datasets. As described in Section 2.1, we generate our training data using the procedural scene generation framework from [16]. We reuse the *Dimsum Table* scene type from [16] and define four additional scene types: *Breakfast Table*, *Living Room Shelf*, *Pantry Shelf*, and *Restaurant*. *Restaurant* is a room-level composition that integrates *Dimsum Table* and *Living Room Shelf* scenes along with additional objects. For greater diversity, we split the *Breakfast Table* and *Restaurant* scenes into low- and high-clutter variants, reflecting the procedural generation parameters used. In total, we sample more than 44 million SE(3) scenes across all scene types, significantly surpassing the scale of prior SE(2) scene datasets, such as 3D-FRONT [51], which contains 18,968 scenes. The appendix provides the full set of quantitative and qualitative results across all datasets.

3.2 Unconditional Generation

Table 1: Unconditional generation results on the Restaurant (High-Clutter) and Living Room Shelf datasets. * indicates that we adjusted the methods for compatibility with our scene representation.

Method	Restaurant (High-Clutter Variant)				Living Room Shelf			
	CA (50 it, %) ↓	KL ($\times 10^4$) ↓	FID ↓	MTP (cm) ↓	CA (100 it, %) ↓	KL ($\times 10^4$) ↓	FID ↓	MTP (cm) ↓
DiffuScene* [30]	84.81 ± 6.49	**0.55**	1.39	18.11	71.73 ± 0.99	4.67	2.18	0.05
MiDiffusion* [31]	78.63 ± 9.79	1.01	1.34	8.80	64.13 ± 1.87	2.51	**2.09**	0.03
Ours	**70.74 ± 8.05**	0.87	**1.31**	**6.31**	**52.84 ± 1.26**	**2.13**	**2.09**	**0.02**

We report unconditional generation results for the Restaurant (High-Clutter) and Living Room Shelf datasets in Table 1; additional results, including samples from a single model jointly trained across all datasets, are provided in the appendix. Rather than training separate unconditional models, we use our text-conditioned models by providing empty conditioning inputs at sampling time. Our model achieves strong FID and significantly lower MTP compared to baselines, indicating that it produces scenes that are both visually realistic and physically plausible. Classifier accuracy (CA)

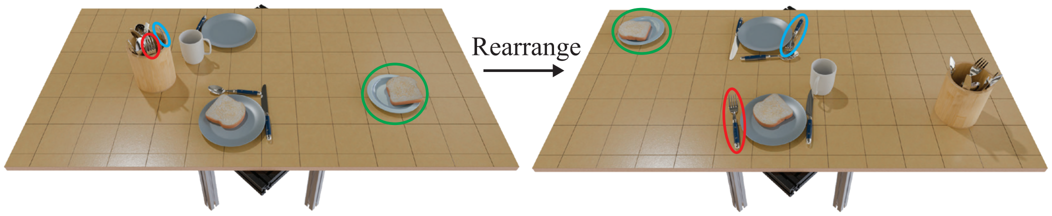
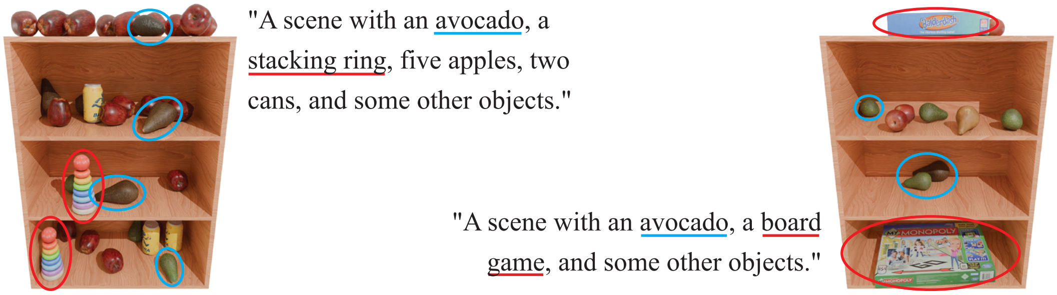


Figure 5: **Scene rearrangement example.** A scene from the Restaurant (Low-Clutter) dataset is rearranged via inpainting by a model trained on the same dataset. Red, green, and blue ellipses highlight corresponding objects. Notably, cutlery is moved from the utensil crock to the table, requiring full SO(3) rotation modeling.



"A scene with an avocado, a stacking ring, five apples, two cans, and some other objects."

"A scene with an avocado, a board game, and some other objects."

Figure 6: **Interpolation between Living Room and Pantry Shelf Scenes.** We train a joint model on both datasets with a 50/50 batch mix. By prompting for objects unique to each dataset (red = Living Room Shelf, blue = Pantry Shelf), we guide the model to generate interpolated scenes.

closer to 50% further supports that our samples are harder to distinguish from dataset scenes. While we do not always achieve the lowest KL divergence, all methods obtain very low KL values on our datasets. Since KL is near saturation, the differences are minor, and this metric is less informative in our setting; therefore, we report it primarily for completeness, following prior work.

3.3 Post Training with Reinforcement Learning

We apply reinforcement learning (RL) post training to a model trained on the Living Room Shelf dataset, using an object count reward. Figure 4 shows the reward curve and sample scenes before and after post training. We choose a checkpoint before overoptimization occurs to maintain scene quality. Additional results are provided in the appendix. RL-based post training successfully adapts the pretrained model to generate scenes with object counts substantially exceeding those observed during pretraining. By expanding the maximum object capacity in the scene representation before post training, we enable the model to extrapolate beyond its original range without requiring re-training from scratch. This demonstrates that post training can effectively shift and reshape scene distributions toward task-specific goals.

3.4 Conditional Generation

Table 2: Conditional generation results on the Breakfast Table (High-Clutter) and Pantry Shelf datasets. * indicates that we adjusted the methods for compatibility with our scene representation.

Method	Breakfast Table (High-Clutter Variant)				Pantry Shelf			
	CA (50 it, %) ↓	KL ($\times 10^4$) ↓	FID ↓	APF ↑	CA (50 it, %) ↓	KL ($\times 10^4$) ↓	FID ↓	APF ↑
DiffuScene* [30]	82.38 ± 3.82	0.96	1.87	0.76	84.65 ± 2.23	0.91	1.93	0.88
MiDiffusion* [31]	82.22 ± 3.11	0.58	1.93	0.60	87.24 ± 1.80	**0.64**	1.89	0.72
Ours	**68.44 ± 4.67**	**0.30**	**1.84**	**0.86**	**82.78 ± 3.44**	0.66	**1.88**	**0.98**

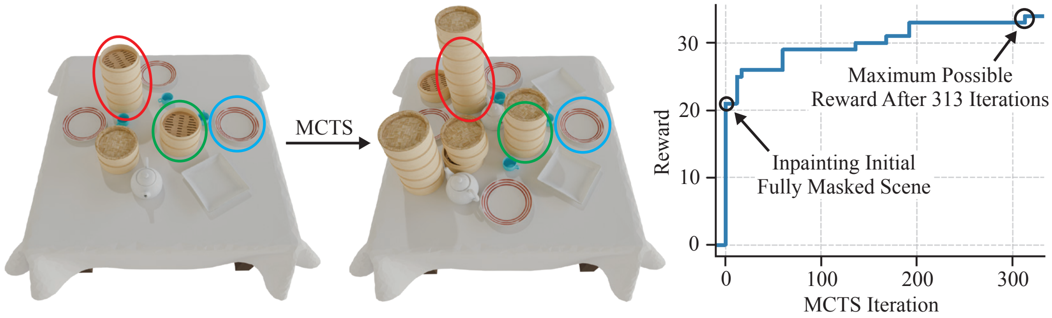


Figure 7: **Inference-time MCTS.** We apply MCTS at inference time to generate a Dimsum scene that maximizes the number of physically feasible objects. *Left*: Initial sample and final result after search. Red, green, and blue ellipses highlight corresponding objects. Note how the search completes the steamer stacks. *Right*: Reward curve. Inpainting the fully masked scene (equivalent to unconditional sampling) yields 21 feasible objects in the best of $B = 3$ samples. MCTS reaches the maximum possible 34 objects after 313 iterations, with reward rising quickly, then plateauing.

We report quantitative results for text-conditioned generation on the Breakfast Table (High-Clutter) and Pantry Shelf datasets in Table 2. Figure 3 shows examples of text-conditioned generation, and Figure 5 illustrates scene rearrangement via partial inpainting. Additional results, including scene completion, are provided in the appendix. Our model outperforms baselines in CA, FID, and APF, indicating stronger prompt adherence and overall generation quality. Qualitative examples further show that our model captures both large-scale layouts and fine-grained object details.

Cotraining across scene types. We also investigate whether cotraining on the Living Room Shelf and Pantry Shelf datasets enables interpolation. During training, we use equal batch mixing ratios across sub-datasets. As shown in Figure 6, prompting the model with mixed object descriptions from both environments leads to interpolated scenes that combine elements of each dataset, demonstrating that the model captures a meaningful joint distribution.

3.5 Inference-Time Search

Figure 7 shows how MCTS optimizes the number of physically feasible objects in a Dimsum scene with a branching factor $B = 3$. The training set had a mean of 17.1 objects and a maximum of 34; MCTS reaches this maximum, demonstrating that inference-time search can push scene complexity well beyond typical training-time levels. Scene distributions encode local structure, for example, steamers often appear in vertical stacks. Our results show that the model captures such patterns: MCTS incrementally builds realistic, physically feasible stacks by exploiting inductive biases learned during pretraining, without requiring retraining.

4 Conclusion

We presented a diffusion-based framework for SE(3) scene generation that distills large-scale procedural data into a flexible, physically grounded prior. Our model predicts object categories from a fixed asset library and continuous poses, and supports adaptation via RL-based post training, conditional generation, and inference-time search. Experiments across five scene types show that the pretrained model enables strong unconditional and conditional generation, that post training improves targeted metrics such as clutter, and that MCTS search can optimize task rewards without retraining. To qualitatively validate simulation readiness, we imported generated scenes into the Drake simulator and successfully teleoperated a mobile KUKA iiwa robot to perform pick-and-place interactions without requiring manual scene corrections (see Figure 1 and supplementary videos). Together, these results demonstrate that a single model can flexibly adapt scene distributions without handcrafted tuning or retraining. Our work highlights a scalable approach to robotic scene generation: pretraining on broad data sources, then steering toward task-specific goals.

5 Limitations

While our method demonstrates the feasibility and benefits of steering scene generative models toward downstream objectives, several limitations remain. First, although procedural data provides scalable supervision, it may not fully capture the complexity and variability of real-world environments. Incorporating real-world SE(3) datasets, potentially extracted from internet-scale image or video corpora, remains an important direction for enhancing realism. Second, we adopt fully continuous diffusion models to enable reinforcement learning-based post training, rather than our full mixed discrete-continuous models. We leave applying post training to mixed discrete-continuous settings as future work. Additionally, we observe that when post training pushes object count close to the maximum allowed by the scene representation, overoptimization can occur: samples exhibit many objects but no longer resemble the original data distribution. While expanding the maximum object capacity helps, fully continuous models still struggle to maintain quality when handling many additional object tokens, limiting the effectiveness of this strategy. Third, our object representation uses a fixed asset library with one-hot encodings, reflecting a practical design choice aligned with robotics workflows, which often depend on pre-validated simulation assets to ensure high-quality geometry and physical properties for realistic simulations. While this limits generalization to novel geometries without retraining, it enables precise control over the asset set, and our steering methods remain compatible with alternative object representations. Fourth, while our object library currently consists of rigid bodies, it naturally extends to articulated objects (e.g., drawers, cabinets) without requiring changes to the method. We leave the exploration of articulated scenes for future work. Fifth, our adaptation strategies—post training, conditional generation, and inference-time search—are proof-of-concept demonstrations. Future work could explore more sophisticated reward functions, conditioning schemes, and search objectives tailored to specific robot tasks. Finally, while we demonstrate simulation-readiness via teleoperation, scaling to large-scale autonomous robot training across generated scenes is an important direction for future work.

Acknowledgments

This work was partially supported by Amazon.com, PO No. 2D-15693043. The Toyota Research Institute partially supported this work. This article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity. We thank Rick Cory, Ge Yang, Phillip Isola, Adam Wei, and Jeremy Binagia for fruitful discussions.

References

[1] H. Choi, C. Crump, C. Duriez, A. Elmquist, G. Hager, D. Han, F. Hearl, J. Hodgins, A. Jain, F. Leve, C. Li, F. Meier, D. Negrut, L. Righetti, A. Rodriguez, J. Tan, and J. Trinkle. On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. *Proceedings of the National Academy of Sciences*, 118(1):e1907856118, 2021. doi:10.1073/pnas.1907856118. URL https://www.pnas.org/doi/abs/10.1073/pnas.1907856118.

[2] W. Zhao, J. P. Queralta, and T. Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744, 2020. doi:10.1109/SSCI47803.2020.9308468.

[3] A. Wei, A. Agarwal, B. Chen, R. Bosworth, N. Pfaff, and R. Tedrake. Empirical analysis of sim-and-real cotraining of diffusion policies for planar pushing from pixels, 2025. URL https://arxiv.org/abs/2503.22634.

[4] A. Maddukuri, Z. Jiang, L. Y. Chen, S. Nasiriany, Y. Xie, Y. Fang, W. Huang, Z. Wang, Z. Xu, N. Chernyadev, S. Reed, K. Goldberg, A. Mandlekar, L. Fan, and Y. Zhu. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation, 2025. URL https://arxiv.org/abs/2503.24361.

[5] C. Eppner, A. Mousavian, and D. Fox. Acronym: A large-scale grasp dataset based on simulation, 2020. URL https://arxiv.org/abs/2011.09584.

[6] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman, B. Ichter, D. Driess, J. Wu, C. Lu, and M. Schwager. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research*, 2024.

[7] E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Bharadhwaj, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Vakil, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Tulsiani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Kumar, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2024. URL https://arxiv.org/abs/2310.08864.

[8] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, V. Guizilini, D. A. Herrera, M. Heo, K. Hsu, J. Hu, M. Z. Irshad, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset, 2025. URL https://arxiv.org/abs/2403.12945.

[9] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, page 2553–2560. IEEE Press, 2022.

[10] N. Pfaff, E. Fu, J. Binagia, P. Isola, and R. Tedrake. Scalable real2sim: Physics-aware asset generation via robotic pick-and-place setups, 2025. URL https://arxiv.org/abs/2503.00370.

[11] Y. Yang, B. Jia, P. Zhi, and S. Huang. Physcene: Physically interactable 3d scene synthesis for embodied ai. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[12] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024.

[13] P. Merrell, E. Schkufza, Z. Li, M. Agrawala, and V. Koltun. Interactive furniture layout using interior design guidelines. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450309431. doi:10.1145/1964921.1964982. URL https://doi.org/10.1145/1964921.1964982.

[14] J. O. Talton, Y. Lou, S. Lesser, J. Duke, R. Měch, and V. Koltun. Metropolis procedural modeling. *ACM Trans. Graph.*, 30(2), Apr. 2011. ISSN 0730-0301. doi:10.1145/1944846.1944851. URL https://doi.org/10.1145/1944846.1944851.

[15] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] G. Izatt and R. Tedrake. *Capturing Distributions over Worlds for Robotics with Spatial Scene Grammars*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2022. URL https://dspace.mit.edu/handle/1721.1/144763.

[17] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, J. Salvador, K. Ehsani, W. Han, E. Kolve, A. Farhadi, A. Kembhavi, and R. Mottaghi. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In *NeurIPS*, 2022. Outstanding Paper Award.

[18] Y. Lin, J. Humplik, S. H. Huang, L. Hasenclever, F. Romano, S. Saliceti, D. Zheng, J. E. Chen, C. Barros, A. Collister, M. Young, A. Dostmohamed, B. Moran, K. Caluwaerts, M. Giustina, J. Moore, K. Connell, F. Nori, N. Heess, S. Bohez, and A. Byravan. Proc4gem: Foundation models for physical agency through procedural generation, 2025. URL https://arxiv.org/abs/2503.08593.

[19] L. Wang, Y. Ling, Z. Yuan, M. Shridhar, C. Bao, Y. Qin, B. Wang, H. Xu, and X. Wang. Gensim: Generating robotic simulation tasks via large language models, 2023. URL https://arxiv.org/abs/2310.01361.

[20] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation, 2024. URL https://arxiv.org/abs/2311.01455.

[21] P. Katara, Z. Xian, and K. Fragkiadaki. Gen2sim: Scaling up robot learning in simulation with generative models, 2023. URL https://arxiv.org/abs/2310.18308.

[22] T.-Y. Lin, C.-H. Lin, Y. Cui, Y. Ge, S. Nah, A. Mallya, Z. Hao, Y. Ding, H. Mao, Z. Li, Y.-C. Lin, X. Zeng, Q. Zhang, D. Xiang, Q. Ma, J. Lewis, J. Jin, P. Jannaty, and M.-Y. Liu. Genusd: 3d scene generation made easy. In *ACM SIGGRAPH 2024 Real-Time Live!*, SIGGRAPH '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705267. doi:10.1145/3641520.3665306. URL https://doi.org/10.1145/3641520.3665306.

[23] H. I. D. Pun, H. I. I. Tam, A. T. Wang, X. Huo, A. X. Chang, and M. Savva. Hsm: Hierarchical scene motifs for multi-scale indoor scene generation, 2025. URL https://arxiv.org/abs/2503.16848.

[24] Z. Chen, A. Walsman, M. Memmel, K. Mo, A. Fang, K. Vemuri, A. Wu, D. Fox, and A. Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv preprint arXiv:2405.11656*, 2024.

[25] K. Yao, L. Zhang, X. Yan, Y. Zeng, Q. Zhang, L. Xu, W. Yang, J. Gu, and J. Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image, 2025. URL https://arxiv.org/abs/2502.12894.

[26] P. Engstler, A. Shtedritski, I. Laina, C. Rupprecht, and A. Vedaldi. Syncity: Training-free generation of 3d worlds, 2025. URL https://arxiv.org/abs/2503.16420.

[27] D. Paschalidou, A. Kar, M. Shugrina, K. Kreis, A. Geiger, and S. Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[28] X. Wang, C. Yeshwanth, and M. Nießner. Sceneformer: Indoor scene generation with transformers. In *2021 International Conference on 3D Vision (3DV)*, pages 106–115, 2021. doi:10.1109/3DV53792.2021.00021.

[29] Q. A. Wei, S. Ding, J. J. Park, R. Sajnani, A. Poulenard, S. Sridhar, and L. Guibas. LEGO-Net: Learning Regular Rearrangements of Objects in Rooms . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19037–19047, Los Alamitos, CA, USA, June 2023. IEEE Computer Society. doi:10.1109/CVPR52729.2023.01825. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.01825.

[30] J. Tang, Y. Nie, L. Markhasin, A. Dai, J. Thies, and M. Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[31] S. Hu, D. M. Arroyo, S. Debats, F. Manhardt, L. Carlone, and F. Tombari. Mixed diffusion for 3d indoor scene synthesis. *arXiv preprint: 2405.21066*, 2024.

[32] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

[33] K. Kumar, T. Ashraf, O. Thawakar, R. M. Anwer, H. Cholakkal, M. Shah, M.-H. Yang, P. H. S. Torr, S. Khan, and F. S. Khan. Llm post-training: A deep dive into reasoning large language models, 2025. URL https://arxiv.org/abs/2502.21321.

[34] K. Black, M. Janner, Y. Du, I. Kostrikov, and S. Levine. Training diffusion models with reinforcement learning, 2023.

[35] Y. Zhang, E. Tzeng, Y. Du, and D. Kislyuk. Large-scale reinforcement learning for diffusion models, 2024. URL https://arxiv.org/abs/2401.12244.

[36] Y. Jia and B. Chen. Cluttergen: A cluttered scene generator for robot learning. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=k0ogr4dnhG.

[37] A. R. Geist, J. Frey, M. Zhobro, A. Levina, and G. Martius. Learning with 3D rotations, a hitchhiker's guide to SO(3). In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 15331–15350. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/geist24a.html.

[38] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[39] J. Austin, D. D. Johnson, J. Ho, D. Tarlow, and R. van den Berg. Structured denoising diffusion models in discrete state-spaces, 2023. URL https://arxiv.org/abs/2107.03006.

[40] M. Zaheer, S. Kottur, S. Ravanbhakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola. Deep sets. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3394–3404, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[41] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3744–3753, 2019.

[42] B. F. Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

[43] Z. Fei, M. Fan, C. Yu, and J. Huang. FLUX that plays music, 2024. URL https://arxiv.org/abs/2409.00587.

[44] R. Tedrake and the Drake Development Team. Drake: Model-based design and verification for robotics, 2019.

[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

[46] J. Ho and T. Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

[47] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL https://arxiv.org/abs/1503.03585.

[48] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012. doi:10.1109/TCIAIG.2012.2186810.

[49] L. Kocsis and C. Szepesvári. Bandit based monte-carlo planning. In J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, editors, *Machine Learning: ECML 2006*, pages 282–293, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-46056-5.

[50] N. Ma, S. Tong, H. Jia, H. Hu, Y.-C. Su, M. Zhang, X. Yang, Y. Li, T. Jaakkola, X. Jia, and S. Xie. Inference-time scaling for diffusion models beyond scaling denoising steps, 2025. URL https://arxiv.org/abs/2501.09732.

[51] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.