

Steer2Grasp: Inference-Time Embodiment-Aware Steering for Diverse Physically Feasible Grasp Diffusion

Abstract—Current grasp diffusion models provide rich priors for generation, yet their object-centric approach can violate the kinematic and collision constraints imposed by the embodiment and the environment. Existing embodiment-aware methods primarily perform local corrections around generated grasps through gradient guidance or optimization, making it difficult to recover from fundamentally infeasible modes. We present Steer2Grasp, a training-free, embodiment-agnostic framework for inference-time grasp steering that adapts a frozen Cartesian grasp diffusion model using deployment-specific rewards. Through Feynman-Kac (FK) inspired particle reweighting and resampling, the method reallocates population mass from infeasible to high-reward grasp modes, enabling population-level mode transitions without modifying the pre-trained diffusion model or requiring differentiable constraints. The framework enables a unified treatment for single and dual arm grasping through reachability and collision aware rewards, followed by gradient free gripper level local refinement. Across diverse objects, robot embodiments, and constrained environments, our method substantially improves feasible grasp generation while maintaining proximity to the underlying grasp prior. Project Page: steer2grasp.github.io

I. INTRODUCTION

Manipulation of everyday objects requires not only identifying stable grasp configurations, but ensuring that they are executable by the robot(s). A grasp is *feasible* if it is force-closure stable, admits a valid robot joint configuration, and is collision-free with respect to the environment. With regard to the first criterion, early approaches demonstrated the effectiveness of analytic grasp metrics and learned grasp quality models [1], while more recent diffusion-based methods learn multimodal distributions over 6-DoF grasp poses [2], [3]. This paradigm has been extended to large-object and bimanual manipulation, with newer datasets [4], [5] enabling generation of coordinated grasp pairs using diffusion models such as DAGDiff [6] and transformer networks such as BiGraspFormer [7].

Despite this progress in generating stable and diverse grasps, most grasp synthesis methods remain largely *arm agnostic*: they operate in an object-centric representation and do not explicitly account for robot kinematics or deployment-time environmental constraints. As a result, a grasp that is geometrically stable in task space, may be unreachable by the robot, or may require a robot joint configuration that collides with surrounding obstacles. This problem is substantially harder in the bimanual setting, where feasibility is determined by the joint configurations of both arms, and grasps that are individually reachable and collision free may still violate inter arm or coordination constraints, making the jointly feasible set a small, sparse subset of full grasp space.

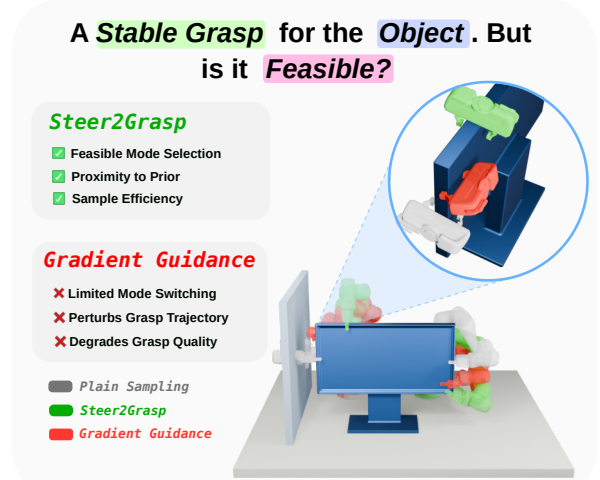


Fig. 1: Object-centric grasp models may generate force closure valid grasps that are unreachable or colliding for a specific robot embodiment configuration and environment. Local gradient guidance addresses this by perturbing individual denoising trajectories toward feasibility, but such local updates can degrade grasp quality and struggle to move between distant grasp modes. **Steer2Grasp** instead reweights and resamples the particle population, reallocating probability mass toward feasible modes while leaving the pretrained diffusion model unchanged.

Three natural strategies have emerged for closing this gap, each with a fundamental limitation. The first is to incorporate embodiment directly into training by filtering grasp data for a specific robot and scene prior to training [8], allowing the model to learn a feasibility-aware distribution from the outset. While effective, this ties the learned distribution to a particular embodiment, and deploying on a different robot or in a different scene requires collecting new data and retraining from scratch. The second is rejection filtering, leaving the prior unchanged and discarding generated grasps that violate kinematic or collision constraints. This avoids retraining but is inherently sample-inefficient, since the feasible set can constitute a small, disconnected subset of the full grasp space. Finally, recent inference time diffusion guidance methods [9]–[11] have shown that the iterative denoising process can be guided by task-specific objectives, enabling generated grasps to satisfy constraints without retraining the underlying prior. However, such guidance comes at the cost of modifying the learned grasp distribution: aggressive guidance can pull samples away from high-probability regions, potentially producing less stable or lower-quality grasps. Moreover, gradient guidance is fundamentally local: it can refine a sample toward a nearby feasible region, but cannot transport it across an infeasible region to reach a distant feasible mode, a critical

limitation when feasible grasps are sparse or disconnected, as is common for geometrically constrained objects and in bimanual grasping as shown in Fig. 1.

In this work, we introduce **Steer2Grasp**, an inference-time steering framework for zero-shot embodiment-aware grasp generation applicable to any grasp diffusion network. Rather than modifying the learned prior or post-hoc filtering, feasibility constraints are expressed as *potential functions* evaluated at intermediate denoising steps, which softly reweight the particle population and trigger resampling to concentrate probability mass on particles likely to yield feasible grasps. Unlike gradient guidance [9], which directly perturbs the denoising trajectory, embodiment constraints govern which particles survive and propagate, leaving individual denoising trajectories faithful to the learned prior. This also enables population-level mode reallocation in a way that local gradient steps cannot achieve, particles tracking infeasible modes are removed and replaced by resampling from survivors. To further reduce gripper collisions which are inherently local, we employ a refinement stage in the last stages of the steering that correct fine grained collisions. We validate Steer2Grasp with refinement on GraspGen [3] for single-arm grasping and DAGDiff [6] for bimanual grasping, improving feasibility over the base model by $\sim 32\text{pp}$ and $\sim 43\text{pp}$, and over gradient-based guidance by $\sim 22\text{pp}$ and $\sim 35\text{pp}$ respectively.

To summarize our contributions:

- 1) An inference-time steering framework that converts object-centric diffusion grasps into embodiment-aware ones, improving over baselines in both single- and dual-arm grasping. We are the first to extend embodiment-aware steering to dual-arm grasps, where feasibility must hold jointly across both arms.
- 2) A two-level steering mechanism that combines population-level mode reallocation, enabling transitions beyond local gradient corrections, with a final refinement stage that resolves fine-grained collisions.
- 3) Extensive simulation on the Franka Panda and UR5e across two priors (GraspGen and DAGDiff) and diverse task-space settings, with real-world demonstrations on the xArm7, showing our framework’s robustness.

II. RELATED WORK

A. Inference Time Alignment in Diffusion Models

Diffusion models provide flexible generative priors adaptable to downstream objectives during inference without modifying their learned parameters. Classifier and classifier-free guidance [12], [13] steer samples toward target conditions but require a differentiable classifier or conditions fixed at training time. Reward-guided methods [14], [15] extend this to arbitrary task rewards, but still guide with gradients of a learned reward or value model, while FK-Steering [16] removes the gradient requirement entirely via Feynman–Kac particle resampling with non-differentiable rewards. We leverage this paradigm for grasp generation by introducing deployment specific rewards induced by robot embodiment and environmental constraints.

B. Cross-Embodiment Generalization for Grasping

One aspect of cross-embodiment grasping is generalization across gripper embodiments [17], while arm-level kinematic and environmental feasibility are typically handled separately. [2] integrates learned grasp costs with collision and joint constraints for joint grasp and motion optimization, while projection-based approaches enforce feasibility by projecting samples onto constraint-satisfying regions during denoising [18]. Gradient-based approaches use reachability and arm–environment constraints to guide a pretrained arm-agnostic diffusion model toward feasible grasps [9]. [11] further formulates the inference-time diffusion guidance as a constrained non-linear optimization problem, replacing reverse-step perturbations with optimized corrections that enforce embodiment and task specific feasibility. However, these approaches primarily refine individual grasps toward local feasibility, which can fail when feasible grasps lie in a different mode of the grasp distribution. Our work instead steers a population of grasp candidates using embodiment and environment feedback, enabling transitions between feasible modes while maintaining closeness to the pretrained grasp prior.

III. PRELIMINARIES

We build on two grasp diffusion models as our base priors. For single-arm grasping, we use GraspGen [3], which generates 6-DoF grasp poses $H \in SE(3)$ conditioned on an object pointcloud P via DDPM [19] in $T = 10$ steps. For dual-arm grasping, we use DAGDiff [6], which jointly generates grasp pairs $H = (H_1, H_2) \in SE(3) \times SE(3)$ via score-based Langevin dynamics in $T = 225$ steps. In both cases, an estimate of the clean grasp at each denoising step is obtained via Tweedie’s formula [20], [21]. For GraspGen and DAGDiff, respectively,

$$\hat{H}_0(H_t) = \frac{1}{\sqrt{\bar{\alpha}_t}} (H_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(H_t, P, t)) \quad (1a)$$

$$\hat{H}_0(H_t) = \text{Expmap}_2(\text{Logmap}_2(H_t) + \sigma_t^2 s_\theta(H_t, P, t)) \quad (1b)$$

where t is the diffusion step, $\bar{\alpha}_t$ and σ_t are the respective noise schedules, ϵ_θ and s_θ denote the GraspGen noise predictor and DAGDiff score function, and Logmap_2 and Expmap_2 operate component-wise on $SE(3) \times SE(3)$. The Tweedie estimates $\hat{H}_0(H_t)$ play a central role in our method, as they provide a clean grasp prediction at each intermediate step against which feasibility rewards can be evaluated.

IV. METHOD

Given a point-cloud observation $P \in \mathbb{R}^{N_p \times 3}$ of an object, a robot embodiment $\mathcal{R}$, and a deployment environment $\mathcal{E}$, our goal is to generate a grasp (or a coordinated grasp pair) that remains close to the pretrained task-space grasp prior $p_{0:T}^\theta(H_{0:T} \mid P)$, while satisfying the environment and embodiment specific constraints, where H_t denotes the grasp state at diffusion timestep t , and T is the total number of diffusion steps. Importantly, the prior is independent of the robot embodiment and the deployment environment; these

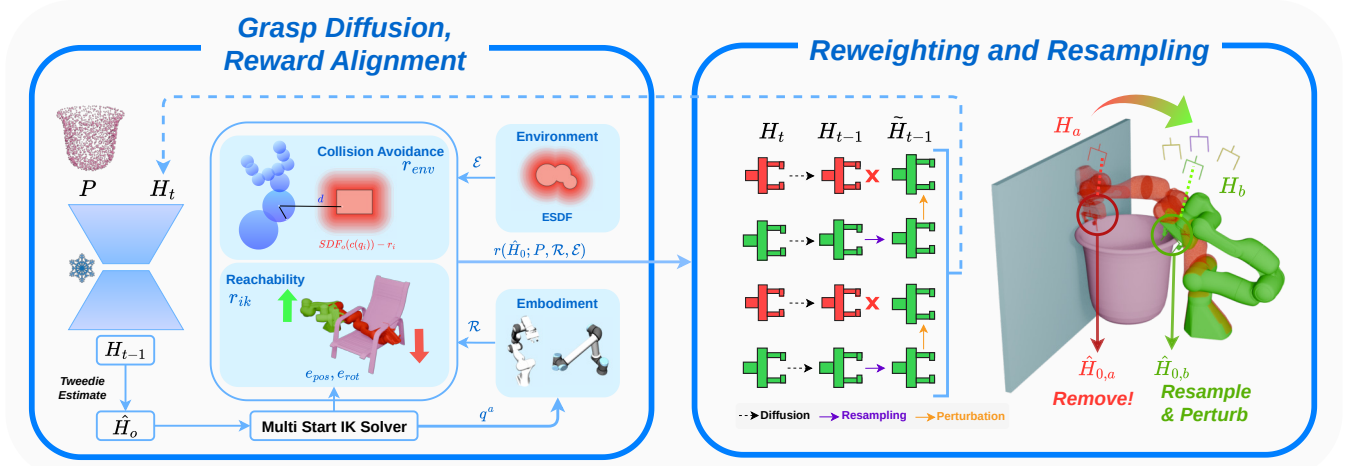


Fig. 2: **Overview of the proposed method:** Given an full object point cloud P and a pretrained Cartesian space grasp diffusion model, the reverse diffusion produces grasp hypotheses H_{t-1} which are mapped to their clean grasp estimates $\hat{H}_0(H_t)$ using the Tweedie estimate, which is evaluated against deployment specific feasibility rewards. The resulting reward $\mathbf{r}(\hat{H}_0; P, \mathcal{R}, \mathcal{E})$ combine embodiment and environment information through multi-start IK, reachability and signed geometric clearances. These rewards instantiate the Feynman-Kac twisting potential used to re-weight and resample (H_b) the particle population in turn removing particles (H_a). Thus embodiment and environment constraints act through population level reweighting and resampling rather than modifying individual denoising transitions, enabling mode level steering toward feasible regions while retaining the multi modal structure of the pretrained grasp prior.

quantities are introduced only at inference time to evaluate and steer grasps toward executable solutions. We assume access to the robot’s forward and inverse kinematics, allowing a Cartesian wrist pose $H^a \in SE(3)$ to be mapped to a feasible joint configuration $q^a \in \mathcal{Q}^a$ for each arm $a \in \{L, R\}$, where $\mathcal{Q}^a$ is the joint limit constrained configuration space of arm a .

Following the reward-tilted diffusion sampling through Twisted Sequential Monte Carlo [22], we formulate the deployment time, embodiment aware and physically feasible grasp generation as sampling from a feasibility tilted version of the pretrained grasp prior. We seek to align the prior p^θ at inference time using a deployment reward $\mathbf{r}(H_0; P, \mathcal{R}, \mathcal{E})$, which evaluates the feasibility of a grasp for the target embodiment and the environment and seek a reward-aligned distribution by regularizing the deployment reward against the pretrained grasp prior,

$$p^* = \arg \max_p [\mathbb{E}_p[\mathbf{r}(H_0; P, \mathcal{R}, \mathcal{E})] - \tau D_{\text{KL}}(p \| p^\theta)] \quad (2)$$

where $\tau > 0$ controls the strength of the reward alignment relative to adherence to the prior. The optimal solution $p^* = p_{0:T}^*(H_{0:T} | P, \mathcal{R}, \mathcal{E})$ is an exponential tilt of the pretrained path distribution [16],

$$p^* \propto p^\theta \exp(\mathbf{r}(H_0; P, \mathcal{R}, \mathcal{E})/\tau). \quad (3)$$

The corresponding diffusion-time twisting potential is obtained by conditioning this terminal tilt on the intermediate state H_t . Following [16], [23], the corresponding twisting potential at diffusion timestep t is

$$\psi_t^*(H_t) = \mathbb{E}_{p^\theta(H_0|H_t, P)} \left[\exp\left(\frac{\mathbf{r}(H_0; P, \mathcal{R}, \mathcal{E})}{\tau}\right) \right] \quad (4)$$

Since this conditional expectation is intractable, we approximate the terminal grasp using its Tweedie estimate and realize the resulting reward aligned distribution through

inference time steering. The following sections describe the inference time reward alignment procedure IV-A, the local refinement procedure IV-B and the deployment specific feasibility rewards IV-C.

A. Inference Time Reward Alignment

The objective in Eq. (2) defines the desired reward-aligned distribution, but direct sampling from this distribution is intractable. We therefore approximate its diffusion time twisting potential and realize the resulting distribution through inference time reward alignment. For continuous diffusion, evaluating the conditional expectation in Eq. (4) can be expensive. We approximate the twist using the denoised final grasp estimate obtained from the Tweedie’s formula [20] from Eq. (1a), and (1b), giving a tractable approximation

$$\psi_t(H_t; P, \mathcal{R}, \mathcal{E}) \approx \exp\left(\mathbf{r}(\hat{H}_0(H_t); P, \mathcal{R}, \mathcal{E})/\tau\right) \quad (5)$$

to the reward-aligned intermediate distribution. We next instantiate the reward $\mathbf{r}(\hat{H}_0(H_t); P, \mathcal{R}, \mathcal{E})$ using robot and environment specific feasibility rewards evaluated on the denoised estimate $\hat{H}_0$. The deployment reward combines kinematic reachability with environment, and object collision related geometric terms. These rewards are evaluated on the denoised grasp estimate and used to construct the FK potential for particle reweighting and resampling. The specific reward terms, their scaling, and aggregation for single- and dual-arm grasps are detailed in Sec. IV-C.

Given the deployment reward $\mathbf{r}(\hat{H}_0; P, \mathcal{R}, \mathcal{E})$, we approximate sampling from the corresponding reward-aligned distribution without modifying the pretrained diffusion model. At diffusion timestep t , we maintain a population of N particles, $\mathcal{H}_t = \{H_t^{(n)}\}_{n=1}^N$ representing multiple hypotheses under the pretrained grasp distribution. Each particle is first propagated using the pretrained reverse diffusion step,

$$H_{t-1}^{(n)} \sim p^\theta(H_{t-1} | H_t^{(n)}, P, t) \quad (6)$$

To align this population toward deployment-feasible configurations, we evaluate the twisting potential $\psi_{t-1}(H_{t-1}^{(n)}; P, \mathcal{R}, \mathcal{E})$ for each particle using the Tweedie-based approximation in Eq. (5). We use the ratio of the current and previous steering event twisting potentials to obtain the incremental difference potential

$$\phi_{t-1}^{(n)} = \psi_{t-1}^{(n)} / \psi_{\text{prev}}^{(n)} = \exp\left(\left(\mathbf{r}_{t-1}^{(n)} - \mathbf{r}_{\text{prev}}^{(n)}\right) / \tau\right) \quad (7)$$

where $\mathbf{r}_{\text{prev}}^{(n)}$ is the reward associated with the particle n from the previous steering event (0 for first event). This incremental potential assigns larger weights to particles whose deployment reward improves between successive steering events. The propagated particles are assigned normalized importance weights according to

$$w_{t-1}^{(n)} = \phi_{t-1}^{(n)} / \sum_{j=1}^N \phi_{t-1}^{(j)} \quad (8)$$

and the particle population is resampled according to $\{w_{t-1}^{(n)}\}_{n=1}^N$. The reweight and resampling step is applied intermittently at a predefined set of diffusion time steps $\mathcal{T}_s \subseteq \{1, \dots, T\}$. At $t \notin \mathcal{T}_s$, particles follow the unmodified reverse diffusion processes. Resampling is performed with replacement, allowing high reward particles to produce multiple offspring. To mitigate the resulting particle impoverishment caused by resampling, duplicated offspring are perturbed in the tangent space $\mathfrak{se}(3)$ as

$$\tilde{H}_{t-1}^{(n)} = \text{Expmap}\left(\text{Logmap}\left(\bar{H}_{t-1}^{(n)}\right) + \gamma\sigma_t\epsilon\right), \quad (9)$$

where $\bar{H}_{t-1}^{(n)}$ is the resampled particle, $\epsilon \sim \mathcal{N}(0, I_6)$, and σ_t is the current diffusion noise scale, and γ controls the perturbation magnitude. For dual arm grasps, the same operation is applied jointly on $SE(3) \times SE(3)$ using $\text{Logmap}_2, \text{Expmap}_2$, and $\epsilon \sim \mathcal{N}(0, I_{12})$. The intermittent steering schedule $\mathcal{T}_s$ allows particles to evolve under the pretrained reverse diffusion between interventions, retaining proximity to the learned multimodal prior, while reward-based alignment reallocates the particles toward higher feasibility modes and tangent-space perturbations maintain local diversity around duplicated particles. Figure 2 summarizes this inference-time process, showing how the pretrained reverse diffusion trajectory is intermittently coupled to deployment specific reward evaluation, particle reweighting, resampling and perturbation.

B. Local Refinement

Toward the end of the diffusion trajectory, $\forall t \in \mathcal{T}_{LR}$, our reward alignment method is complemented by a gradient-free local refinement of near feasible grasps ($\mathcal{T}_{LR} = \{8, 9\}$ for single arm and $\{223, 224, 225\}$ for dual arm). For each particle, a local neighborhood is constructed by applying translational and rotational perturbations along the $\mathfrak{se}(3)$ tangent space. Candidate joint configurations are first obtained through a damped Jacobian update for efficient screening through local rewards, after which the best candidate is

subjected to hard feasibility checks as the global reward. The candidate Jacobian update Δq^a is given by

$$\Delta q^a = J^\top (JJ^\top + \eta^2 I_6)^{-1} \begin{bmatrix} \mathbf{t}_c - \mathbf{t}(q^a) \\ \text{Logmap}(\mathbf{R}_c \mathbf{R}(q^a)^\top) \end{bmatrix} \quad (10)$$

where $H_c = (\mathbf{R}_c, \mathbf{t}_c)$ and $H(q^a) = (\mathbf{R}(q^a), \mathbf{t}(q^a))$ denote the perturbed and parent grasp poses respectively, J is the end-effector spatial Jacobian evaluated at q^a , and $\eta > 0$ is the damping coefficient. The local objective focuses on fine grained gripper-object interactions while whole arm constraints are enforced by rejecting candidates that violate or degrade the parent’s feasibility constraints. The local and global steering objectives are instantiated from the same deployment-specific feasibility measures, defined next.

C. Grasp Feasibility Rewards

We evaluate the Tweedie estimate of each denoised grasp using geometric and kinematic quantities induced by the embodiment $\mathcal{R}$ and the deployment environment $\mathcal{E}$. The evaluators are queries directly at the inference time and need not be differentiable with respect to the grasp pose, unlike gradient-based approaches [9], [11]. For each arm a , the candidate grasp pose is first resolved to a joint configuration q^a through multi-start inverse kinematics using `PyTorch Kinematics` [24].

a) Arm Collision Avoidance: We represent the robot links using collision spheres for efficient geometric collision evaluation. For an arm a , arm-environment clearance is evaluated against the environment obstacles using their signed distance functions,

$$d_{\text{env}}^a = \min_{i,o} [\text{SDF}_o(c_i^a(q^a)) - \rho_i^a] \quad (11)$$

where $c_i^a(q^a)$ and ρ_i^a denote the center and radius of the i -th collision sphere, and $o \in \mathcal{E}$ indexes the environment obstacles. In contrast, arm-object clearance is evaluated directly from the point cloud P using sphere-point signed distances,

$$d_{\text{obj},ij}^a = \|c_i^a(q^a) - p_j\|_2 - \rho_i^a, \quad p_j \in P \quad (12a)$$

$$d_{\text{obj}}^a = \text{BottomKMean}_k\left(\{d_{\text{obj},ij}^a\}_{i,j}\right) \quad (12b)$$

where BottomKMean_k denotes the mean distances between the k smallest pairwise sphere point distances, excluding the gripper collision spheres. This clearance is then used for the gripper-object term in the local refinement objective to rank candidate grasp perturbations.

Self collision d_{self}^a is evaluated from the minimum pairwise signed clearance among spheres of the same arm, whereas for dual-arm grasps the inter-arm clearance d_{inter} is evaluated between the two collision sphere sets in the shared world frame. The pairwise sphere clearance is

$$d_{\text{inter},ij} = \|c_i^L(q^L) - c_j^R(q^R)\|_2 - \rho_i^L - \rho_j^R \quad (13a)$$

$$d_{\text{inter}} = \min_{i,j} \{d_{\text{inter},ij}\} \quad (13b)$$

To deploy with dense geometry (Real-World in Sec. V-B), we instantiate the environment distance queries via the Euclidean Signed Distance Fields (ESDFs) using `cuRoboV2` [25].

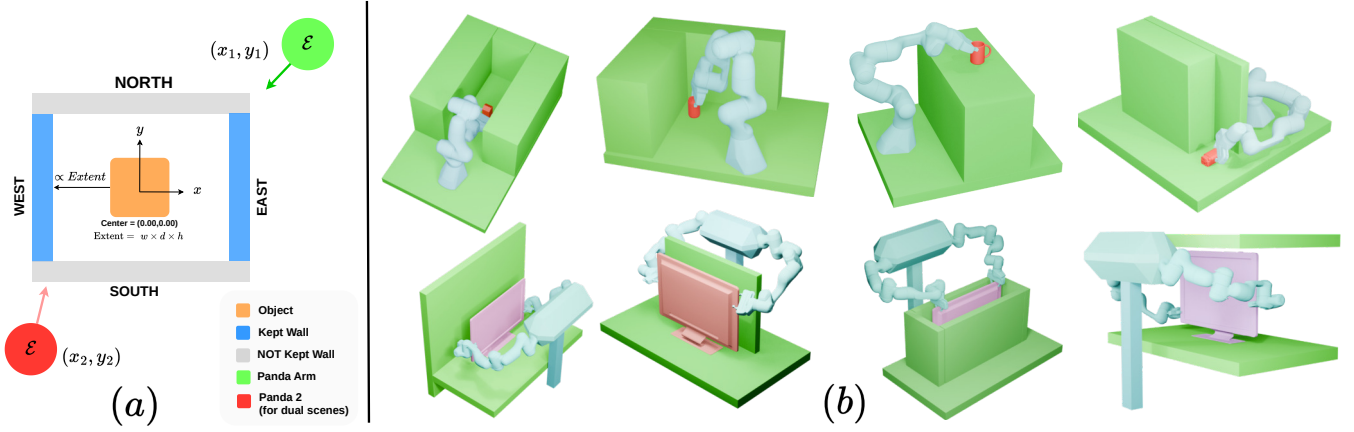


Fig. 3: **Evaluation settings.** (a) **Large Scale Evaluation.** Top-down schematic of the scene generation scheme (see legend); the local frame at the object’s center shows its position and extent ($w \times d \times h$), while θ and (x, y) at the sampled Panda arm(s) base give its heading and location relative to the world origin. The wall labels (NORTH / SOUTH / EAST / WEST) indicate the four candidate wall positions, of which kept vs. not-kept walls are distinguished per scene. (b) **Curated Scenes.** Top-Row: **Single Arm Scenes** from L-R: S1. *Corridor*, S2. *Room Corner*, S3. *Shelf Top* and S4. *Partition Board*. Bottom-Row: **Dual Arm Scenes** from L-R: S1. *Table* S2. *Wall* S3. *Box* S4. *Shelf*

b) **Grasp Reachability:** For each arm a , the candidate grasp is mapped to a joint configuration q^a through multi-start IK, selecting the solution with minimum weighted pose residual $e_{IK}^a = w_t e_{pos}^a + w_R e_{rot}^a$, where $e_{pos}^a = \|\mathbf{t}^a - \mathbf{t}_{FK}^a(q^a)\|_2$ and $e_{rot}^a = \|\text{Logmap}(\mathbf{R}^a \mathbf{R}_{FK}^a(q^a)^T)\|_2$ measure translational and rotational errors between the candidate grasp pose H^a and its forward-kinematic realization $H_{FK}^a(q^a)$, respectively. This residual provides the kinematic reachability measure, with smaller values indicating closer realization of the desired grasp pose.

c) **Reward scaling and aggregation:** The resulting signed clearances and kinematic residuals are converted to bounded rewards using a $\tanh(\cdot)$ function $\mathcal{S}(d; s) = \tanh(d/s)$, where s is a fixed characteristic scale. Thus all clearance measures $d \in \{d_{env}^a, d_{obj}^a, d_{self}^a, d_{inter}^a\}$ are mapped as $\mathbf{r} = \mathcal{S}(d; s)$, while the non-negative IK residual is negated before normalization as $\mathbf{r}_{IK}^a = \tanh(-e_{IK}^a/s_{IK})$

For a single-arm grasp, the deployment reward is given by

$$\mathbf{r}_{single} = w_{IK} \mathbf{r}_{IK} + w_{env} \mathbf{r}_{env} + w_{obj} \mathbf{r}_{obj} + w_{self} \mathbf{r}_{self} \quad (14)$$

For a dual-arm grasp, the two predicted grasp poses are first assigned to the physical arms by selecting the assignment with the lowest total IK residual. Assuming $\mathbf{r}_{single}^a$ for the reward in Eq. 14 evaluated on each arm $a \in \{L, R\}$, the dual-arm reward is

$$\mathbf{r}_{dual} = w_{inter} \mathbf{r}_{inter} + \min_{a \in \{L, R\}} \mathbf{r}_{single}^a \quad (15)$$

where $\mathbf{r}_{inter}$ penalizes collisions between the two arms. Taking the minimum over arms ensures that a grasp is rewarded only if it is feasible for both.

V. EXPERIMENTS

A. Baselines

We compare our method with the following baselines:

a) **Plain Sampling:** Generate N grasps per object using the base diffusion sampler and then evaluate them all.

b) **Rejection Filter:** Generate N grasps per object using the base diffusion sampler and filter out grasps that don’t have strictly positive scalar reward (Sec. IV-C). This can generate less than N grasps per object on average.

c) **Oversample ($M \rightarrow N$):** Generate M grasps where $M > N$, and retain only top N by reward. (M, N) is $(500, 100)$ for dual-arm and $(150, 50)$ for single-arm.

d) **Gradient Guidance:** Additionally, we compare with an existing gradient guidance method [9]. We extend the single arm implementation to the dual-arm case by introducing an additional inter-arm collision cost term f_{inter} computed using the differentiable log-sum-exp smoothing from Appendix A.2 of [10]. We additionally augment per-arm objective with an arm–object clearance soft constraint, using the sphere–point signed distance in Eq. 12a and the same smooth SoftPlus collision cost as in [9]. For each Tweedie estimated grasp pose $(\hat{H}_0^L, \hat{H}_0^R)$ and feasible IK solution set $\mathcal{Q}^L$ and $\mathcal{Q}^R$, we evaluate the original single-arm cost f^L and f^R . We jointly select the optimal pair $(q^{L,*}, q^{R,*})$, where $q^{L,*} \in \mathcal{Q}^L$ and $q^{R,*} \in \mathcal{Q}^R$, by solving

$$(q^{L,*}, q^{R,*}) = \arg \min_{q^L, q^R} \left[f^L(\hat{H}_0^L, q^L) + f^R(\hat{H}_0^R, q^R) + \lambda_{inter} f_{inter}(q^L, q^R) \right] \quad (16)$$

where λ_{inter} is the Lagrange multiplier for the inter-arm collision constraint, while the per-arm multipliers are contained in f^L and f^R . The task space guidance gradients are then evaluated at the current grasp poses’ Tweedie estimate $(\hat{H}_0^L, \hat{H}_0^R)$ and the jointly selected IK solutions $(q^{L,*}, q^{R,*})$,

$$\begin{aligned} \nabla_{\hat{H}_0^L} F &= \nabla_{\hat{H}_0^L} f^L + \lambda_{inter} (J_L^+)^T \nabla_{q^{L,*}} f_{inter}, \\ \nabla_{\hat{H}_0^R} F &= \nabla_{\hat{H}_0^R} f^R + \lambda_{inter} (J_R^+)^T \nabla_{q^{R,*}} f_{inter}, \end{aligned} \quad (17)$$

where J_L^+ and J_R^+ denote the damped pseudoinverses of the left and right arm Jacobians, respectively. These gradients are concatenated to form a joint 12D guidance update for the given grasp pair.

	Method	Reach.↑	Coll.↓	CS ↑		SR _{hold} ↑	SR _{hold} +CS ↑		Time↓	VRAM↓
		%	%	@10	@all	%	@10	@all	s (+base)	GB (+base)
Single-Arm	Plain sampling	72.51	76.24	40.90	15.60	18.44	24.00	8.93	0.34 (base)	0.53 (base)
	Rejection filter†	89.50	71.41	47.14	18.81	22.31	27.02	10.78	+1.12	+0.00
	Oversampling	97.84	56.88	60.34	41.06	29.77	34.84	22.80	+1.52	+0.08
	Gradient guidance	76.41	31.52	69.64	57.71	21.51	32.01	18.52	+11.23	+0.01
	Ours, O	98.26	28.41	72.99	70.42	38.84	37.84	38.13	+3.32	+0.00
	Ours, O+LR	98.43	26.56	74.54	72.41	41.41	40.31	40.79	+3.72	+0.31
Dual-Arm	Plain sampling	40.91	88.85	19.55	4.88	67.52	17.17	4.21	10.46 (base)	1.28 (base)
	Rejection filter†	76.22	86.30	20.11	9.26	78.05	17.48	8.23	−0.16	+0.00
	Oversampling	78.70	82.99	33.92	14.55	79.91	30.30	12.74	+14.67	+1.86
	Gradient Guidance	61.74	75.51	49.07	15.97	67.55	43.43	12.48	+175.36	+1.56
	Ours, O	98.57	49.23	62.18	50.77	85.11	54.98	45.03	+2.43	+0.00
	Ours, O+LR	98.80	46.47	61.93	53.53	84.64	54.28	47.59	+6.47	+0.09

TABLE I: **Large-scale scene evaluation.** We generate N grasps for all methods; metrics are averaged across all scenes. **Reach.** is the reachability rate (IK convergence; for dual-arm, both arms must converge). **Coll.** refers to the collision rate. **CS@ k** is constraint satisfaction among the top- k ranked grasps per scene. **SR_{hold}** denotes stability under all pull axes. **SR_{hold}+CS@ k** is the joint fraction that is stable, reachable, and collision-free. **Time** and **VRAM** are end-to-end generation latency (s) and peak GPU memory (GB), reported as the change relative to plain sampling of the same setting; the plain-sampling row gives the absolute baseline. Rows below the rule are ours. Green / Orange indicate best/second-best per column within each setting. † Rejection filter on average generates fewer than N grasps per scene. Our method is denoted by shorthand O and with local refinement added to it, shorthand is O+LR

B. Evaluation settings

Large-scale evaluation. Following the scheme in Fig. 3(a), we randomly spawn up to four walls around an object and randomize the base position of each Franka Emika Panda arm, retaining 50 scenes per object after discarding those with no executable grasp. Using 18 single-arm objects from [26] and 12 dual-arm objects from [5], with $N=50$ grasps and $N=100$ grasp pairs per scene respectively, yielding 45,000 single-arm grasps and 60,000 dual-arm grasp pairs, all evaluated in MuJoCo.

Curated scenes. For both single and dual arm settings we use four representative scenes, each with a single object in a distinct configuration (Fig. 3(b)). We evaluate all methods on two embodiments, a UR5e and a Franka Emika Panda, both with Robotiq grippers, using the execution metrics of Sec. V-C and $N=50$ grasps or grasp pairs per scene.

Real world. To demonstrate sim-to-real transfer, we evaluate on xArm7 manipulators with xArm grippers across three scenes per object, using two objects for each of the single and dual arm settings (Fig. 5). The *Easy* scene tests reachability and ground collision avoidance, with additional inter-arm clearance in the dual arm setting. The *Medium* scene constrains the workspace with a corridor-like configuration, and the *Hard* scene combines narrow clearances with restricted approach directions. Point clouds are captured from two RealSense views Fig. 5 and fused to compute the ESDF. Grasps are executed using the cuRoboV2 motion planner.

C. Metrics

Unless otherwise specified, all metrics reported are calculated for N total grasp candidates generated.

Reachability (Reach). A grasp is considered *reachable* if an IK solution exists with a positional and rotational error less than 1 cm and 2 degrees respectively (Sec. IV-C (b)). For dual-arm grasp, both arms must admit valid IK solutions.

Collision (Coll). Each grasp is instantiated via its corresponding joint configuration produced by the method and the arm is checked for collisions with the environment, with itself, or in the dual-arm case, with each other.

Constraint Satisfaction (CS@ k). A grasp is constraint-satisfying if it is reachable and collision-free. CS@{10, all} reports the fraction of the top- k grasps that satisfy this, where grasps are ranked by the harmonic mean of the normalized values of the model specific grasp quality scores (force closure scores for DAGDiff, and discriminator confidence for GraspGen). For plain sampling, rejection filtering, oversampling, and our method, the feasibility score is Eq. 14 (single arm) or Eq. 15 (dual arm); for Gradient Guidance, we use the objective of [9] (or Eq. 16 for our dual-arm extension), min-max normalized over all sampled grasps in the scene.

Holding Success Rate (SR_{hold}). Each grasp is instantiated in MuJoCo with its arm joint configuration, and the object but without environment obstacles. Following [11], a 3 N pull is applied sequentially along the $\pm x$, $\pm y$, and $\pm z$ axes of the object frame, and the grasp succeeds if the object’s center of mass stays within 5 cm of its initial position.

SR_{hold}+CS@ k . Under the same harmonic ranking, the fraction of top- k grasps which satisfy the constraints and pass SR_{hold}. This is what we define as *feasibility*.

Task Success Rate (SR_{task}). The fraction of grasps that a motion planner in MuJoCo can execute through the full task trajectory (e.g., lifting, pulling the object out) in the presence of environment obstacles while maintaining a stable grasp.

D. Results

1) Quantitative Evaluation: We report large scale results in Tab. I and curated scene results in Tab. II. Plain sampling represents the unguided prior and, unless mentioned otherwise, serves as the reference for all comparisons, since the complete set of executable grasps in a scene is unknown.

Arm Method		Single-Arm (SR_{task})				Dual-Arm (SR_{task})			
		S1	S2	S3	S4	S1	S2	S3	S4
Franka	Plain sampling	10/50	3/50	4/50	22/50	7/50	7/50	0/50	3/50
	Rejection filter	10/29	3/34	4/27	22/37	7/26	7/16	0/14	3/9
	Oversampling	48/50	13/50	29/50	49/50	39/50	45/50	1/50	9/50
	Gradient Guidance	24/50	14/50	37/50	34/50	30/50	39/50	18/50	29/50
	Ours	50/50	33/50	46/50	50/50	48/50	48/50	35/50	50/50
	Ours + LR	50/50	36/50	48/50	50/50	50/50	49/50	35/50	50/50
UR5e	Plain sampling	11/50	6/50	1/50	21/50	9/50	11/50	3/50	0/50
	Rejection filter	11/34	6/29	1/25	21/45	7/26	5/9	3/22	1/7
	Oversampling	47/50	23/50	10/50	50/50	37/50	35/50	34/50	6/50
	Gradient Guidance	39/50	13/50	31/50	27/50	48/50	42/50	37/50	33/50
	Ours	46/50	35/50	44/50	50/50	49/50	48/50	50/50	50/50
	Ours + LR	47/50	37/50	46/50	50/50	50/50	50/50	50/50	50/50

TABLE II: **Curated-scene evaluation.** SR_{task} (successful/total grasps) on curated scenes (refer Fig. 3) for single- and dual-arm grasps across arm types. SR_{task} is the fraction of generated grasps that are constraint-satisfying and successfully transport the object in MuJoCo. **Green** / **Orange** indicate best/second-best per column within each arm.

Sample Efficiency: Rejection filter produces only 35.77 single- and 37.3 dual-arm grasps on average in large scale evaluation, reflecting less sample efficiency than other methods that retain all single and dual grasps respectively. Unlike oversampling, which improves feasibility by discarding grasps based on reward, our method reallocates particle mass toward feasible modes, increasing yield of feasible grasps for a fixed number of samples.

Constraint Satisfaction: Only 15.6% of single-arm grasps and 4.9% of dual-arm grasps from the prior are reachable and collision-free. Our method (**O**) raises constraint satisfaction over the full population ($CS@all$) by +55pp and +46pp, with near-complete reachability in both settings and a collision rate reduced by 48 pp and 40 pp. Meanwhile gradient guidance trails our method (**O**) by 13 pp in the single-arm case, but on moving to dual-arm it lags by 35pp, since resampling reallocates the population to **feasible modes** regardless of their distance in $SE(3) \times SE(3)$. These modes exist in the prior but carry little mass, so drawing more samples, as in oversampling, recovers them only in proportion to that mass, adding +26 pp and +10 pp, whereas our method (**O**) concentrates the population on them at a fixed budget.

Stability: Steering toward feasibility does not cost grasp stability. SR_{hold} for our method (**O**) over all generated grasps improves by +20 pp (single-arm) and +18 pp (dual-arm) over plain sampling, and by a similar margin over gradient guidance. Gradient guidance adds a reward gradient to every reverse step, while our method (**O**) leaves the transition kernel untouched, and every surviving trajectory remains close to a sample path of *the learned prior*. Our method (**O**) improves feasibility, ($SR_{hold} + CS$) over all grasps, by +29 pp and +41 pp over plain sampling and by +20 pp and +32 pp over gradient guidance. The feasible grasps of the baselines are confined to a few top-ranked samples while ours span the population, emphasizing our methods ability to provide population-wide correction.

Effect of Local Refinement: Local refinement acts only on near-feasible particles, so it leaves the top-ranked grasps untouched and instead attempts to make marginal ones feasible. The result is a consistent gain in collision-free grasps and

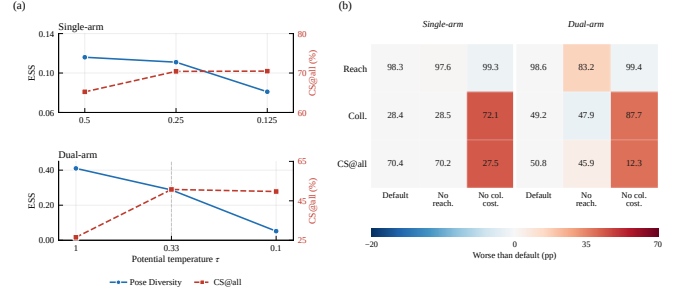


Fig. 4: **Ablations:** a) Effect of Temperature τ on Effective Sample Size and Constraint Satisfaction. b) Effect of rewards on Reachability and Collision population-wide feasibility of 2–3 pp in both settings, with stability and ranking unchanged or slightly improved. Since it accounts for the entire memory overhead (0.3 GB) and most of the dual-arm latency (4 of 6.5 s), we report (**O+LR**) as an option rather than as part of the core steering.

Downstream execution and cross-embodiment transfer: Tab. II evaluates SR_{task} , i.e. whether generated grasps can actually be executed by a motion planner in MuJoCo in the presence of the obstacles, on the curated scenes of Fig. 3. Across the 16 scene embodiment combinations, our method (**O**) is best or tied-best in 7 scenes and, with local refinement (**O+LR**), in all 16. The gains are largest where the prior fails (Ex. In the dual-arm *Box* scene (S3) on the Franka, plain sampling yields no executable grasp pair and $5 \times$ oversampling a single pair), and gradient guidance recovers about a third, and our method (**O**) more than two thirds (35/50), showing that the improved static feasibility translates to reliable downstream manipulation execution.

Real World: After three trials per scene, our method (**O+LR**) achieves **30/36** successful executions (Table III). Failures mainly occur for the bucket in the *Hard* setting due to asymmetric loading and the lack of compliant torque control.

Computational cost: Steering adds 3.3 s to single-arm and 2.4 s to dual-arm generation, against 11.2 s and 175.4 s for gradient guidance, which evaluates IK and its gradient at every diffusion step, in contrast to our intermittent steering (**O**). The cost of steering is roughly constant, while the cost of oversampling grows with the prior and the sample factor it requires. For the cheap single-arm prior, $3 \times$ oversampling is about half our overhead, but for dual-arm grasping where $5 \times$ oversampling is needed, our method is $6 \times$ faster and adds no GPU memory against nearly 2 GB overhead. Using local (**O+LR**) adds a mere 0.4 s and 4 s overhead in single and dual settings to our method respectively, while improving collision avoidance and reachability, and hence the overall constraint satisfaction over our method (**O**).

2) **Ablation Study:** All ablations use our method (**O**) to isolate the hyperparameters. We first vary the temperature τ (Fig. 4a). Lowering τ raises constraint satisfaction until it saturates, at $\tau \approx 0.25$ for single-arm and $\tau \approx 0.33$ for dual-arm, while diversity keeps falling as the population collapses onto a few high-reward modes. Fig. 4b removes individual reward terms. Without the collision reward the collision rate returns to nearly the level of plain sampling in both settings, so reachability alone steers particles toward the robot but

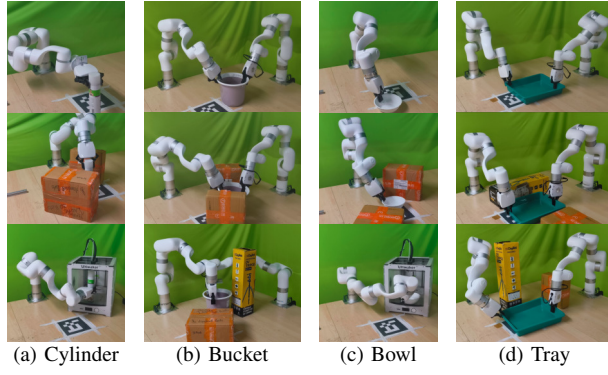


Fig. 5: **Real-world grasping.** Columns correspond to the four objects; rows from top to bottom correspond to *Easy*, *Medium*, and *Hard* scenarios.

Arm Type	Object	Easy	Medium	Hard	Total
Single	Bowl	3/3	3/3	3/3	9/9
	Cylinder	3/3	3/3	2/3	8/9
Dual	Bucket	3/3	2/3	1/3	6/9
	Tray	3/3	2/3	2/3	7/9

TABLE III: **Real-world success.** 9 trials per object across varying difficulty, not around obstacles. Removing the reachability reward has a much smaller effect, since the collision reward already disfavors unreachable poses which tend to place the arm in collision. This coupling is strongest for single-arm grasping, where objects are small and an unreachable pose usually forces the arm into collision. For the larger dual-arm objects the coupling is weaker, and reachability drops by 15 pp, highlighting the necessity of the rewards. More results are provided in the video.

VI. CONCLUSION

We presented **Steer2Grasp**, a training-free framework that introduces embodiment awareness into frozen grasp diffusion models through population-level inference-time steering. By reweighting and resampling particles using kinematic and collision rewards, the method enables population reallocation to feasible modes while preserving the closeness to the underlying grasp prior. Experiments across single- and dual-arm models, multiple embodiments, constrained scenes, and hardware show improved feasible and stable grasp generation over plain sampling and local gradient guidance at substantially lower cost than aggressive oversampling. As future work, we plan to extend the framework to incorporate sparse or long-horizon objectives, such as manipulability, future trajectory feasibility, and contact dynamics, through trajectory-level or dynamics-aware guidance.

REFERENCES

- [1] H. Zhang, J. Tang, S. Sun, and X. Lan, “Robotic grasping from classical to modern: A survey,” *arXiv preprint arXiv:2202.03631*, 2022.
- [2] J. Urain, N. Funk, J. Peters, and G. Chaltatzaki, “Se(3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5923–5930.
- [3] A. Murali, B. Sundaralingam, Y.-W. Chao, W. Yuan, J. Yamada, M. Carlson, F. Ramos, S. Birchfield, D. Fox, and C. Eppner, “Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training,” 2025. [Online]. Available: <https://arxiv.org/abs/2507.13097>

- [4] G. Zhai, Y. Zheng, Z. Xu, X. Kong, Y. Liu, B. Busam, Y. Ren, N. Navab, and Z. Zhang, “Da² dataset: Toward dexterity-aware dual-arm grasping,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, p. 8941–8948, 2022.
- [5] M. F. Karim, M. S. Hashmi, S. Bollimuntha, M. R. Tapeti, G. Singh, N. Govindan, and K. M. Krishna, “Dg16m: A large-scale dataset for dual-arm grasping with force-optimized grasps,” *arXiv preprint arXiv:2503.08358*, 2025.
- [6] M. F. Karim, V. Vembar, K. Patra, G. Singh, and K. M. Krishna, “Dagdiff: Guiding dual-arm grasp diffusion to stable and collision-free grasps,” *arXiv preprint arXiv:2509.21145*, 2025.
- [7] K. Kim, S. Back, G. Lee, S. Lee, S. Noh, and K. Lee, “Bi-grasformer: End-to-end bimanual grasp transformer,” *arXiv preprint arXiv:2509.19142*, 2025.
- [8] X. Lou, Y. Yang, and C. Choi, “Learning to generate 6-dof grasp poses with reachability awareness,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1532–1538.
- [9] Y. Jia, Y. Jiang, K. Lv, Y. Ren, and X. Li, “Arm-aware guided dexterous grasp generation with arm-agnostic grasp models,” *IEEE Robotics and Automation Letters*, vol. 11, no. 5, pp. 5875–5882, 2026.
- [10] S. Wang, K. Lv, M. Yu, and X. Li, “Embodisteer: Steering embodiment-agnostic visuomotor policies with joint-space guidance for zero-shot cross-embodiment deployment,” *arXiv preprint arXiv:2606.12965*, 2026.
- [11] S. Bodmer, R. Zurbrugg, T. Portela, H. Ma, A. Didier, M. Hutter, C. Jones, and M. Zeilinger, “Grounding generative policies in physics: Optimization-guided diffusion for robot control,” *arXiv preprint arXiv:2606.24208*, 2026.
- [12] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” 2021. [Online]. Available: <https://arxiv.org/abs/2105.05233>
- [13] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” *arXiv preprint arXiv:2207.12598*, 2022.
- [14] 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>
- [15] M. Uehara, X. Su, Y. Zhao, X. Li, A. Regev, S. Ji, S. Levine, and T. Biancalani, “Reward-guided iterative refinement in diffusion models at test-time with applications to protein and dna design,” *arXiv preprint arXiv:2502.14944*, 2025.
- [16] R. Singhal, Z. Horvitz, R. Teehan, M. Ren, Z. Yu, K. McKeown, and R. Ranganath, “A general framework for inference-time scaling and steering of diffusion models,” 2025. [Online]. Available: <https://arxiv.org/abs/2501.06848>
- [17] B. Han, Y.-W. Chao, E. Coumans, C. Eppner, J. Deng, S. Birchfield, and A. Murali, “Graspgen-x: Cross-embodiment 6-dof diffusion-based grasping,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 20 878–20 889.
- [18] R. Römer, A. von Rohr, and A. P. Schoellig, “Diffusion predictive control with constraints,” *arXiv preprint arXiv:2412.09342*, 2024.
- [19] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840–6851.
- [20] B. Efron, “Tweedie’s formula and selection bias,” *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1605–1614, 2011.
- [21] A. Bansal, H.-M. Chu, A. Schwarzschild, R. Sengupta, M. Goldblum, J. Geiping, and T. Goldstein, “Universal guidance for diffusion models,” in *International Conference on Learning Representations*, vol. 2024, 2024, pp. 51 304–51 323.
- [22] L. Wu, B. L. Trippe, C. A. Naesseth, D. M. Blei, and J. P. Cunningham, “Practical and asymptotically exact conditional sampling in diffusion models,” 2024. [Online]. Available: <https://arxiv.org/abs/2306.17775>
- [23] J. Kim, T. Yoon, P. Phunphibarn, S. Kim, M. Mardani, and M. Sung, “Contrastive distribution matching for amortized sequential monte carlo in discrete diffusion,” *arXiv preprint arXiv:2605.23346*, 2026.
- [24] S. Zhong, T. Power, A. Gupta, and P. Mitran, “PyTorch Kinematics,” Feb. 2024.
- [25] B. Sundaralingam, A. Murali, and S. Birchfield, “curobo2: Dynamics-aware motion generation with depth-fused distance fields for high-dof robots,” *arXiv preprint arXiv:2603.05493*, 2026.
- [26] C. Eppner, A. Mousavian, and D. Fox, “Acronym: A large-scale grasp dataset based on simulation,” 2020. [Online]. Available: <https://arxiv.org/abs/2011.09584>