Proceedings of Machine Learning Research vol 283:1–15, 2025 7th Annual Conference on Learning for Dynamics and Control

# STRiDE: State-space Riemannian Diffusion for Equivariant Planning

**Nishanth Rao*** NISHRAO@SEAS.UPENN.EDU
*University of Pennsylvania, GRASP Lab*

**Evangelos Chatzipantazis*** VAGHAT@SEAS.UPENN.EDU
*University of Pennsylvania, GRASP Lab*

**Kostas Daniilidis** KOSTAS@CIS.UPENN.EDU
*University of Pennsylvania, GRASP Lab; Archimedes, Athena RC*

**Editors:** N. Ozay, L. Balzano, D. Panagou, A. Abate

## Abstract

Fast and reliable motion planning is essential for robots with many degrees of freedom in complex, dynamic environments. Diffusion models offer a promising alternative to classical planners by learning informative trajectory priors. In current imitation-learning paradigms, these models are kept lightweight—lacking encoders—and trained to overfit to a single environment. As a result, adaptation relies solely on diffusion guidance, which fails under large execution-time changes or varying initializations. In addition, current approaches ignore the underlying topology of the state space thus requiring heavy guidance that dominates planning time and reduces efficiency dramatically. We introduce **STRiDE**, a novel diffusion motion planner that operates directly on the **state space manifold** and learns **equivariant trajectory priors**. Our approach eliminates the need for retraining under rotations around the gravity axis and enables faster convergence using Riemannian (rather than ambient) guidance. STRiDE delivers efficient, robust, and generalizable planning, overcoming key limitations of existing approaches. Supplementary Material.

**Keywords:** Equivariance, Diffusion Models, Motion Planning, State Space Manifold, Riemannian Guidance, Robot Manipulation

## 1. Introduction

Motion planning is a crucial component of autonomous systems. The goal is to find smooth, feasible trajectories between given states while avoiding obstacles and respecting kinematic constraints. The problem is notoriously challenging for robots with many degrees of freedom in environments with intricate geometries and dynamic obstacles. Classical methods like sampling-based (Kavraki et al., 1996; Lavalle, 1998; Kuffner and LaValle, 2000; Gammell et al., 2014) and optimization-based approaches (Ratliff et al., 2009; Toussaint, 2009; Kalakrishnan et al., 2011) face issues such as computational intensity, non-smooth trajectories, and reliance on good initialization.

To overcome these limitations deep learning priors learned from previously successful plans have been proposed (Ichter et al., 2018; Wang et al., 2020; Bency et al., 2019) guiding optimization towards more promising regions and reducing planning time. Diffusion and score-based models (Sohl-Dickstein et al., 2015; Song and Ermon, 2019) have shown promise in accelerating motion planning (Janner et al., 2022b; Carvalho et al., 2023) by integrating efficient sampling from the diffusion prior with motion optimization costs through guidance (Dhariwal and Nichol, 2021).

* Equal Contribution

© 2025 N. Rao, E. Chatzipantazis & K. Daniilidis.

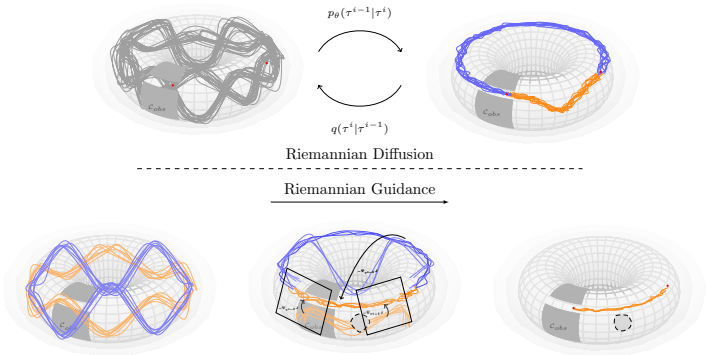


Figure 1: **Riemannian Trajectory Diffusion Model**. $(\text{Top}; \text{Left} \leftarrow \text{Right})$ Multi-scale Riemannian Diffusion is applied at the state-space trajectory-level during training. $(\text{Top}; \text{Left} \rightarrow \text{Right})$ During Riemannian Denoising $p_\theta(\tau^{i-1}|\tau^i)$ random trajectories are sampled on the state-space manifold, anchored to start and goal states and provided to the network that iteratively predicts the noise on the state space until it creates feasible trajectories (e.g. avoid C-space obstacle in the figure). $(\text{Bot}; \text{Left} \rightarrow \text{Right})$ Denoising is interleaved with on-manifold guidance via Riemannian gradient descent ($\nabla_{\tau^{i-1}} J$) (e.g. avoiding novel object in the figure) resulting in sampled trajectories that are both kinematically-feasible (high prior) and cost-minimizing (high-likelihood). The blue and orange distributions show alternative paths between start and goal states for a continuous joint on which the diffusion model is trained (see Sec. 3.3). For a revolute joint, the branch that goes through joint limits will be discarded $(\text{Bot}; \text{Right})$ .

Without encoders, diffusion-guidance is the only means to adapt the distributions to local changes of the environment but it cannot handle global changes that happen during execution or at initialization and full retraining is required. Moreover, current approaches largely ignore the topology of the state space which leads to ineffective training and heavier guidance that dominates the computation cost and reduces planning efficiency. To overcome these challenges, we propose STRiDE, a novel diffusion-based motion planning algorithm that *operates directly on the state space manifold* and *produces equivariant trajectory distributions*. Our contributions are two-fold:

**1. State Space Diffusion Planning and Optimization:** Our model accounts for the complex topology of the state space of a kinematic chain during all of the stages *sampling, denoising and guidance* by operating on the embedded *hypertorus* instead of the *euclidean space*. This leads to stable training and faster inference with less guidance steps which was a bottleneck of previous methods. Operating at the trajectory-level, we have to account for continuity between the states which motivates our representation in an embedding space instead of the quotient representation. We perform on-manifold motion optimization via Riemannian gradient descent, which has not been explored for diffusion models on non-flat spaces to our knowledge. The steps are visualized in Fig.1.

**2. Equivariance via Positive-Negative Embedding:** As in previous approaches we keep the diffusion model lightweight and avoid large encoders of the environment. We propose to achieve generalization via planning on a *learned canonicalized environment* that remains invariant under rotations around the base. To account for symmetry-breaking effects due to joint limits we create an equivariant pair of trajectories, embed both on the same diffusion prior and discard the infeasible branch during inference after decanonicalization. We perform experiments in cluttered environments and pick-and-place tasks to demonstrate the efficiency of our planner over previous approaches.

## 2. Related Work

**Diffusion Models for Planning.** In Janner et al. (2022a), diffusion models were combined with motion optimization via guidance for long-horizon trajectory generation. The idea is to predict all

timesteps simultaneously by iteratively refining sampled trajectories. MPD (Carvalho et al., 2023) built on this idea introducing guidance costs for manipulation. Recent surveys categorize current methods in motion planning (Ubukata et al., 2024) and beyond (Urain et al., 2024). MPD has been used for long range composition tasks conditioned on visual and language input (Liang et al., 2024) as well as in hierarchical control (Chen et al., 2024).

**Diffusion models on Manifolds.** Generalizating diffusion or score-based models on riemannian manifolds (Bortoli et al., 2022; Huang et al., 2022; Lou et al., 2023) has spiked interest recently. Jing et al. (2022) design a diffusion model in the intrinsic representation of the torus for conformal molecule generation. Our diffusion model, on the other hand, operates on the embedded hypertorus to account for trajectory continuity. (Leach et al., 2022) performs $SO(3)$ denoising for rotational alignment. Manifold preserving guidance for diffusion models is only discussed for linear subspaces of the data distribution (He et al., 2024; Chang et al., 2023). We perform guidance on the state space manifold via a Riemannian Gradient descent on the hypertorus.

**Equivariance.** Geometric deep learning (Bronstein et al., 2021) provides strong structural inductive biases to deep neural networks via the design of constrained layers (Cohen and Welling, 2016; Worrall et al., 2016; Weiler et al., 2018; Thomas et al., 2018), or learned canonicalization (Kaba et al., 2022). It has been successfully applied to a number of applications in robotics ranging from Depth estimation (Kumar et al., 2022), 3d Reconstruction (Chatzipantazis et al., 2023), Pose estimation (Howell et al., 2023), 3d Registration (Pertigkiozoglou et al., 2024a), Inertial odometry (Jayanth et al., 2024), etc. In policy learning (Yang et al., 2023; Wang et al., 2024a), equivariance provides a solution to the problem of learning from few demonstrations. The learned policies are in the gripper space and defer motion planning to a classical algorithm which needs to solve the hard problem of inverse kinematics while avoiding obstacles (See Experiments). Equivariance has been successful in learning grasp poses for manipulation (Simeonov et al., 2022; Hu et al., 2024; Choi and Figueroa, 2024). Urain et al. (2023) uses equivariant diffusion models on $\mathrm{SE}(3)$ to learn distributions over poses; motion planning is deferred to guidance. Reversely, we learn trajectories in the state space and use the end-effector poses for guidance. EDGI (Brehmer et al., 2023) proposes a general pipeline for problems that admit $\mathrm{SE}(3)$ symmetry for diffusion-based motion planning. Our task is not natively in that framework; we work directly on the state space of the manipulator and consider $\mathrm{SO}(2)$ symmetry. Moreover, we perform the diffusion, denoising and guidance on the state space manifold instead of the Euclidean space, which is crucial for *trajectory-level* diffusion. Instead of conditioning the model to high-dimensional raw observations we achieve equivariance via a novel *positive-negative embedding* framework. Symmetry-breaking effects have led to active research (Finzi et al., 2021; Romero and Lohit, 2022) by designing parametric layers that relax the equivariant constraints (Wang et al., 2024b; Pertigkiozoglou et al., 2024b). Leveraging diffusion models to handle symmetry-breaking effects is novel and can impact these fields too.

## 3. Method

In this section we introduce our method - STRiDE: State-space Riemannian Diffusion for Equivariant Planning. After formulating the problem we discuss how to perform diffusion, denoising and guidance on the state space manifold and how to incorporate equivariance in the system.

### 3.1. Problem Formulation and Notation

Let $\mathcal{E} = \{O_1, \cdots, O_K\}$ describe an environment with $K$ collidable objects represented as 3d point clouds i.e. $O_i = \{p_i \in \mathbb{R}^3 | i \in [K_i]\}$. Let $\mathcal{C}$ be the configuration space of the robot, $q := (\theta_1, \cdots, \theta_n) \in \mathcal{C}$ a configuration consisting of $n$ 1-DoF joints and $S$ the state space of the robot

i.e. $s := [q, \dot{q}] \in S$ including the joint velocities $\dot{q} \in T_q\mathcal{C}$. Given two points from the state space $s_{start}, s_{goal}$ our goal is to find a smooth, collision-free trajectory represented as $T$ ordered waypoints $\tau = (s_1, s_2, \cdots, s_T)$ with $s_1 = s_{start}, s_T = s_{goal}$ respecting physical, kinematic constraints and other user defined costs encoded in the functional $J : S^T \to \mathbb{R}_+$. We consider kinematic motion planning on the state space (which can account for non-holonomic constraints) and assume that a low-level controller will execute the state transition.

In the supplementary material, we include analytical notation where we distinguish between $\mathrm{SO}(2) = \{R_\theta \in \mathbb{R}^{2\times 2} | R_\theta^T R_\theta = I, \det(R_\theta) = 1, \theta \in [-\pi, \pi)\}$, the parametrization $\theta \in [-\pi, \pi) \subset \mathbb{R}$ and the embedding $S^1 = \{(\cos\theta, \sin\theta) | \theta \in [-\pi, \pi)\} \subset \mathbb{R}^2$ and explain the isomorphisms, we overload the definitions of the exponential map $\exp : \mathfrak{so}(2) \to \mathrm{SO}(2)$ by identifying the Lie algebra $\mathfrak{so}(2)$ with the parametrization space, we define $Exp : \mathbb{R} \to S^1$, $Exp(\theta) = (\cos\theta, \sin\theta)$ and appropriately restrict it to define its inverse $Log : S^1 \to \mathbb{R}$, $Log(x, y) = atan2(y, x)$. We distinguish 1) the planning problem time $t \in T$, which we use as subscript and 2) the diffusion process time $i \in N$ which we use as superscript i.e. $\tau^i = (s_t^i)_{t\in T}$ is the $i$-times diffused trajectory from $\tau = \tau^0$; the denoised trajectory. We denote by $\tau_q, \tau_{\dot{q}}$ the configuration and velocity parts of the state space trajectory. We also provide preliminaries on diffusion models for motion planning.

### 3.2. State Space Riemannian Diffusion for Motion Planning

In this section, we develop our method. We focus on a state space of $n$ revolute joints, since it has non-trivial topology but our method extends seamlessly to prismatic joints. In our case, $S \subset T\mathbb{T}^n \simeq \mathbb{T}^n \times \mathbb{R}^n$. One option to solve the wrap-around problem is to operate directly on the quotient space $\mathbb{R}/2\pi\mathbb{Z}$. However, this (mod) representation creates discontinuous target trajectories for the neural network that operates in Euclidean space and makes the denoising particularly hard. To solve both the wrap-around and the trajectory continuity problem, we propose a diffusion model that operates on the *embedding* of the state space. We embed each degree of freedom of the hypertorus $\mathbb{T}^n$ separately and work on the product manifold $(S^1)^n \subset \mathbb{R}^{2n}$ so that the representation remains disentangled. The states are represented in this embedding space as:$\tilde{s} = \{(\cos\theta_1, \sin\theta_1, \cdots, \cos\theta_n, \sin\theta_n), (\omega_1, \cdots, \omega_n)\}$, i.e. $\tilde{\tau} \in \mathbb{R}^{(2n+n)T}$. Since we still use the intrinsic coordinates ($\theta$'s) and not project to unit circles in $\mathbb{R}^2$, a description in $SO(2)^n$ is more natural due to the connection with the lie algebra $\mathfrak{so}(2)^n$ which we will use to discuss the three stages of *diffusion, denoising, guidance* next. When appropriate we leverage the isomorphism $S^1 \simeq SO(2)$ to reduce the representation. In $S^1$ the variables are denoted as $\tilde{\tau}$, in $SO(2)$ as ${}^R\tau$ and in the lie algebra as $\tau$.

#### 3.2.1. RIEMANNIAN DIFFUSION AND DENOISING

Inspired by Leach et al. (2022) who perform diffusion on $SO(3)$ using the isotropic normal we design an $SO(2)$ analogue, which we name $\mathcal{IG}_{\mathrm{SO}(2)}(R, \sigma^2)$ and is defined as the pushforward of the gaussian measure through the exponential map i.e. $\mu_{\mathcal{IG}_{SO(2)}(R_\mu,\sigma^2)}(A) := \mu_{\mathcal{N}(\mu,\sigma^2)}(\log A), A \subset SO(2)$. Since $\mathcal{IG}_{SO(2)}(R_\mu, \sigma^2)$ is contained in a 1d-submanifold of $\mathbb{R}^{2\times 2}$ it is not absolutely continuous w.r.t. the Lebesque measure, thus it does not have a density. But we get exact samples $R \sim \mathcal{IG}_{SO(2)}(R_\mu, \sigma^2)$ via $\theta \sim \mathcal{N}(\theta; \mu, \sigma^2), R = \exp(\theta)$. We can also relate it to a corresponding density of an intrinsic parametrization of $SO(2)$, such as the lie algebra $\mathfrak{so}(2)$ which reveals the connection with the representation of Jing et al. (2022) at least in the diffusion stage.

**Lemma 1** *If $R \sim \mathcal{IG}_{SO(2)}(R_\mu, \sigma^2)$ then $\theta := log(R) \sim \mathcal{WN}(\theta; \mu mod 2\pi, \sigma^2)$, where $\mathcal{WN}(\theta; \mu, \sigma^2)$ is the wrapped Gaussian with location,uncertainty parameters $\mu \in [-\pi, \pi)$, $\sigma > 0$ and density: $\mathcal{WN}(\theta; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{k=-\infty}^{\infty} \exp\left(-\frac{(\theta-\mu-2\pi k)^2}{2\sigma^2}\right)$, $\theta \in [-\pi, \pi)$.*

We remind that our values are still on $SO(2)$ and not $[-\pi, \pi)$. There are some qualitative differences between $\mathcal{WN}(\mu, \sigma^2)$ and $\mathcal{IG}_{SO(2)}(R, \sigma^2)$. For example, while in $[-\pi, \pi)$ the wrapped normal does not necessarily have mean proportional to $\mu$ we can use the circular mean (Mardia and Jupp, 2009) $\mathbb{E}[\cos\theta + i\sin\theta] = e^{-\sigma^2}(\cos\mu + i\sin\mu)$ to prove that $\mathcal{IG}_{SO(2)}(R_\mu, \sigma^2)$ indeed has a mean proportional to $R_\mu$: $\mathbb{E}_{R\sim\mathcal{IG}_{SO(2)}(R_\mu,\sigma^2)}[R] = e^{-\sigma^2}R_\mu$. Our representation has deep connections to directional statistics (Mardia and Jupp, 2009).

Standard diffusion models perform efficient sampling of the forward process by performing multiscale diffusion in a single step. We can show that $\mu_{\mathcal{IG}_{\mathrm{SO}(2)}(R,\sigma^2)}$ is closed under (measure) convolutions. It is known that $\phi_1 \sim \mathcal{WN}(\mu_1, \sigma_1^2)$ and $\phi_2 \sim \mathcal{WN}(\mu_2, \sigma_2^2)$ then $\phi = (\phi_1 + \phi_2)\mathrm{mod}2\pi \sim \mathcal{WN}((\mu_1 + \mu_2)\mathrm{mod}2\pi, \sigma_1^2 + \sigma_2^2)$ (Jammalamadaka et al., 2001). From this using the exponential map it is straightforward to show that if $R_{1/2} \sim \mathcal{IG}_{SO(2)}(R_{\mu,1/2}, \sigma_{1/2}^2)$ then $R_1R_2 \sim \mathcal{IG}_{SO(2)}(R_{\mu,1}R_{\mu,2}, \sigma_1^2 + \sigma_2^2)$. Moreover, the wrapped Gaussian fulfills a central limit theorem on the circle (and limiting distribution of random walks) which can be extended to $\mathcal{IG}_{SO(2)}$ on $SO(2)$ due to closure of measure (Parthasarathy, 1964). For the n-fold product measure we overload the notation: $R \sim \mathcal{IG}_{SO(2)^n}(R_\mu, \sigma)$, where now $R, R_\mu, \sigma$ are n-dimensional lists. Whenever an operation (like multiplication) is between lists it is assumed to be pointwise.

**Diffusion:** Suppose the expert planner $P(s_{start}, s_{goal}, T, \mathcal{E})$ is queried to provide trajectories $\tau^0 = (s_t^0)_{t\in[T]}$ with $s_t^0 \in [-\pi, \pi)^n \times \mathbb{R}^n$ which we gather in a dataset $\mathcal{D}$. Then, let $\tau^0 \sim q(\tau^0|\mathcal{E})$. Using the closure under convolutions we can perform forward sampling in one step. For $i \in [N]$, $R^{\epsilon_i} \sim \mathcal{IG}_{SO(2)^{nT}}(I, (1-\bar{\alpha}_i))$ then for the configuration part of the trajectory $\tau_q$ we create the noisy trajectories as ${}^R\tau_q^i = R_i^\epsilon \exp(\sqrt{\bar{\alpha}_i}\tau_q^0)$. Then, $\tilde{\tau}^i \leftarrow ({}^R\tau_q^i, \tau_{\dot{q}}^i)$ where we use $SO(2) \simeq S^1$ for the configuration space while for the angular velocities we use standard Euclidean diffusion.

**Denoising:** We focus on the configuration space and follow Leach et al. (2022). Since $SO(2)$ is compact we sample uniformly (as $\theta \in U[-\pi, \pi), R = \exp\theta$): ${}^R\tau_q^N \sim U_{SO(2)^{nT}}$. We parametrize the inverse process as $p_w({}^R\tau_q^{i-1}|{}^R\tau^i, i) = \mathcal{IG}_{SO(2)^{nT}}(R_\mu({}^R\tau^i, i; w), \tilde{\beta}_i)$, where $R_\mu({}^R\tau^i, i; w) = \exp\left(\frac{\sqrt{\alpha_{i-1}}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\log{}^R\tau_q^i\right)\exp\left(\frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\log{}^R\mu_w({}^R\tau^i, i)\right)$ and ${}^R\mu_w({}^R\tau^i, i) = \exp\left(\frac{1}{\sqrt{\bar{\alpha}_i}}\log{}^R\tau^i\right)\cdot \exp\left(-\frac{1}{\sqrt{1-\bar{\alpha}_i}}Log\epsilon_w^q(\tilde{\tau}^i, i)\right)$. The network is $\epsilon_w : (S^1)^n \times \mathbb{R}^n \times [N] \to (S^1)^n \times \mathbb{R}^n$ and $\epsilon_w^q, \epsilon_w^{\dot{q}}$ the configuration and velocity parts. Here we leverage $SO(2) \simeq S^1$ to reduce the input size. The network predicts the noise on $S^1$ which is easier than predicting in $SO(2)$ since the former only needs a normalization, while the later has orthogonality and determinant constraints that need to be satisfied too. Note that if we move between the $\epsilon$ and $(\cos\epsilon, \sin\epsilon)$ we do not actually need to normalize the output since this is done by $arctan$. Our representation imposes continuity in the input of the network for continuous trajectories. A quotient representation that uses $\theta mod 2\pi$ to represent the input trajectories would not be continuous since the operator is not continuous.

**Training Loss:** $\mathcal{L}(w) = \mathbb{E}\left[d\left(Exp(\frac{1}{\sqrt{1-\bar{\alpha}_i}}\log R^\epsilon), \epsilon_w^q(\tilde{\tau}^i, i)\right) + \|\Omega - \epsilon_w^{\dot{q}}(\tilde{\tau}^i, i)\|_2^2\right]$ where the expectation is over $(i, \Omega, \tau^0) \sim \mathcal{U}(1, N) \times \mathcal{N}(0, I) \times q(\tau^0|\mathcal{E})$ and $R^\epsilon|i \sim \mathcal{IG}_{SO(2)^{nT}}(I, \sqrt{1-\bar{\alpha}_i})$. $\tilde{\tau}^i$ is described in the diffusion step and $d : \mathbb{T}^n \times \mathbb{T}^n \to \mathbb{R}_+$ is the chordal distance on torus here.

### 3.2.2. RIEMANNIAN GUIDANCE

We observe that the update step in MPD during posterior sampling can be conceived as a two-stage process: 1. deterministic denoising: $\tau^{i-1} \leftarrow \mu_w(\tau^i, i)$ and 2. (noisy) gradient descent (around $J^{i-1}$): $\tau^{i-1} \leftarrow \tau^{i-1} - \eta_{i-1}\nabla J^{i-1}(\tau^{i-1}) + \tilde{\beta}_i\mathbf{z}, \mathbf{z} \sim \mathcal{N}(0, I)$. Thus, guidance tries to minimize the cost term $J^{i-1}$ in a neighborhood around the denoised sample $\tau^i$. However, this optimization

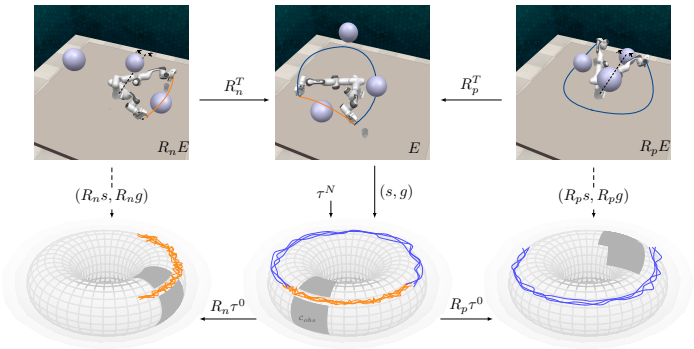


Figure 2: **Equivariant Priors**: Given an environment $\mathcal{E}$, we first canonicalize it with a learned equivariant frame ($R_n, R_p$ in the figure), together with the start and goal states. We perform diffusion on the canonical triplet $(\mathcal{E}, s, g)$ and then decanonicalize the resulting trajectories by applying the same frame reversely. Following the arrows one can see that this is the desired output as if we applied the diffusion on one of $\{(R_n\mathcal{E}, R_ns, R_ng), (R_p\mathcal{E}, R_ps, R_pg)\}$ (depending on whether the sampled trajectory is orange or blue). In the figure you can see the joint limits (at $-\pi, \pi$ for simplicity). Only one branch of trajectories is feasible for a specific rotation of the environment. Both can be sampled in the diffusion model, which is trained on the canonical environment (middle), but the one that goes through the joint limits will be discarded after de-canonicalization.

uses gradient descent in the Euclidean space which requires clipping or small gradient steps to converge which is inefficient. First we note that costs are expressed on the trajectories which live on the state space manifold and not in the Euclidean space. Then, $c(\theta)$ with $\theta \in [-\pi, \pi)$ can be written in $SO(2)$ as $c(\theta) = c(\log(R_\theta)) = (c \circ \log)(R_\theta)$. By operating on the manifold we can also introduce more useful metrics that have been well studied on $SO(2)$ (Chirikjian and Kyatkin, 2016). For example, depending on the requirements we can either optimize the cost that accounts for the longer arclength $d(R_1, R_2) = \| \log R_1 - \log R_2 \|_2$ as before, or costs that minimize the geodesic distance $d(R_1, R_2) = \| \log(R_1^T R_2) \|$. To optimize $J : SO(2)^n \to \mathbb{R}_+$ we will perform Riemannian gradient descent (RGD) (Boumal, 2023) on $SO(2)^n$. At iteration $k$, given gradient step $\alpha_k$, RGD updates the current state as $R_{k+1} = R_k \exp(-\alpha_k R_k^{-1} \nabla J(R_k))$, where multiplication between lists is pointwise. We use the uniformity of tangent spaces on lie groups to simplify to an expression amenable to automatic differentiation. The proof is included in Supplementary Material.

**Lemma 2** *The RGD update on $SO(2)^n$ can be written:* $R_{k+1} = R_k \exp\left(-\alpha_k \nabla_\theta J(R_k \exp(\theta))|_{\theta=0}\right)$.

All trajectories are on the manifold after each guidance step. These are subsequently given as input to the network for the next denoising step on the manifold. Reversely, denoising provides RGD with a trajectory on the manifold to initialize the updates. We visualize all steps in Fig. 1. We use similar costs as in Carvalho et al. (2023) but reparametrized on the manifold as explained above.

### 3.3. Equivariant Motion Planning by Positive-Negative Embedding

Planning on the state space removes the representation redundancy by constraining the learning on the actual degrees of freedom. Another form of redundancy is the selection of the coordinate system. Equivalent trajectories arise by rotating the environment and start and goal states around the base, and we can exploit this inductive bias to learn a more generalizable prior. As in previous works (Carvalho et al., 2023) we avoid conditioning the diffusion model on the environment to keep it lightweight. In Carvalho et al. (2023), the only means to adapt the trajectories to the environment is diffusion-guidance, which has been successful for local changes but fails for large changes that

can happen during execution or from different initialization. In those cases, full retraining might be required! To adapt the planner to large changes and exploit trajectory symmetry, while staying lightweight we propose **to learn to plan on a transformed space** and subsequently deploy the plan on the original space. This includes mapping to a **learned canonical environment** w.r.t. rotations around the base and **relaxing constraints on the state-space** to deal with symmetry-breaking due to joint limits, but utilizing the probabilistic framework to keep alternative trajectories.

**Learning a Canonical Environment:** We propose to *learn an equivariant frame* and learn on the warped environment after applying this frame. Moreover, the frame has to equivary smoothly with the rotations of the environment around the base. For example, the PCA eigenvectors on the point cloud $\mathcal{E}$ do not satisfy the smoothness requirement due to sign ambiguity that creates many possible frames for each rotation. They are also very susceptible to noise. Instead we use a small $SO(3)$-equivariant network (Thomas et al., 2018) built using e3nn library (Geiger et al., 2022). The network $f$ does not output a 3d frame only a single 3d vector $\boldsymbol{v} = f(\mathcal{E})$, from which we create $R_{\boldsymbol{v}} = \begin{pmatrix} \hat{\boldsymbol{v}}_x & -\hat{\boldsymbol{v}}_y & 0 \\ \hat{\boldsymbol{v}}_y & \hat{\boldsymbol{v}}_x & 0 \\ 0 & 0 & 1 \end{pmatrix}^T$, where $\hat{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\|(\boldsymbol{v}_x, \boldsymbol{v}_y)\|}$. The canonical environment is then computed as $\mathcal{E}^c = R_{\boldsymbol{v}}\mathcal{E}$ where $R\mathcal{E} = (R\mathcal{O}_i)_{i\in[K]}$. It is easy to see that the canonical environment remains invariant to $SO(2)$-rotations. For all $R \in \mathrm{SO}(2)_z \subset \mathrm{SO}(3)$ ($SO(2)_z$ is isomorphic to $SO(2)$ but lifted along z to act on 3d vectors), due to equivariance of $f$ we have $f(R\mathcal{E}) = Rf(\mathcal{E}) = R\boldsymbol{v}$. Then, $\hat{R\boldsymbol{v}} = R\hat{\boldsymbol{v}}$ and $R_{R\hat{\boldsymbol{v}}} = R_{\boldsymbol{v}}R^T$ from which we get $(R_{\boldsymbol{v}}R^T)(R\mathcal{E}) = R_{\boldsymbol{v}}\mathcal{E} = \mathcal{E}^c$. Also the frame is a smooth function of the environment by construction. We denote the action of the rotation on the state space as $R_{\boldsymbol{v}}s = ((\log(R_{\boldsymbol{v}} \exp s_1), \omega_1), s_2, \cdots, s_n)$.

**Set Equivariance via Positive-Negative Embedding:** An important intricacy regarding feasibility arises due to joint limits. If $\tau \in P(s, g, T, \mathcal{E})$ (here inclusion means that the trajectory is kinematically feasible for the problem) then $R^T\tau \in P(Rs, Rg, T, R\mathcal{E})$ only for some rotations $R \in SO(2)$ as long as the resulting trajectory does not go through the joint limits. Any deterministic equivariant model (independently of whether the environment is conditioning the model or not) is doomed to predict the rotated trajectory even beyond joint limits thus overconstraining the problem. However, if we assume that for a given environment, start and goal there is a feasible trajectory contained between the start and goal then even if we cannot get a feasible trajectory by rotating-back the predicted trajectory (call it *positive*) there is another (not necessarily unique) trajectory (call it *negative*) that starts and ends at the corresponding states, is feasible and the union of the ranges cover the whole $SO(2)$ (minus the joint limit space). At most one of the two is feasible for a given rotation of the environment (the ranges though depend on the trajectories). Thus, together they form an *equivariant pair* Fig. 2. The *negative* trajectory might differ a lot from the original depending on the environment. We cannot construct it as a transformation of the *positive* but we can query the planner appropriately to provide one such negative (if it exists) as we show next. If the joint limits are at $-\pi \leq \theta_{min} < \theta_{max} \leq \pi$ then for a fixed environment $\mathcal{E}$ and $(s, g)$ we query:

$$\tau^p \sim R^T_{\theta_{min}-s_0} P(R_{\theta_{min}-s_0}s, R_{\theta_{min}-s_0}g, T, R_{\theta_{min}-s_0}\mathcal{E}) \tag{1a}$$

$$\tau^n \sim R^T_{\theta_{max}-s_0} P(R_{\theta_{max}-s_0}s, R_{\theta_{max}-s_0}g, T, R_{\theta_{max}-s_0}\mathcal{E}) \tag{1b}$$

With these queries we guarantee that the planner will never return two positive/negative trajectories, since any feasible trajectory for the first equation necessarily crosses the joint limits in the second equation. Thus, the expert planner has to select the opposite path that connects $s, g$ in the circle if one exists. We create a dataset from these pairs. We will utilize the power of multimodality of diffusion models in order to embed both trajectories (that are feasible for *different* environments)

in the same start and goal position for the *canonical space*, thus creating an equivariant prior. I.e. $(s, g) \mapsto (R_{\boldsymbol{v}} s, R_{\boldsymbol{v}} g)$ and $(\tau^p, \tau^n) \mapsto (R_{\boldsymbol{v}} \tau^p, R_{\boldsymbol{v}} \tau^n)$. We can have multiple positive (and negative) trajectories from the same start and goal. Every query to a non deterministic expert (e.g. sampling-based) will give a different one. The distribution on the trajectories given $s, g$ does not need to be uniform although having them balanced would help during generation. During training the diffusion model $\epsilon(\tilde{\tau}^i, i)$ will learn to denoise both $\tilde{\tau}^p, \tilde{\tau}^n$ in the canonical environment. The planner operates on a continuous base joint to produce alternative paths (even infeasible in the transformed environment) that will be mapped to the original space. (The joint limit cost from Carvalho et al. (2023) is also not used). During inference, given $(s, g, \mathcal{E})$ we canonicalize the environment and start and goal states and sample trajectories from $(f(\mathcal{E})s, f(\mathcal{E})g)$, including both branches. The costs are also canonicalized using $f(\mathcal{E})J$ and since all costs are assumed to be scalar-fields for rotation ($J(\tau) \in \mathbb{R}_+$), the particular action of $SO(2)$ is: $(f(\mathcal{E})J)(\tau) = J(f(\mathcal{E})^{-1}\tau)$. Guidance will be performed in both branches and the generated trajectories will be decanonicalized as $f(\mathcal{E})^T\tau$. The infeasible branch will be discarded after simulating the trajectory from the original $s, g$ by checking which of the two crosses the joint limits. See Fig.2. If we denote our planner with $\hat{P}$ and $\tau \in \hat{P}(s, g, T, \mathcal{E})$ a predicted feasible trajectory, then we can show the following statement which can be written as *set- equivariance without elementwise equivariance*:

**Lemma 3** *If* $\exists R_1, R_2 \in SO(2) : \tau^p \in \hat{P} \circ R_1, \tau^n \in \hat{P} \circ R_2$ *then* $\forall R \in SO(2) : RR_1^{-1}\tau^p \in \hat{P} \circ R$ *or* $RR_2^{-1}\tau^n \in \hat{P} \circ R$ *but not both.*

## 4. Experiments

In this section, we verify our claims through simulation experiments, and answer the following questions: (1) Is our on-manifold diffusion model more effective in achieving lower costs and hence, better performance with fewer guidance steps? (2) Can we learn feasible plans that are also generalizable to different transformations of the environment?

**Environments and Tasks** We evaluate our algorithm on the 7-dof Emika Franka Panda arm that is deployed in two environments - the `PandaSpheres` in Isaac Gym, as described in Carvalho et al. (2023), and a custom environment shown below in `CoppeliaSim` integrated with `RLBench`. The custom environment has spherical obstacles on the right, and a shelf in close vicinity on the left while being restricted by the table from below. The task in `PandaSpheres` environment is to generate feasible trajectories from random initial and final states, while minimizing an objective cost function, thus, providing a venue for fair baseline comparison and planner assessment. Our custom environment is more task-oriented, where the success criterion requires not only collision-free navigation but also planning/replanning feasible trajectories from random-initialized positions, picking up the cup, and placing it at a given position on the shelf while keeping it upright.

**Algorithms and Baselines** We compare our proposed algorithm's performance against the RRT-Connect + GPMP which is a sampling-based optimization planner, and MPD (Carvalho et al., 2023) in the canonical environment (MPD Canonical). The MPD Canonical model operates over 5 cycles where each cycle consists of 25 diffusion steps, followed by 5 guidance steps, whereas our model executes the same 5-cycle process, but with only 2 guidance steps per cycle. Moreover, in the rotation augmentation test, the trajectories are first obtained in the canonical environment and then rotated back to the original environment. We also consider EQ-prior-guidance, which consists of our Equivariant On-manifold diffusion-based planner as Equivariant priors that are denoised for a total of 125 steps, followed by 10 guidance steps.

**Metrics** We chose 5 metrics to assess our performance with that of the baselines - (1) $\mathcal{S}$ denotes

the success rate of the trajectory i.e., for a given trial, the success rate is one if at least one of the trajectories in the output batch is feasible. (2) $\mathcal{C}_s$ denotes the smoothness cost, which is a measure of how smooth on average, the trajectories are in the batch. While the Gaussian process promotes smoothness and thus, lowers $\mathcal{C}_s$ we compute it as the average of the sum of pairwise norm of the velocities of the trajectories, as done in Carvalho et al. (2023) to keep the comparison fair. (3) $\mathcal{C}_p$ denotes the path length cost that is computed as the average of the sum of the pairwise norm of the joint angles. (4) $\mathcal{C}_b$ denotes the *best cost* (least) (sum of path length and smoothness costs) that a trajectory exhibited in the batch. (5) $t$ - denotes the overall inference time that the planner took to output a batch of 50 trajectories.

| | $\mathcal{S}\uparrow$ | $\mathcal{C}_s\downarrow$ | $\mathcal{C}_p\downarrow$ | $\mathcal{C}_b\downarrow$ | $t\downarrow$ | |
|---|---|---|---|---|---|---|
| RRTC+GPMP | 1.0 | – | $8.1 \pm 1.1$ | – | $226.14 \pm 13.4$ | $\mathbb{E}_{q_i,q_f}[.]$ |
| MPD Canonical | $1.0 \pm 0.0$ | $7.6 \pm 3.41$ | $6.5 \pm 2.74$ | $11.54 \pm 6.07$ | $23.12 \pm 1.1$ | |
| **STRiDE** | $1.0 \pm 0.0$ | $8.7 \pm 1.7$ | $7 \pm 1.45$ | $12.6 \pm 2.92$ | $9.98 \pm 0.9$ | |
| MPD Canonical | $0.43 \pm 0.22$ | $8.77 \pm 2.82$ | $4.65 \pm 1.49$ | $11.15 \pm 3.68$ | – | $\mathbb{E}_{q_i,q_f}\mathbb{E}_g[.]$ |
| **STRiDE** | $0.97 \pm 0.03$ | $8.61 \pm 1.57$ | $7.01 \pm 0.9$ | $12.9 \pm 2.31$ | – | |
| EQ-Prior-Guidance | $0.85 \pm 0.17$ | $9.3 \pm 1.46$ | $7.51 \pm 1.83$ | $13.96 \pm 3.02$ | $10.31 \pm 1.85$ | |

Table 1: Metrics are reported as $(\mu \pm \sigma)$ with $\uparrow$ indicate that higher values are better, and with $\downarrow$ indicate that lower values are better.

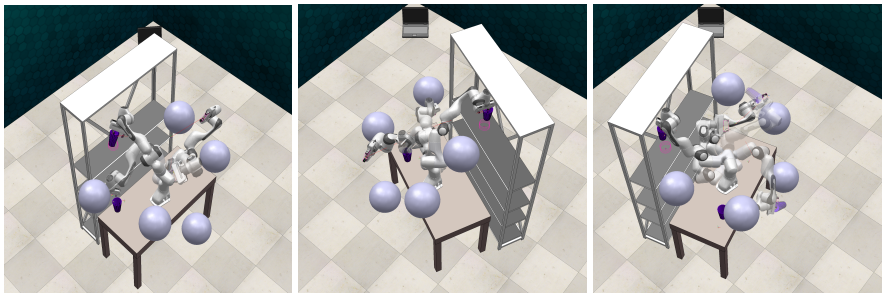

Figure 3: (Left) The canonical environment. (Middle) A random rotation of the environment. (Right) A random rotation of the environment, while the spherical obstacles are slightly perturbed away from their usual positions (global+local).

**Discussion** The results in the `PandaSpheres` Environment are summarized in Table 1. The first three rows depict the performance of each of the algorithms in the canonical frame only, where the results are averaged over 10 initial and final configurations, randomly chosen. The last three rows depict the performance of each of the algorithms with 10 randomly sampled initial and final configurations, but each consisting of 72 rotation transformations i.e., $5^\circ$ increments in the range of $[0 - 360)^\circ$ of the environment. We can infer from the table that (1) STRiDE compares similarly to MPD in the canonical frame, albeit with lower variations and lower guidance steps. (2) STRiDE obtains an overall success rate of $97\%$, while the MPD canonical achieves about $43\%$, due to the lack of the *negative trajectory*. Although our path length appears to be high, this is precisely because our formulation can predict the *negative* trajectory in cases where MPD Canonical fails. We do so with fewer guidance steps as compared to MPD canonical, thereby achieving a more efficient planner with faster inference time. (3) Our performance across all metrics is better than the EQ-Prior-Guidance model, even though the total number of denoising steps and guidance steps remains

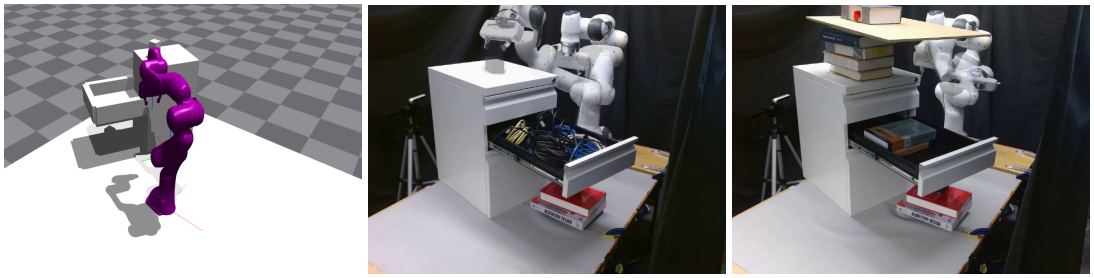

Figure 4: Left figure depicts the canonical environment in the simulation (random $SO(2)$ rotation about the base by $148.6°$). The middle figure is a real-world replication, with a rotation of $17°$, with the place goal being on top of the cabinet, with the drawer full. The figure on the right is the same but with additional obstruction added from the top during test time, with the place goal being inside the drawer (partially free). In all three environments, the pick goal is under the opened drawer[1].

the same. This empirically shows us that the optimization is more effective when *interleaved* with smaller chunks of denoising steps over multiple loops.

**RLBench experiments:** Traditional motion planners often infer the configuration-space trajectory based on the end-effector path by solving the inverse-kinematics problem at each stage, with some suboptimal velocity profile like constant velocity. This may be particularly disadvantageous in a cluttered environment. We base our custom environment in this regard by making the workspace cluttered with spherical obstacles and following a *task-based* theme, that allows for replanning between two waypoints. The arm configuration and the cup position (within the workspace in the front) are randomly initialized in the beginning, and the goal is fixed on a shelf in close vicinity to the arm. Fig. 3 describes the simulation environment, along with two variations. In all three cases, we can see that the proposed STRiDE succeeds in finding a feasible trajectory, and hence completing the task by succeeding with an average of 8.2 out of 10 success rate, within an average inference time of $9.12s$. The inverse-kinematics motion planner in RLBench has an average of 4.6 out of 10 success rate, with an average inference time of $12.32s$.

**Real world experiments:** To test our motion planner on real-world tasks, we consider a pick-place experiment in a relatively cluttered environment, as shown in Fig. 4. The goal is to pick an object from underneath the middle drawer of a cabinet and place it on top of the cabinet, or inside the drawer. The environment is obstructed from the top during test time (right figure in Fig. 4, with reduced workspace. We train the diffusion model on a simpler canonical environment with only the cabinet present, using only 50 contexts with 25 trajectories each. The rotation for the canonical environment is randomly chosen to be $148.6°$ about the base joint, while during testing, the rotation is randomly chosen to be $\sim 17°$, with local changes to the cabinet itself with test-time obstructions from the top. The `Emika Franka Panda` arm is run using the *effort trajectory controller*, to which we provide a subsampled set of state waypoints. Our motion planner can generalize well to these transformed versions of the environment, producing feasible (i.e., collision-free and smooth) trajectories. The video recordings for the hardware experiments are provided here[1].

**Conclusion:** We propose *STRiDE*, a novel diffusion motion planner that is topologically and symmetry informed. By operating on the state space manifold during all stages of diffusion, denoising and guidance it achieves planning efficiency and by learning an equivariant prior it achieves generalization, advancing beyond prior limitations. We verify our claims with simulated and real-world experiments on static manipulation. In future research we aim to integrate our planner in a closed-loop framework and exploit its fast replanning capabilities.

1. https://drive.google.com/drive/folders/1G-f2X0aSm14Q2knOqF3CGxvnb6AdUNQX?usp=sharing

## 5. Acknowledgements

We would like to thank Office of Naval Research grant N00014-22-1-2677 for supporting our research.

## References

Mayur J. Bency, Ahmed Hussain Qureshi, and Michael C. Yip. Neural path planning: Fixed time, near-optimal path generation via oracle imitation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019*, pages 3965–3972. IEEE, 2019. doi: 10.1109/IROS40897.2019.8968089. URL https://doi.org/10.1109/IROS40897.2019.8968089.

Valentin De Bortoli, Emile Mathieu, Michael John Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=oDRQGo8I7P.

Nicolas Boumal. *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, 2023.

Johann Brehmer, Joey Bose, Pim de Haan, and Taco S Cohen. Edgi: Equivariant diffusion for planning with embodied agents. In *Advances in Neural Information Processing Systems*, volume 36, pages 63818–63834. Curran Associates, Inc., 2023.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. *arXiv preprint arXiv:2104.13478*, 2021.

J. Carvalho, A.T. Le, M. Baierl, D. Koert, and J. Peters. Motion planning diffusion: Learning and planning of robot motions with diffusion models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

Junwoo Chang, Hyunwoo Ryu, Jiwoo Kim, Soochul Yoo, Joohwan Seo, Nikhil Prakash, Jongeun Choi, and Roberto Horowitz. Denoising heat-inspired diffusion with insulators for collision free motion planning. *arXiv preprint arXiv:2310.12609*, 2023.

Evangelos Chatzipantazis, Stefanos Pertigkiozoglou, Edgar Dobriban, and Kostas Daniilidis. $\mathrm{SE}(3)$-equivariant attention networks for shape reconstruction in function space. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=RDy3IbvjMqT.

Chang Chen, Fei Deng, Kenji Kawaguchi, Caglar Gulcehre, and Sungjin Ahn. Simple hierarchical planning with diffusion, 2024.

G.S. Chirikjian and A.B. Kyatkin. *Harmonic Analysis for Engineers and Applied Scientists: Updated and Expanded Edition*. Dover Books on Mathematics. Dover Publications, 2016. ISBN 9780486795645. URL https://books.google.com/books?id=EO1lDAAAQBAJ.

Ho Jin Choi and Nadia Figueroa. Towards feasible dynamic grasping: Leveraging gaussian process distance fields, se(3) equivariance and riemannian mixture models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. URL https://arxiv.org/abs/2311.02576.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/cohenc16.html.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

Marc Finzi, Gregory Benton, and Andrew G Wilson. Residual pathway priors for soft equivariance constraints. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 30037–30049. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/fc394e9935fbd62c8aedc372464e1965-Paper.pdf.

Jonathan D. Gammell, Siddhartha S. Srinivasa, and Timothy D. Barfoot. Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pages 2997–3004. IEEE, 2014. doi: 10.1109/IROS.2014.6942976.

Mario Geiger, Tess Smidt, Alby M., Benjamin Kurt Miller, Wouter Boomsma, Bradley Dice, Kostiantyn Lapchevskyi, Maurice Weiler, Michał Tyszkiewicz, Simon Batzner, Dylan Madisetti, Martin Uhrin, Jes Frellsen, Nuri Jung, Sophia Sanborn, Mingjian Wen, Josh Rackers, Marcel Rød, and Michael Bailey. e3nn/e3nn: 2022-04-13, April 2022. URL https://doi.org/10.5281/zenodo.6459381.

Yutong He, Naoki Murata, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Dongjun Kim, Wei-Hsiang Liao, Yuki Mitsufuji, J Zico Kolter, Ruslan Salakhutdinov, and Stefano Ermon. Manifold preserving guided diffusion. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=o3BxOLoxm1.

Owen Lewis Howell, David Klee, Ondrej Biza, Linfeng Zhao, and Robin Walters. Equivariant single view pose prediction via induced and restriction representations. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=dxVN2fZjx6.

Boce Hu, Xupeng Zhu, Dian Wang, Zihao Dong, Haojie Huang, Chenghao Wang, Robin Walters, and Robert Platt. Orbitgrasp: Se (3)-equivariant grasp learning. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=clqzoCrulY.

Chin-Wei Huang, Milad Aghajohari, Joey Bose, Prakash Panangaden, and Aaron Courville. Riemannian diffusion models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=ecevn9kPm4.

Brian Ichter, James Harrison, and Marco Pavone. Learning sampling distributions for robot motion planning. In *IEEE ICRA*, 2018.

S.R. Jammalamadaka, A. Sengupta, and A. Sengupta. *Topics in Circular Statistics*. Series on multivariate analysis. World Scientific, 2001. ISBN 9789812779267. URL https://books.google.com/books?id=sKqWMGqQXQkC.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *ICML*, 2022a.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9902–9915. PMLR, 17–23 Jul 2022b.

Royina Karegoudra Jayanth, Yinshuang Xu, Ziyun Wang, Evangelos Chatzipantazis, Daniel Gehrig, and Kostas Daniilidis. Eqnio: Subequivariant neural inertial odometry, 2024. URL https://arxiv.org/abs/2408.06321.

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24240–24253. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/994545b2308bbbbc97e3e687ea9e464f-Paper-Conference.pdf.

Sékou-Oumar Kaba, Arnab Kumar Mondal, Yan Zhang, Yoshua Bengio, and Siamak Ravanbakhsh. Equivariance with learned canonicalization functions. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022. URL https://openreview.net/forum?id=pVD1k8ge25a.

M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal. Stomp: Stochastic trajectory optimization for motion planning. In *IEEE International Conference on Robotics and Automation*, 2011.

L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 1996. doi: 10.1109/70.508439.

J.J. Kuffner and S.M. LaValle. Rrt-connect: An efficient approach to single-query path planning. In *IEEE ICRA*, 2000. doi: 10.1109/ROBOT.2000.844730.

Abhinav Kumar, Garrick Brazil, Enrique Corona, Armin Parchami, and Xiaoming Liu. DEVIANT: Depth EquiVarIAnt NeTwork for Monocular 3D Object Detection. In *ECCV*, 2022.

Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning, 1998.

Adam Leach, Sebastian M Schmon, Matteo T. Degiacomi, and Chris G. Willcocks. Denoising diffusion probabilistic models on SO(3) for rotational alignment. In *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*, 2022. URL https://openreview.net/forum?id=BY88eBbkpe5.

Zhixuan Liang, Yao Mu, Hengbo Ma, Masayoshi Tomizuka, Mingyu Ding, and Ping Luo. Skilldiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16467–16476, June 2024.

Aaron Lou, Minkai Xu, Adam Farris, and Stefano Ermon. Scaling riemannian diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=FLTg8uA5xI.

K.V. Mardia and P.E. Jupp. *Directional Statistics*. Wiley Series in Probability and Statistics. Wiley, 2009. ISBN 9780470317815. URL https://books.google.com/books?id=PTNiCm4Q-M0C.

K. R. Parthasarathy. The central limit theorem for the rotation group. *Theory of Probability & Its Applications*, 9(2):248–257, 1964. doi: 10.1137/1109037. URL https://doi.org/10.1137/1109037.

Stefanos Pertigkiozoglou, Evangelos Chatzipantazis, and Kostas Daniilidis. Biequiformer: Bi-equivariant representations for global point cloud registration, 2024a. URL https://arxiv.org/abs/2407.08729.

Stefanos Pertigkiozoglou, Evangelos Chatzipantazis, Shubhendu Trivedi, and Kostas Daniilidis. Improving equivariant model training via constraint relaxation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=tWkL7k1u5v.

Nathan Ratliff, Matt Zucker, J. Andrew Bagnell, and Siddhartha Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *IEEE International Conference on Robotics and Automation*, 2009. doi: 10.1109/ROBOT.2009.5152817.

David W. Romero and Suhas Lohit. Learning partial equivariances from data. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=pNHT6oBaPr8.

Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B. Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se(3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pages 6394–6400. IEEE, 2022. doi: 10.1109/ICRA46639.2022.9812146. URL https://doi.org/10.1109/ICRA46639.2022.9812146.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*. JMLR.org, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019.

Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds, 2018. URL https://arxiv.org/abs/1802.08219.

Marc Toussaint. Robot trajectory optimization using approximate inference. In *ICML*. Association for Computing Machinery, 2009. ISBN 9781605585161. doi: 10.1145/1553374.1553508.

Toshihide Ubukata, Jialong Li, and Kenji Tei. Diffusion model for planning: A systematic literature review, 2024. URL https://arxiv.org/abs/2408.10266.

Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se(3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *IEEE ICRA*, 2023.

Julen Urain, Ajay Mandlekar, Yilun Du, Mahi Shafiullah, Danfei Xu, Katerina Fragkiadaki, Georgia Chalvatzaki, and Jan Peters. Deep generative models in robotics: A survey on learning from multimodal demonstrations, 08 2024.

Dian Wang, Stephen Hart, David Surovik, Tarik Kelestemur, Haojie Huang, Haibo Zhao, Mark Yeatman, Jiuguang Wang, Robin Walters, and Robert Platt. Equivariant diffusion policy. In *8th Annual Conference on Robot Learning*, 2024a. URL https://openreview.net/forum?id=wD2kUVLT1g.

Jiankun Wang, Wenzheng Chi, Chenming Li, Chaoqun Wang, and Max Q.-H. Meng. Neural rrt*: Learning-based optimal path planning. *IEEE T-ASE*, 2020. doi: 10.1109/TASE.2020.2976560.

Rui Wang, Elyssa Hofgard, Han Gao, Robin Walters, and Tess Smidt. Discovering symmetry breaking in physical systems with relaxed group convolution. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=59oXyDTLJv.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data, 2018. URL https://arxiv.org/abs/1807.02547.

Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7168–7177, 2016. URL https://api.semanticscholar.org/CorpusID:206596746.

Cheng-Fu Yang, Haoyang Xu, Te-Lin Wu, Xiaofeng Gao, Kai-Wei Chang, and Feng Gao. Planning as inpainting: A diffusion-based embodied task planning framework for environments under uncertainty, 2023.