Diffusion-EDFs: Bi-equivariant Denoising Generative Modeling on SE(3) for Visual Robotic Manipulation

Supplementary Material

A. Bi-equivariance

For robust pick-and-place manipulation, the trained policy needs to be generalizable to previously unseen configurations of the target objects to pick/place. This can be achieved by inferring end-effector poses that keep the relative pose between the grasped object and the placement target invariant. Note that in our formulation, picking is essentially a special case of placing tasks, in which the gripper is *placed* at appropriate grasp points of the target object to pick with an appropriate orientation.

Consider the scenario in which the policy is trained with a demonstration (g_{we}, o_s, o_e) in which g_{we} is the end-effector pose, and o_s and o_e are respectively the point cloud observations of the scene and grasp. We denote the world frame using subscript w and the end-effector frame using subscript e. Note that o_s is observed in frame w and o_e in frame e. Now, let the placement target be moved by $\Delta g = g_{w'w}$, inducing the transformation of the observation $o_s \rightarrow \Delta g\, o_s$. This is equivalent to changing the world reference frame from w to w' with respect to the observation. Therefore, the end-effector pose should also be transformed equivariantly as $g_{we} \rightarrow g_{w'e} = \Delta g\, g_{we}$ (see Fig. 6-(a)). This *scene equivariance* is also referred to as *left equivariance* [37, 61], as the transformation Δg comes to the left side of g_{we}.

On the other hand, consider the transformation of the grasped object $\Delta g = g_{e'e}$, which induces the transformation of the observation $o_e \rightarrow \Delta g\, o_e$. This is equivalent to changing the end-effector reference frame from e to e' with respect to the observation. In the world frame, this corresponds to the transformation of the end-effector pose by $g_{we} \rightarrow g_{we'} = g_{we}\Delta g^{-1}$ (see Fig. 6-(b)). This *grasp equivariance* is also referred to as *right equivariance* [37, 61], as the transformation Δg^{-1} comes to the right side of g_{we}. Combining these left and right equivariance conditions, we obtain the bi-equivariance condition, which can be formally expressed in a probabilistic form as Eq. (10).

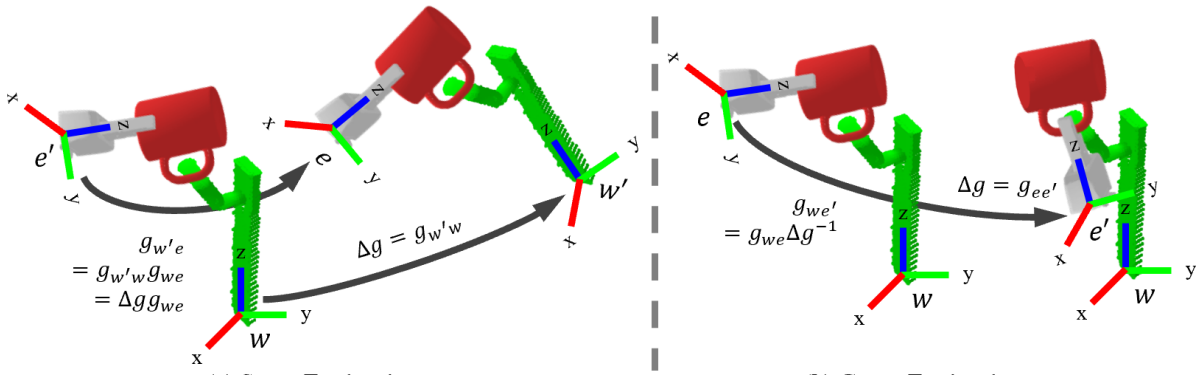

Figure 6. **Scene Equivariance and Grasp Equivariance. (a)** The end-effector pose must follow the transformation of the placement target within the scene. This *scene equivariance* can be achieved by multiplying the transformation Δg on the left side of the end-effector pose. Therefore, we also refer to this property as the *left equivariance*. **(b)** The end-effector pose must move contravariantly to the transformation of the grasped object to compensate for the changes. This *grasp equivariance* involves the inverse transformation Δg^{-1} coming to the right side of the end-effector pose. Therefore, we also refer to this property as the *right equivariance*.

B. Analytic Form of the Target Score in Eq. (21)

In this section, we provide the analytic form of the target score function in Eq. (21)

$$\nabla \log \mathcal{B}_t(g_{ed}^{-1} g_0^{-1} g g_{ed}) \tag{34}$$

By definition, the i-th component of the target score function is calculated as follows:

$$\begin{aligned}
\mathcal{L}_i \log \mathcal{B}_t(g_{ed}^{-1} g_0^{-1} g g_{ed}) &= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \log \mathcal{B}_t \left(g_{ed}^{-1} g_0^{-1} g \exp[\epsilon \hat{\boldsymbol{e}}_i] g_{ed}\right) \\
&= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \log \mathcal{B}_t \left(g_{ed}^{-1} g_0^{-1} g\, g_{ed} \exp[\epsilon Ad_{g_{ed}^{-1}} \hat{\boldsymbol{e}}_i]\right) \\
&= \left[\mathcal{L}_{Ad_{g_{ed}^{-1}} \hat{\boldsymbol{e}}_i} \log \mathcal{B}_t\right] \left(g_{ed}^{-1} g_0^{-1} g g_{ed}\right) \\
&= \left[\sum_{j=1}^{6} \left[Ad_{g_{ed}^{-1}}\right]_{ji} \mathcal{L}_j \log \mathcal{B}_t\right] \left(g_{ed}^{-1} g_0^{-1} g g_{ed}\right) \\
&= \sum_{j=1}^{6} \left[Ad_{g_{ed}^{-1}}\right]_{ji} \left[\mathcal{L}_j \log \mathcal{B}_t\right] \left(g_{ed}^{-1} g_0^{-1} g g_{ed}\right)
\end{aligned} \tag{35}$$

$$\Rightarrow \nabla \log \mathcal{B}_t(g_{ed}^{-1} g_0^{-1} g g_{ed}) = [Ad_{g_{ed}}]^{-T} \left[\nabla \log \mathcal{B}_t\right] \left(g_{ed}^{-1} g_0^{-1} g g_{ed}\right) \tag{36}$$

Therefore, all we need is the score of the Brownian diffusion kernel $\nabla \log \mathcal{B}_t(g)$ which can be decomposed into its translation and rotation parts using Eq. (5)

$$\nabla \log \mathcal{B}_t(g) = \nabla \log \mathcal{N}(\boldsymbol{p}; \boldsymbol{\mu} = \mathbf{0}, \Sigma = tI) + \nabla \log \mathcal{IG}_{SO(3)}(R; \epsilon = t/2) \tag{37}$$

where $\nabla \log \mathcal{N}(\boldsymbol{p}; \boldsymbol{\mu} = \mathbf{0}, \Sigma = tI) = -\boldsymbol{p}/t$ can be easily computed. A common practice for the calculation of the rotational part $\nabla \log \mathcal{IG}_{SO(3)}(R; \epsilon = t/2)$ is to use automatic differentiation packages [17, 34, 42, 61, 75, 84]. However, the explicit form can be easily calculated without automatic differentiation packages.

$$\mathcal{L}_i \ \log \mathcal{IG}_{SO(3)}(R; \epsilon) = \frac{\mathcal{L}_i \, \mathcal{IG}_{SO(3)}(R; \epsilon)}{\mathcal{IG}_{SO(3)}(R; \epsilon)} \tag{38}$$

$$\mathcal{L}_i \, \mathcal{IG}_{SO(3)}(R; \epsilon) = \sum_{l=0}^{\infty} (2l+1) \exp\left[-l(l+1)\epsilon\right] \left[\frac{(l+1)\sin(l\theta) - l\sin((l+1)\theta)}{\cos(\theta) - 1}\right] \left[\frac{-\mathrm{tr}\left[R[\hat{\boldsymbol{e}}_i]^\wedge\right]}{2\sin\theta}\right] \tag{39}$$

We denote the skew-symmetric matrix of the i-th $\mathfrak{so}(3)$ basis $\hat{\boldsymbol{e}}_i$ as $[\hat{\boldsymbol{e}}_i]^\wedge$, whose matrix element is $[\hat{\boldsymbol{e}}_i]^\wedge_{jk} = -\epsilon_{ijk}$ where ϵ_{ijk} is the Levi-Civita symbol.

The derivation is as follows. First, we rewrite Eq. (6) with the *character* $\mathcal{X}(R)$ of $SO(3)$ [85].

$$\mathcal{IG}_{SO(3)}(R; \epsilon) = \sum_{l=0}^{\infty} (2l+1) \exp\left[-l(l+1)\epsilon\right] \mathcal{X}_l(R) \tag{40}$$

$$\mathcal{X}_l(R) = \mathrm{tr}\left[D_l(R)\right] = \sin\left((2l+1)\frac{\theta}{2}\right) / \sin(\frac{\theta}{2}) \tag{41}$$

$\theta \in (0, \pi)$ is the rotation angle of R. Now we calculate the Lie derivative of $\mathcal{IG}_{SO(3)}$ as follows:

$$\mathcal{L}_i \, \mathcal{IG}_{SO(3)}(R; \epsilon) = \sum_{l=0}^{\infty} (2l+1) \exp\left[-l(l+1)\epsilon\right] \mathcal{L}_i \, \mathcal{X}_l(R) \tag{42}$$

$$\mathcal{L}_i \, \mathcal{X}_l(R) = \left[\frac{(l+1)\sin(l\theta) - l\sin((l+1)\theta)}{\cos(\theta) - 1}\right] \mathcal{L}_i \, \theta \tag{43}$$

$$\mathcal{L}_i \, \theta = \left[\frac{-1}{\sin\theta}\right] \mathcal{L}_i \left[\cos\theta\right] \tag{44}$$

The last line can be easily calculated using $\cos\theta = \frac{1}{2}\left(\mathrm{tr}\left[R\right] - 1\right)$ and $\mathcal{L}_{\mathcal{V}} \, \mathrm{tr}\left[R\right] = \mathrm{tr}\left[R[\mathcal{V}]^\wedge\right]$.

$$\mathcal{L}_i \left[\cos\theta\right] = \frac{1}{2}\left(\mathrm{tr}\left[R[\hat{\boldsymbol{e}}_i]^\wedge\right]\right) \tag{45}$$

Combining these results, one can derive Eq. (39). In practice, the infinite sum in Eq. (39) is approximated with $\sum_{l=0}^{l_{max}}$ where $l_{max} = 1000 \sim 10000$, which can be computed within a millisecond when appropriately parallelized. Although we have derived Eq. (39) for $\theta = (0, \pi)$, the result can be asymptotically extended to $\theta = 0$ and π as $\mathcal{IG}_{SO(3)}$ is an infinitely differentiable on $SO(3)$ [55].

C. Proofs and Derivations

C.1. Proof of Proposition 1

Proof of the left invariance of the score function.

$$\begin{aligned}
\mathcal{L}_i \log P(\Delta g\, g|\Delta g \cdot o_s, o_e) &= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \log P(\Delta g\, g \exp[\epsilon \hat{\boldsymbol{e}}_i]|\Delta g \cdot o_s, o_e) \\
&= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \log P(g \exp[\epsilon \hat{\boldsymbol{e}}_i]|o_s, o_e) \\
&= \mathcal{L}_i \log P(g|o_s, o_e)
\end{aligned}$$

where we used $P(\Delta g\, g|\Delta g \cdot o_s, o_e) = P(g|o_s, o_e)$ in the second line. □

Proof of the right equivariance of the score function.

$$\begin{aligned}
\mathcal{L}_i \log P(g\,\Delta g^{-1}|o_s, \Delta g \cdot o_e) &= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \log P(g\,\Delta g^{-1} \exp[\epsilon \hat{\boldsymbol{e}}_i]|o_s, \Delta g \cdot o_e) \\
&= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \log P(g\,\Delta g^{-1} \exp[\epsilon \hat{\boldsymbol{e}}_i]\Delta g|o_s, o_e) \\
&= \left.\frac{d}{d\epsilon}\right|_{\epsilon=0} \log P(g \exp\left[\epsilon Ad_{\Delta g^{-1}} \hat{\boldsymbol{e}}_i\right]|o_s, o_e) \\
&= \mathcal{L}_{Ad_{\Delta g^{-1}} \hat{\boldsymbol{e}}_i} \log P(g|o_s, o_e) \\
&= \mathcal{L}_{\sum_j \left[Ad_{\Delta g^{-1}}\right]_{ji} \hat{\boldsymbol{e}}_j} \log P(g|o_s, o_e) \\
&= \sum_{j=1}^{6} \left[Ad_{\Delta g^{-1}}\right]_{ji} \mathcal{L}_j \log P(g|o_s, o_e) \\
&\qquad (\because \text{Linearity of Lie-derivatives [13]}\ \ \mathcal{L}_{\sum_i v_i \hat{\boldsymbol{e}}_i} = \textstyle\sum_i v_i \mathcal{L}_i) \\
\Rightarrow \nabla \log P(g\,\Delta g^{-1}|o_s, \Delta g \cdot o_e) &= \left[Ad_{\Delta g^{-1}}\right]^T \nabla \log P(g|o_s, o_e) = [Ad_{\Delta g}]^{-T} \nabla \log P(g|o_s, o_e)
\end{aligned}$$

where we denote the (j, i)-th matrix element of $Ad_{\Delta g^{-1}}$ with $\left[Ad_{\Delta g^{-1}}\right]_{ji}$. We used $P(g\,\Delta g^{-1}|o_s, \Delta g \cdot o_e) = P(g|o_s, o_e)$ in the second line. □

C.2. Proof of Proposition 2

It is straightforward to prove the bi-equivariance of the diffused marginal using the bi-invariance of the integral measure (Haar measure) dg

$$\int_{SE(3)} d(\Delta g\, g) = \int_{SE(3)} dg = \int_{SE(3)} d(g\,\Delta g) \qquad \forall \Delta g \in SE(3) \tag{46}$$

where $dg = dR\, d\boldsymbol{p} = \frac{1}{8\pi^2} (\sin\beta)\, d\alpha\, d\beta\, d\gamma\, dx\, dy\, dz$ in the rotation-translation coordinate with the Euler angles α, β, γ and the frame origin x, y, z. See Chirikjian [12, 13, 14], Murray et al. [53] and Appendix A of Ryu et al. [61] for more details on the bi-invariant integral measure of $SE(3)$.

We first prove that the marginal is bi-equivariant if the kernel is bi-equivariant.

Proof of left equivariance.

$$
\begin{aligned}
P_t(g|o_s, o_e) &= \int_{SE(3)} dg_0 \; P_{t|0}(g|g_0, o_s, o_e) P_0(g_0|o_s, o_e) \\
&= \int_{SE(3)} dg_0 \; P_{t|0}(g|g_0, o_s, o_e) P_0(\Delta g \, g_0|\Delta g \cdot o_s, o_e) && (\because \text{Eq. (10)}) \\
&= \int_{SE(3)} dg_0 \; P_{t|0}(\Delta g \, g|\Delta g \, g_0, \Delta g \cdot o_s, o_e) P_0(\Delta g \, g_0|\Delta g \cdot o_s, o_e) && (\because \text{Eq. (15)}) \\
&= \int_{SE(3)} dg_0 \; P_{t|0}(\Delta g \, g|g_0, \Delta g \cdot o_s, o_e) P_0(g_0|\Delta g \cdot o_s, o_e) && (\because \text{Eq. (46)}, \; \Delta g \, g_0 \to g_0) \\
&= P_t(\Delta g \, g|\Delta g \cdot o_s, o_e)
\end{aligned}
$$

□

Proof of right equivariance.

$$
\begin{aligned}
P_t(g|o_s, o_e) &= \int_{SE(3)} dg_0 \; P_{t|0}(g|g_0, o_s, o_e) P_0(g_0|o_s, o_e) \\
&= \int_{SE(3)} dg_0 \; P_{t|0}(g|g_0, o_s, o_e) P_0(g_0 \, \Delta g^{-1}|o_s, \Delta g \cdot o_e) && (\because \text{Eq. (10)}) \\
&= \int_{SE(3)} dg_0 \; P_{t|0}(g \, \Delta g^{-1}|g_0 \, \Delta g^{-1}, o_s, \Delta g \cdot o_e) P_0(g_0 \, \Delta g^{-1}|o_s, \Delta g \cdot o_e) && (\because \text{Eq. (15)}) \\
&= \int_{SE(3)} dg_0 \; P_{t|0}(g \, \Delta g^{-1}|g_0, o_s, \Delta g \cdot o_e) P_0(g_0|o_s, \Delta g \cdot o_e) && (\because \text{Eq. (46)}, \; g_0 \, \Delta g^{-1} \to g_0) \\
&= P_t(g \, \Delta g^{-1}|o_s, \Delta g \cdot o_e)
\end{aligned}
$$

□

Similarly, it can be proven that the kernel must be bi-equivariant (up to measure zero) to guarantee the bi-equivariance of the diffused marginal for any arbitrary initial distribution $dP_0 = P_0 \, dg_0$.

Proof.

$$
\begin{aligned}
P_t(g|o_s, o_e) &= \int_{SE(3)} dg_0 \; P_{t|0}(g|g_0, o_s, o_e) P_0(g_0|o_s, o_e) \\
P_t(\Delta g \, g|\Delta g \cdot o_s, o_e) &= \int_{SE(3)} dg_0 \; P_{t|0}(\Delta g \, g|g_0, \Delta g \cdot o_s, o_e) P_0(g_0|\Delta g \cdot o_s, o_e) \\
&= \int_{SE(3)} dg_0 \; P_{t|0}(\Delta g \, g|g_0, \Delta g \cdot o_s, o_e) P_0(\Delta g^{-1} \, g_0|o_s, o_e) && (\because \text{Eq. (10)}) \\
&= \int_{SE(3)} dg_0 \; P_{t|0}(\Delta g \, g|\Delta g \, g_0, \Delta g \cdot o_s, o_e) P_0(g_0|o_s, o_e) && (\because \text{Eq. (46)}, \; g_0 \to \Delta g \, g_0)
\end{aligned}
$$

$$
\begin{aligned}
P_t(g \, \Delta g^{-1}|o_s, \Delta g \cdot o_e) &= \int_{SE(3)} dg_0 \; P_{t|0}(g \, \Delta g^{-1}|g_0, o_s, \Delta g \cdot o_e) P_0(g_0|o_s, \Delta g \cdot o_e) \\
&= \int_{SE(3)} dg_0 \; P_{t|0}(g \, \Delta g^{-1}|g_0, o_s, \Delta g \cdot o_e) P_0(g_0 \, \Delta g|o_s, o_e) && (\because \text{Eq. (10)}) \\
&= \int_{SE(3)} dg_0 \; P_{t|0}(g \, \Delta g^{-1}|g_0 \, \Delta g^{-1}, o_s, \Delta g \cdot o_e) P_0(g_0|o_s, o_e) && (\because \text{Eq. (46)}, \; g_0 \to g_0 \, \Delta g^{-1})
\end{aligned}
$$

$$\Rightarrow \int_{SE(3)} dg_0\ P_0(g_0|o_s,o_e) \times \left[P_{t|0}(g|g_0,o_s,o_e) - P_{t|0}(\Delta g\, g|\Delta g\, g_0, \Delta g\cdot o_s, o_e)\right] = 0$$
$$\int_{SE(3)} dg_0\ P_0(g_0|o_s,o_e) \times \left[P_{t|0}(g|g_0,o_s,o_e) - P_{t|0}(g\,\Delta g^{-1}|g_0\,\Delta g^{-1}, o_s, \Delta g\cdot o_e)\right] = 0$$

Therefore, for this equation to hold for any arbitrary bi-equivariant initial distribution $dP_0 = P_0\, dg_0$, the diffusion kernel must be bi-equivariant $\forall g, \Delta g \in SE(3)$

$$P_{t|0}(g|g_0,o_s,o_e) - P_{t|0}(\Delta g\, g|\Delta g\, g_0, \Delta g\cdot o_s, o_e) = 0 \tag{47}$$
$$P_{t|0}(g|g_0,o_s,o_e) - P_{t|0}(g\,\Delta g^{-1}|g_0\,\Delta g^{-1}, o_s, \Delta g\cdot o_e) = 0 \tag{48}$$
$$\Rightarrow\ P_{t|0}(g|g_0,o_s,o_e) = P_{t|0}(\Delta g\, g|\Delta g\, g_0, \Delta g\cdot o_s, o_e) = P_{t|0}(g\,\Delta g^{-1}|g_0\,\Delta g^{-1}, o_s, \Delta g\cdot o_e) \tag{49}$$

□

C.3. Non-existence of Bi-Invariant Diffusion Kernels on SE(3)

Note that any left invariant kernel $P_{t|0}(g|g_0)$ can be written in a univariate form $K_t(g_0^{-1}g)$.

$$P_{t|0}(\Delta g\, g|\Delta g\, g_0) = P_{t|0}(g|g_0) \quad \forall\, \Delta g, g \quad \Rightarrow \quad P_{t|0}(g|g_0) = P_{t|0}(g_0^{-1}g|I) \quad \forall\, g \tag{50}$$

The right invariance requires this kernel to satisfy $K_t(\Delta g\, g\, \Delta g^{-1}) = K_t(g)$, meaning that it is a *class function*, which does not exist for $L^2(SE(3))$ [12, 40].

C.4. Proof of Proposition 3

Proof. The right equivariance can be proved as follows.

$$\begin{aligned}
P_{t|0}(g|g_0,o_s,o_e) &= \int_{SE(3)} dg_{ed} P(g_{ed}|g_0^{-1}\cdot o_s, o_e) K_t(g_{ed}^{-1} g_0^{-1} g g_{ed}) \\
&= \int_{SE(3)} dg_{ed} P(\Delta g\, g_{ed}|(\Delta g\, g_0^{-1})\cdot o_s, \Delta g\cdot o_e) K_t(g_{ed}^{-1} g_0^{-1} g g_{ed}) && (\because \text{Eq. (18)}) \\
&= \int_{SE(3)} dg_{ed} P(g_{ed}|(\Delta g\, g_0^{-1})\cdot o_s, \Delta g\cdot o_e) K_t(g_{ed}^{-1} \left(g_0\,\Delta g^{-1}\right)^{-1} \left(g\,\Delta g^{-1}\right) g_{ed}) \\
& && (\because \text{invariance of integral } \textstyle\int dg_{ed} \text{ under } g_{ed} \to \Delta g^{-1}\, g_{ed}) \\
&= P_{t|0}(g\,\Delta g^{-1}|g_0\,\Delta g^{-1}, o_s, \Delta g\cdot o_e)
\end{aligned}$$

The left equivariance proof is straightforward using the following equations:

$$g_0^{-1}\, g = (\Delta g\, g_0)^{-1}\, (\Delta g\, g) \tag{51}$$
$$g_0^{-1}\cdot o_s = (\Delta g\, g_0)^{-1}\cdot (\Delta g\cdot o_s) \tag{52}$$

□

C.5. Proof of Proposition 4

Note that the Brownian diffusion kernel $\mathcal{B}_t(g)$ is right-invariant to rotation, that is,

$$\begin{aligned}
&\mathcal{B}_t\left((g_0\,\Delta R)^{-1}\,(g\,\Delta R)\right) = \mathcal{B}_t(g_0^{-1}\, g) \\
&\quad \Rightarrow \mathcal{B}_t(\Delta R^{-1}\, g\, \Delta R) = \mathcal{B}_t(g) \qquad \forall \Delta R \in SO(3)
\end{aligned} \tag{53}$$

where we abuse the notation to denote the action of a pure rotation ΔR on $g = (\boldsymbol{p}, R)$ as $\Delta R\, g = (\Delta R\, \boldsymbol{p}, \Delta R\, R)$ and $g\, \Delta R = (\boldsymbol{p}, R\, \Delta R)$. Eq. (53) holds because the Gaussian distribution in Eq. (5) is rotation-invariant and $\mathcal{IG}_{SO(3)}$ in Eq. (6) is a linear combination of *characters* of $SO(3)$, which are *class functions* due to the permutation invariance of trace operations (see Supp. B and C.3). Consider the following diffusion kernel with the equivariant origin selection mechanism in Eq. (20):

$$P_{t|0}(g|g_0,o_s,o_e) = \int_{\mathbb{R}^3} d\boldsymbol{p}_{ed} P\left(\boldsymbol{p}_{ed}|g_0^{-1}\cdot o_s, o_e\right) \mathcal{B}_t\left((g_0 \triangleleft \boldsymbol{p}_{ed})^{-1}\,(g \triangleleft \boldsymbol{p}_{ed})\right) \tag{54}$$

where $\triangleleft \boldsymbol{p}_{ed} : SE(3) \to SE(3)$ denotes the right action of a pure translation $\boldsymbol{p}_{ed} \in \mathbb{R}^3$ onto $g = (\boldsymbol{p}, R) \in SE(3)$ such that

$$g \triangleleft \boldsymbol{p}_{ed} = \begin{bmatrix} R & \boldsymbol{p} \\ \emptyset & 1 \end{bmatrix} \begin{bmatrix} I & \boldsymbol{p}_{ed} \\ \emptyset & 1 \end{bmatrix} = \begin{bmatrix} R & R\,\boldsymbol{p}_{ed} + \boldsymbol{p} \\ \emptyset & 1 \end{bmatrix} \tag{55}$$

Note that the following equation holds for all $g_1, g_2 \in SE(3)$ and $\boldsymbol{p}_{ed} \in \mathbb{R}^3$:

$$\begin{aligned} (g_1\, g_2) \triangleleft \boldsymbol{p}_{ed} &= \begin{bmatrix} R_1 & \boldsymbol{p}_1 \\ \emptyset & 1 \end{bmatrix} \begin{bmatrix} R_2 & \boldsymbol{p}_2 \\ \emptyset & 1 \end{bmatrix} \begin{bmatrix} I & \boldsymbol{p}_{ed} \\ \emptyset & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_1\, R_2 & R_1\,(R_2 \boldsymbol{p}_{ed} + \boldsymbol{p}_2) + \boldsymbol{p}_1 \\ \emptyset & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_1 & \boldsymbol{p}_1 \\ \emptyset & 1 \end{bmatrix} \begin{bmatrix} I & g_2\,\boldsymbol{p}_{ed} \\ \emptyset & 1 \end{bmatrix} \begin{bmatrix} R_2 & \mathbf{0} \\ \emptyset & 1 \end{bmatrix} \\ &= (g_1 \triangleleft (g_2\,\boldsymbol{p}_{ed}))\ \Delta R_2 \end{aligned} \tag{56}$$

The bi-equivariance of $P_{t|0}$ can be proved using Eq. (53) and Eq. (56).

Proof. The proof of left equivariance is straightforward as $g_0^{-1} \cdot o_s = (\Delta g\, g_0)^{-1} \cdot (\Delta g \cdot o_s)$. The proof of right equivariance is as follows:

$$\begin{aligned} &P_{t|0}(g\,\Delta g^{-1} | g_0\,\Delta g^{-1}, o_s, \Delta g \cdot o_e) \\ &= \int_{\mathbb{R}^3} d\boldsymbol{p}_{ed} P\left(\boldsymbol{p}_{ed} | (\Delta g\, g_0^{-1}) \cdot o_s, \Delta g \cdot o_e\right) \mathcal{B}_t \left(\left((g_0\,\Delta g^{-1}) \triangleleft \boldsymbol{p}_{ed}\right)^{-1} \left((g\,\Delta g^{-1}) \triangleleft \boldsymbol{p}_{ed}\right)\right) \\ &= \int_{\mathbb{R}^3} d\boldsymbol{p}_{ed} P\left(\Delta g^{-1}\,\boldsymbol{p}_{ed} | g_0^{-1} \cdot o_s, o_e\right) \mathcal{B}_t \left(\left((g_0\,\Delta g^{-1}) \triangleleft \boldsymbol{p}_{ed}\right)^{-1} \left((g\,\Delta g^{-1}) \triangleleft \boldsymbol{p}_{ed}\right)\right) && (\because \text{Eq. (20)}) \\ &= \int_{\mathbb{R}^3} d\boldsymbol{p}_{ed} P\left(\Delta g^{-1}\,\boldsymbol{p}_{ed} | g_0^{-1} \cdot o_s, o_e\right) \mathcal{B}_t \left(\Delta R\ \left(g_0 \triangleleft \left(\Delta g^{-1}\,\boldsymbol{p}_{ed}\right)\right)^{-1} \left(g \triangleleft \left(\Delta g^{-1}\,\boldsymbol{p}_{ed}\right)\right)\ \Delta R^{-1}\right) && (\because \text{Eq. (56)}) \\ &= \int_{\mathbb{R}^3} d\boldsymbol{p}_{ed} P\left(\Delta g^{-1}\,\boldsymbol{p}_{ed} | g_0^{-1} \cdot o_s, o_e\right) \mathcal{B}_t \left(\left(g_0 \triangleleft \left(\Delta g^{-1}\,\boldsymbol{p}_{ed}\right)\right)^{-1} \left(g \triangleleft \left(\Delta g^{-1}\,\boldsymbol{p}_{ed}\right)\right)\right) && (\because \text{Eq. (53)}) \\ &= \int_{\mathbb{R}^3} d\boldsymbol{p}_{ed} P\left(\boldsymbol{p}_{ed} | g_0^{-1} \cdot o_s, o_e\right) \mathcal{B}_t \left(\left(g_0 \triangleleft (\boldsymbol{p}_{ed})\right)^{-1} \left(g \triangleleft (\boldsymbol{p}_{ed})\right)\right) \\ & && (\because \text{invariance of Euclidean integral under roto-translation, } \boldsymbol{p}_{ed} \to \Delta g\,\boldsymbol{p}_{ed}) \\ &= P_{t|0}(g | g_0, o_s, o_e) \end{aligned}$$

□

In fact, any left-invariant kernel that is also right-invariant to rotation as in Eq. (53) can be used.

C.6. Derivation of Eq. (22)

We first show that $\boldsymbol{s}_t^*(g|o_s, o_e) = \mathbb{E}_{g_0, g_{ed}|g, O_s, O_e}\left[\nabla \log K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})\right]$ using a simple variational calculus.

Proof. Let $\delta \boldsymbol{s}_t(g|o_s, o_e)$ be a perturbation of the score model $\boldsymbol{s}_t(g|o_s, o_e)$. For the optimal score model $\boldsymbol{s}_t^*(g|o_s, o_e)$, any small perturbation would result in zero perturbation of the objective.

$$\begin{aligned} \boldsymbol{s}_t^*(g|o_s, o_e) &= \underset{\boldsymbol{s}_t(g|O_s, O_e)}{\arg\min}\ \mathcal{J}_t\left[\boldsymbol{s}_t(g|o_s, o_e)\right] \\ \Rightarrow\ \delta \mathcal{J}_t\left[\boldsymbol{s}_t^*(g|o_s, o_e)\right] &= 0 \qquad \forall\ \delta \boldsymbol{s}_t^*(g|o_s, o_e) \end{aligned} \tag{57}$$

The explicit form of the perturbation of the objective with regard to $\delta \boldsymbol{s}_t(g|o_s, o_e)$ is written as follows:

$$\begin{aligned} &\delta \mathcal{J}_t\left[\boldsymbol{s}_t(g|o_s, o_e)\right] = \delta\left(\mathbb{E}_{g, g_0, g_{ed}, O_s, O_e}\left[\frac{1}{2}\left\|\boldsymbol{s}_t(g|o_s, o_e) - \nabla \log K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})\right\|^2\right]\right) \\ &= \mathbb{E}_{g, O_s, O_e}\left[\delta \boldsymbol{s}_t(g|o_s, o_e) \cdot \left[\boldsymbol{s}_t(g|o_s, o_e) - \mathbb{E}_{g_0, g_{ed}|g, O_s, O_e}\left[\nabla \log K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})\right]\right]\right] \end{aligned} \tag{58}$$

Therefore, assuming $P_t(g|o_s, o_e) > 0 \;\; \forall g, o_s, o_e$, the optimal score model must be

$$\boldsymbol{s}_t^*(g|o_s, o_e) = \mathbb{E}_{g_0, g_{ed}|g, o_s, o_e}\left[\nabla \log K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})\right] \tag{59}$$

□

We now show that $\mathbb{E}_{g_0, g_{ed}|g, o_s, o_e}\left[\nabla \log K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})\right] = \nabla \log P_t(g|o_s, o_e)$.

Proof.

$$\begin{aligned}
&\mathbb{E}_{g_0, g_{ed}|g, o_s, o_e}\left[\nabla \log K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})\right] \\
&= \int dg_0 \int dg_{ed}\, P(g_0, g_{ed}|g, o_s, o_e; t) \frac{\nabla K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})}{K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})} \\
&= \int dg_0 \int dg_{ed} \left[P(g|g_0, g_{ed}, o_s, o_e; t) \frac{P(g_0, g_{ed}|o_s, o_e)}{P_t(g|o_s, o_e)}\right] \frac{\nabla K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})}{K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})} \\
&= \int dg_0 \int dg_{ed}\, \cancel{P(g|g_0, g_{ed}, o_s, o_e; t)} \frac{P(g_0, g_{ed}|o_s, o_e)}{P_t(g|o_s, o_e)} \frac{\nabla K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})}{\cancel{K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})}} \\
&\qquad\qquad (\because\; P(g|g_0, g_{ed}, o_s, o_e; t) = P(g|g_0, g_{ed}; t) = K_t(g_{ed}^{-1} g_0^{-1} g g_{ed})\,) \\
&= \frac{1}{P_t(g|o_s, o_e)} \int dg_0 \int dg_{ed}\, P(g_0, g_{ed}|o_s, o_e) \nabla K_t(g_{ed}^{-1} g_0^{-1} g g_{ed}) \\
&= \frac{1}{P_t(g|o_s, o_e)} \nabla \int dg_0 \int dg_{ed}\, P(g_0, g_{ed}|o_s, o_e) K_t(g_{ed}^{-1} g_0^{-1} g g_{ed}) \\
&= \frac{1}{P_t(g|o_s, o_e)} \nabla \int dg_0\, P_0(g_0|o_s, o_e) \int dg_{ed}\, P(g_{ed}|g_0^{-1} \cdot o_s, o_e) K_t(g_{ed}^{-1} g_0^{-1} g g_{ed}) \\
&= \frac{1}{P_t(g|o_s, o_e)} \nabla P_t(g|o_s, o_e) \qquad (\because \text{Eq. (17) and Eq. (14)}) \\
&= \frac{\nabla P_t(g|o_s, o_e)}{P_t(g|o_s, o_e)} = \nabla \log P_t(g|o_s, o_e)
\end{aligned}$$

□

Therefore, we prove that $\boldsymbol{s}_t^*(g|o_s, o_e) = \nabla \log P_t(g|o_s, o_e)$.

C.7. Proof of Proposition 5

For readers' convenience, we reproduce the bi-equivariance conditions for the score functions in Proposition 1 with explicit components.

$$\boldsymbol{s}(\Delta g\, g|\Delta g \cdot o_s, o_e) = \boldsymbol{s}(g|o_s, o_e) \tag{60}$$

$$\begin{aligned}
\boldsymbol{s}(g\, \Delta g^{-1}|o_s, \Delta g \cdot o_e) &= [\mathrm{Ad}_{\Delta g}]^{-T}\, \boldsymbol{s}(g|o_s, o_e) \\
&= \begin{bmatrix} \Delta R & \emptyset \\ [\Delta \boldsymbol{p}]^\wedge \Delta R & \Delta R \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_\nu(g|o_s, o_e) \\ \boldsymbol{s}_\omega(g|o_s, o_e) \end{bmatrix} \\
&= \Delta R\, \boldsymbol{s}_\nu(g|o_s, o_e) \oplus [\Delta R\, \boldsymbol{s}_\omega(g|o_s, o_e) + \Delta \boldsymbol{p} \wedge \Delta R\, \boldsymbol{s}_\nu(g|o_s, o_e)]
\end{aligned} \tag{61}$$

where we used the fact that the inverse transpose of the adjoint matrix is as follows [51, 53]:

$$[\mathrm{Ad}_{\Delta g}]^{-T} = \begin{bmatrix} \Delta R & \emptyset \\ [\Delta \boldsymbol{p}]^\wedge \Delta R & \Delta R \end{bmatrix} \tag{62}$$

We begin by proving the bi-equivariance of the linear (translational) score term

Proof. The left invariance of the linear score model is proved as

$$\begin{aligned}
\boldsymbol{s}_{\nu;t}(\Delta g\, g|\Delta g \cdot o_s, o_e) &= \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\boldsymbol{x}|o_e)\ \widetilde{\boldsymbol{s}}_{\nu;t}(\Delta g\, g, \boldsymbol{x}|\Delta g \cdot o_s, o_e) \\
&= \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\boldsymbol{x}|o_e)\ \widetilde{\boldsymbol{s}}_{\nu;t}(g, \boldsymbol{x}|o_s, o_e) && (\because \text{Eq. (27)}) \\
&= \boldsymbol{s}_{\nu;t}(g|o_s, o_e)
\end{aligned}$$

The right equivariance of the linear score model is proved as

$$\begin{aligned}
\boldsymbol{s}_{\nu;t}(g\,\Delta g^{-1}|o_s, \Delta g \cdot o_e) &= \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\boldsymbol{x}|\Delta g \cdot o_e)\ \widetilde{\boldsymbol{s}}_{\nu;t}(g\,\Delta g^{-1}, \boldsymbol{x}|o_s, \Delta g \cdot o_e) \\
&= \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\Delta g\,\boldsymbol{x}|\Delta g \cdot o_e)\ \widetilde{\boldsymbol{s}}_{\nu;t}(g\,\Delta g^{-1}, \Delta g\,\boldsymbol{x}|o_s, \Delta g \cdot o_e) \\
&\qquad (\because \text{invariance of Euclidean integral under roto-translation } \boldsymbol{x} \to \Delta g\,\boldsymbol{x}) \\
&= \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\boldsymbol{x}|o_e)\ \Delta R\,\widetilde{\boldsymbol{s}}_{\nu;t}(g, \boldsymbol{x}|o_s, o_e) && (\because \text{Eq. (26) and Eq. (28)}) \\
&= \Delta R \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\boldsymbol{x}|o_e)\ \widetilde{\boldsymbol{s}}_{\nu;t}(g, \boldsymbol{x}|o_s, o_e) \\
&= \Delta R\,\boldsymbol{s}_{\nu;t}(g|o_s, o_e)
\end{aligned}$$

□

Let the angular (rotational) score model be decomposed into the spin term $\boldsymbol{s}_{\text{spin};t}$ and the orbital term $\boldsymbol{s}_{\text{orbital};t}$ as in Eq. (25). The bi-equivariance of spin term in the angular (rotational) score model

$$\boldsymbol{s}_{\text{spin};t}(g|o_s, o_e) = \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\omega;t}(\boldsymbol{x}|o_e)\ \widetilde{\boldsymbol{s}}_{\omega;t}(g, \boldsymbol{x}|o_s, o_e) \tag{63}$$

$$\boldsymbol{s}_{\text{spin};t}(\Delta g\, g|\Delta g \cdot o_s, o_e) = \boldsymbol{s}_{\text{spin};t}(g|o_s, o_e) \tag{64}$$

$$\boldsymbol{s}_{\text{spin};t}(g\,\Delta g^{-1}|o_s, \Delta g \cdot o_e) = \Delta R\,\boldsymbol{s}_{\text{spin};t}(g|o_s, o_e) \tag{65}$$

can be proven in a similar fashion to the linear score model. It can be shown that the orbital term satisfies the following bi-equivariance condition

$$\boldsymbol{s}_{\text{orbital};t}(g|o_s, o_e) = \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\boldsymbol{x}|o_e)\ \boldsymbol{x} \wedge \widetilde{\boldsymbol{s}}_{\nu;t}(g, \boldsymbol{x}|o_s, o_e) \tag{66}$$

$$\boldsymbol{s}_{\text{orbital};t}(\Delta g\, g|\Delta g \cdot o_s, o_e) = \boldsymbol{s}_{\text{orbital};t}(g|o_s, o_e) \tag{67}$$

$$\boldsymbol{s}_{\text{orbital};t}(g\,\Delta g^{-1}|o_s, \Delta g \cdot o_e) = \Delta\boldsymbol{p} \wedge \Delta R\,\boldsymbol{s}_{\text{orbital};t}(g|o_s, o_e) \tag{68}$$

Proof. The left invariance is straightforward, as the linear score field $\widetilde{\boldsymbol{s}}_{\nu;t}$ is left-invariant as Eq. (27). The right equivariance can be proved as follows

$$\begin{aligned}
&\boldsymbol{s}_{\text{orbital};t}(g\,\Delta g^{-1}|o_s, \Delta g \cdot o_e) \\
&= \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\boldsymbol{x}|\Delta g \cdot o_e)\ \boldsymbol{x} \wedge \widetilde{\boldsymbol{s}}_{\nu;t}(g\,\Delta g^{-1}, \boldsymbol{x}|o_s, \Delta g \cdot o_e) \\
&= \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\Delta g^{-1}\,\boldsymbol{x}|o_e)\ \boldsymbol{x} \wedge \Delta R\,\widetilde{\boldsymbol{s}}_{\nu;t}(g, \Delta g^{-1}\,\boldsymbol{x}|o_s, o_e) && (\because \text{Eq. (26) and Eq. (28)}) \\
&= \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\boldsymbol{x}|o_e)\ (\Delta R\,\boldsymbol{x} + \Delta\boldsymbol{p}) \wedge \Delta R\,\widetilde{\boldsymbol{s}}_{\nu;t}(g, \boldsymbol{x}|o_s, o_e) \\
&\qquad (\because \text{invariance of Euclidean integral under roto-translation } \boldsymbol{x} \to \Delta g\,\boldsymbol{x} = \Delta R\,\boldsymbol{x} + \Delta\boldsymbol{p}) \\
&= \Delta R \left[\int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\boldsymbol{x}|o_e)\ \boldsymbol{x} \wedge \widetilde{\boldsymbol{s}}_{\nu;t}(g, \boldsymbol{x}|o_s, o_e)\right] + \Delta\boldsymbol{p} \wedge \Delta R \int_{\mathbb{R}^3} d^3\boldsymbol{x}\ \rho_{\nu;t}(\boldsymbol{x}|o_e)\ \widetilde{\boldsymbol{s}}_{\nu;t}(g, \boldsymbol{x}|o_s, o_e) \\
&\qquad (\because\ R\,\boldsymbol{x} \wedge R\,\boldsymbol{y} = R\,(\boldsymbol{x} \wedge \boldsymbol{y})\ \ \forall\,\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3) \\
&= \Delta R\,\boldsymbol{s}_{\text{orbital};t}(g|o_s, o_e) + \Delta\boldsymbol{p} \wedge \Delta R\,\boldsymbol{s}_{\nu;t}(g|o_s, o_e)
\end{aligned}$$

□

As a result, the angular (rotational) score model

$$\boldsymbol{s}_{\omega;t}(g|\mathcal{O}_s,\mathcal{O}_e) = \boldsymbol{s}_{\text{orbital};t}(g|\mathcal{O}_s,\mathcal{O}_e) + \boldsymbol{s}_{\text{spin};t}(g|\mathcal{O}_s,\mathcal{O}_e) \tag{69}$$

satisfies the following bi-equivariance

$$\boldsymbol{s}_{\omega;t}(\Delta g\, g|\Delta g\cdot\mathcal{O}_s,\mathcal{O}_e) = \boldsymbol{s}_{\omega;t}(g|\mathcal{O}_s,\mathcal{O}_e) \tag{70}$$

$$\begin{aligned}\boldsymbol{s}_{\omega;t}(g\,\Delta g^{-1}|\mathcal{O}_s,\Delta g\cdot\mathcal{O}_e) =& \Delta R\,[\boldsymbol{s}_{\text{orbital};t}(g|\mathcal{O}_s,\mathcal{O}_e) + \boldsymbol{s}_{\text{spin};t}(g|\mathcal{O}_s,\mathcal{O}_e)] + \Delta\boldsymbol{p}\wedge\Delta R\,\boldsymbol{s}_{\nu;t}(g|\mathcal{O}_s,\mathcal{O}_e)\\ =& \Delta R\,\boldsymbol{s}_{\omega;t}(g|\mathcal{O}_s,\mathcal{O}_e) + \Delta\boldsymbol{p}\wedge\Delta R\,\boldsymbol{s}_{\nu;t}(g|\mathcal{O}_s,\mathcal{O}_e)\end{aligned} \tag{71}$$

Hence, we have proven Proposition 5 that the score model in Eq. (23) is bi-equivariant, satisfying Eq. (60) and Eq. (61).

C.8. Proof of Proposition 6

Proof.

$$\begin{aligned}\widetilde{\boldsymbol{s}}_{\square;t}(\Delta g\, g,\boldsymbol{x}|\Delta g\cdot\mathcal{O}_s,\mathcal{O}_e) &= \boldsymbol{\psi}_{\square;t}(\boldsymbol{x}|\mathcal{O}_e)\ \otimes^{(\to 1)}_{\square;t}\ \mathbf{D}(R^{-1}\Delta R^{-1})\,\boldsymbol{\varphi}_{\square;t}(\Delta g\, g\,\boldsymbol{x}|\Delta g\cdot\mathcal{O}_s)\\ &= \boldsymbol{\psi}_{\square;t}(\boldsymbol{x}|\mathcal{O}_e)\ \otimes^{(\to 1)}_{\square;t}\ \mathbf{D}(R^{-1}\Delta R^{-1})\mathbf{D}(\Delta R)\,\boldsymbol{\varphi}_{\square;t}(g\,\boldsymbol{x}|\mathcal{O}_s) && (\because \text{Eq. (3)})\\ &= \boldsymbol{\psi}_{\square;t}(\boldsymbol{x}|\mathcal{O}_e)\ \otimes^{(\to 1)}_{\square;t}\ \mathbf{D}(R^{-1}\cancel{\Delta R^{-1}\Delta R})\,\boldsymbol{\varphi}_{\square;t}(g\,\boldsymbol{x}|\mathcal{O}_s) && (\because \text{Eq. (1)})\\ &= \widetilde{\boldsymbol{s}}_{\square;t}(g,\boldsymbol{x}|\mathcal{O}_s,\mathcal{O}_e)\end{aligned}$$

$$\begin{aligned}&\widetilde{\boldsymbol{s}}_{\square;t}(g\,\Delta g^{-1},\Delta g\,\boldsymbol{x}|\mathcal{O}_s,\Delta g\cdot\mathcal{O}_e)\\ &= \boldsymbol{\psi}_{\square;t}(\Delta g\,\boldsymbol{x}|\Delta g\cdot\mathcal{O}_e)\ \otimes^{(\to 1)}_{\square;t}\ \mathbf{D}(\Delta R\,R^{-1})\,\boldsymbol{\varphi}_{\square;t}(g\,\cancel{\Delta g^{-1}\Delta g}\,\boldsymbol{x}|\mathcal{O}_s)\\ &= \mathbf{D}(\Delta R)\,\boldsymbol{\psi}_{\square;t}(\boldsymbol{x}|\mathcal{O}_e)\ \otimes^{(\to 1)}_{\square;t}\ \mathbf{D}(\Delta R\,R^{-1})\,\boldsymbol{\varphi}_{\square;t}(g\,\boldsymbol{x}|\mathcal{O}_s) && (\because \text{Eq. (3)})\\ &= \mathbf{D}(\Delta R)\,\boldsymbol{\psi}_{\square;t}(\boldsymbol{x}|\mathcal{O}_e)\ \otimes^{(\to 1)}_{\square;t}\ \mathbf{D}(\Delta R)\,\mathbf{D}(R^{-1})\,\boldsymbol{\varphi}_{\square;t}(g\,\boldsymbol{x}|\mathcal{O}_s) && (\because \text{Eq. (1)})\\ &= \mathbf{D}_1(\Delta R)\left[\boldsymbol{\psi}_{\square;t}(\boldsymbol{x}|\mathcal{O}_e)\ \otimes^{(\to 1)}_{\square;t}\ \mathbf{D}(R^{-1})\,\boldsymbol{\varphi}_{\square;t}(g\,\boldsymbol{x}|\mathcal{O}_s)\right] && (\because\ [\mathbf{D}(R)\boldsymbol{v}]\otimes^{(\to l)}[\mathbf{D}(R)\boldsymbol{w}] = \mathbf{D}_l(R)\left[\boldsymbol{v}\otimes^{(\to l)}\boldsymbol{w}\right])\\ &= \Delta R\,\widetilde{\boldsymbol{s}}_{\square;t}(g,\boldsymbol{x}|\mathcal{O}_s,\mathcal{O}_e)\end{aligned}$$

where in the last line we assume that the degree-1 Wigner D-matrix $\mathbf{D}_1(\cdot)$ is in the real basis with $x-y-z$ axis ordering. Note that the last line only holds in this specific choice of basis. Therefore, the type-1 or higher descriptors of the two EDFs must be defined in this basis. □

D. Implementation Details

D.1. Score Field Model Details

We assume that the output of the score field model in Eq. (29) is a dimensionless quantity. Therefore, we obtain the dimensionful score by taking

$$\widetilde{\boldsymbol{s}}_{\nu;t} \to \frac{1}{L\sqrt{t}}\widetilde{\boldsymbol{s}}_{\nu;t}, \qquad \widetilde{\boldsymbol{s}}_{\omega;t} \to \frac{1}{\sqrt{t}}\widetilde{\boldsymbol{s}}_{\omega;t}$$

where L is the characteristic length scale unit. The reason for dividing $1/\sqrt{t}$ is because the norm of the target score tend to scale with $O(1/\sqrt{t})$. Likewise, we divide the linear score field by L because score field is a gradient and thus scales reciprocally to the characteristic length scale.

For computational efficiency, we use identical EDFs for $\square = \omega, \nu$ in Eq. (29). In addition, we remove the time dependence of the grasp EDF $\boldsymbol{\psi}_t(\boldsymbol{x}|\mathcal{O}_e)$ so that its field value is computed only once at the beginning of the denoising process. In

conclusion, our actual implementations of Eq. (32) and Eq. (33) are as follows:

$$\boldsymbol{s}_{\nu;t}(g|\mathcal{O}_s,\mathcal{O}_e) = \frac{1}{L\sqrt{t}} \sum_{\boldsymbol{q}\in Q(O_e)} w(\boldsymbol{q}|\mathcal{O}_e) \left[\boldsymbol{\psi}(\boldsymbol{q}|\mathcal{O}_e) \otimes_{\nu;t}^{(\to 1)} \mathbf{D}(R^{-1})\,\boldsymbol{\varphi}_t(g\,\boldsymbol{q}|\mathcal{O}_s)\right] \tag{72}$$

$$\begin{aligned}\boldsymbol{s}_{\omega;t}(g|\mathcal{O}_s,\mathcal{O}_e) = &\frac{1}{\sqrt{t}} \sum_{\boldsymbol{q}\in Q(O_e)} w(\boldsymbol{q}|\mathcal{O}_e)\,\frac{\boldsymbol{q}}{L} \wedge \left[\boldsymbol{\psi}(\boldsymbol{q}|\mathcal{O}_e) \otimes_{\nu;t}^{(\to 1)} \mathbf{D}(R^{-1})\,\boldsymbol{\varphi}_t(g\,\boldsymbol{q}|\mathcal{O}_s)\right] \\ &+ \frac{1}{\sqrt{t}} \sum_{\boldsymbol{q}\in Q(O_e)} w(\boldsymbol{q}|\mathcal{O}_e) \left[\boldsymbol{\psi}(\boldsymbol{q}|\mathcal{O}_e) \otimes_{\omega;t}^{(\to 1)} \mathbf{D}(R^{-1})\,\boldsymbol{\varphi}_t(g\,\boldsymbol{q}|\mathcal{O}_s)\right]\end{aligned} \tag{73}$$

D.2. Sampling with Annealed Langevin Dynamics

It is known to be difficult and unstable to train and sample with the score function for a sparse distribution [38, 71]. To address this issue, *Annealed Langevin Markov Chain Monte Carlo* [71] leverages the score of the diffused marginal P_t instead of P_0. A diffused marginal $P_t(g)$ for a diffusion kernel $P_{t|0}(g|g_0)$ is defined on the $SE(3)$ manifold as

$$P_t(g) = \int_{SE(3)} dg_0 P_{t|0}(g|g_0) P_0(g_0). \tag{74}$$

We utilize the trained score function $s_t(g) = \nabla \log P_t(g)$ for the annealed Langevin MCMC on $SE(3)$ [75] as

$$g_{\tau+d\tau} = g_\tau \exp\left[\frac{1}{2}\boldsymbol{s}_{t(\tau)}(g_\tau|\mathcal{O}_s,\mathcal{O}_e)d\tau + dW\right]. \tag{75}$$

where $t(\tau)$ is the diffusion time scheduling, which is gradually annealed to zero as $\tau \to \infty$, such that $t(\tau = \infty) = 0$. This process will converge to $P_0(g)$ regardless of the initial distribution if it is annealed sufficiently slowly and $\lim_{t\to 0} P_t = P_0$. This SDE can be discretized using the forward Euler-Maruyama method such that

$$g_{n+1} = g_n \exp\left[\frac{1}{2}\boldsymbol{s}_{t[n]}(g_n|\mathcal{O}_s,\mathcal{O}_e)\alpha[n] + \sqrt{\alpha[n]}\boldsymbol{z}_n\right], \quad \boldsymbol{z}_n \sim \mathcal{N}(\mathbf{0}, I) \tag{76}$$

where $t[n]$ and $\alpha[n]$ are respectively the diffusion time and Langevin step size, both of which are scheduled according to the step count n. A commonly used scheduling scheme is taking $\alpha[n] \propto t[n]$ with either a linear or log-linear $t[n]$ schedule [30, 71, 75]. However, the convergence is very slow with this scheduling. Therefore, we use $\alpha[n] \propto t[n]^{k_1}$ schedule with a hyperparameter $k_1 < 1$. To suppress the instability caused by large step sizes when t is small, we also gradually lower the *temperature*[3] of the process. This can be done by using $\sqrt{\alpha[n]T[n]}\boldsymbol{z}_n$ instead of $\sqrt{\alpha[n]}\boldsymbol{z}_n$ for the noise term with the temperature schedule $T[n] = t[n]^{k_2}$, where $k_2 \geq 0$ is another hyperparameter. Intuitively, this makes the sampling process to smoothly transition into a simple gradient descent optimization as $t[n] \to 0$, and hence $T[n] \to 0$. We empirically found that this strategy significantly improves the convergence time without compromising the accuracy and diversity of the sampled poses. The resulting sampling algorithm with a small number ϵ is

$$g_{n+1} = g_n \exp\left[\frac{\epsilon}{2}\boldsymbol{s}_{t[n]}(g_n|\mathcal{O}_s,\mathcal{O}_e)t[n]^{k_1} + \sqrt{\epsilon}\,t[n]^{\frac{k_1+k_2}{2}}\boldsymbol{z}_n\right], \quad \boldsymbol{z}_n \sim \mathcal{N}(\mathbf{0}, I) \tag{77}$$

We use $k_1 = 0.5$ and $k_2 = 1.0$ for the step size and temperature scheduling. For the diffusion time $t[n]$, we use piecewise linear scheduling. For example, we linearly schedule the diffusion time for $t = 1$ to $t = 0.1$ and then with $t = 0.1$ to $t = 0.01$. Similar to diffusion-based image generation models, we separate a low-resolution model and high-resolution model instead of using a single model. We use the low-resolution model for higher t and the high-resolution model for lower t. Similar to Ryu et al. [61], we solve Eq. (77) in the quaternion-translation parameterization of $SE(3)$ instead of performing the actual exponential mapping in Eq. (77).

D.3. Architecture details

See Fig. 7 for the illustration of each module used in Fig. 2.

[3]This temperature annealing should not be confused with that of the *'annealed'* Langevin MCMC in which the diffusion time t is decreased.

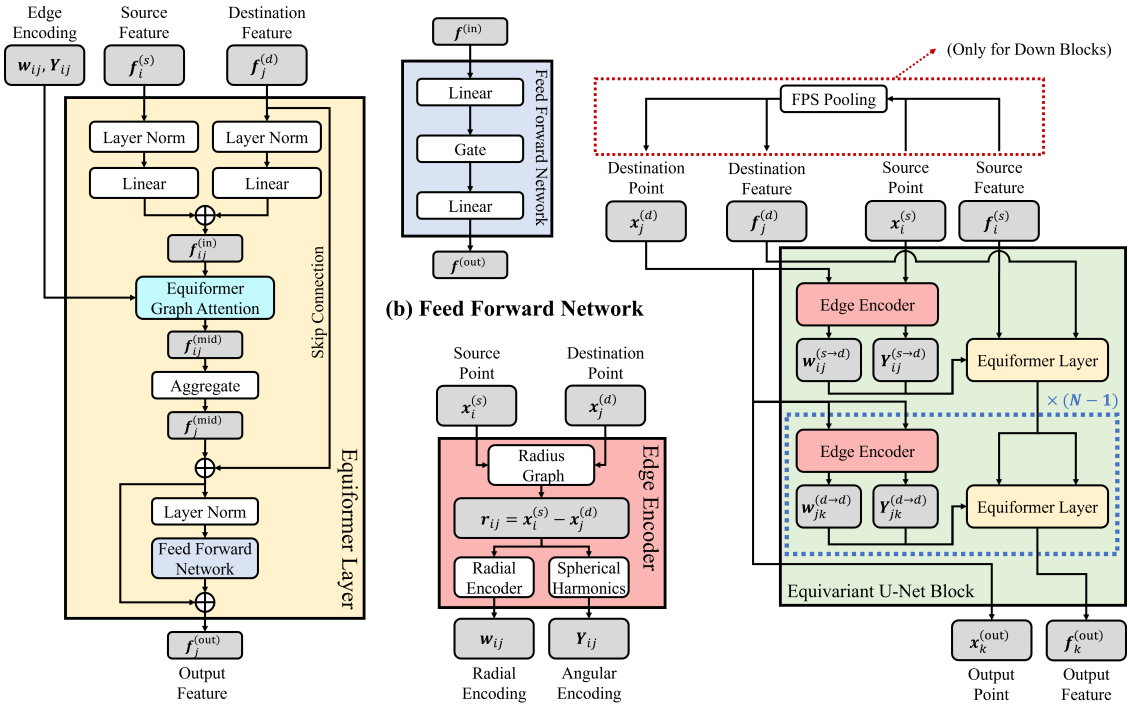

Figure 7. **Overview of Modules Used in Multiscale EDF. (a)** We employ Equiformer [45] to achieve $SE(3)$-equivariance in our model. **(b)** We use an equivariant feed forward network with gate activation from Equiformer. **(c)** We use radius graph to construct graph from points. Graph edge length and orientation are respectively encoded by a radial encoder and spherical harmonics [26, 45, 74]. **(d)** Multiple equiformer layers are stacked and form the equivariant U-Net Block. FPS pooling is used in downward blocks to obtain coarse-grained destination points from source points in lower scale-space.

D.4. Diffusion Frame Selection Mechanism

In this section, we provide further details on the diffusion frame/origin selection mechanism.

Necessity of Diffusion Frame/Origin Selection Mechanism. We first discuss why a diffusion frame/origin selection mechanism is necessary for our diffusion model on the $SE(3)$ manifold. For simplicity, we confine our argument only to the diffusion origin selection mechanism as Proposition 4 suggests.

In Sec. 3.3, we introduced the concept of diffusion frame/origin selection mechanism to achieve bi-equivariance in the diffusion process. However, the diffusion frame/origin selection has further implication, even for non-equivariant diffusion models on the $SE(3)$ manifold. As illustrated in Fig. 8, an arbitrarily small rotational perturbation may result in an arbitrarily large orbital displacement near the critical region depending on the choice of the origin, leading to an unstable diffusion and denoising process. This is in contrast to typical Euclidean diffusion models because vector addition is a commutative operation, and hence origin fixing has no effect. Therefore, a proper diffusion process for our problem must include a diffusion origin selection procedure to minimize the orbital effect of rotation near critical regions.

A natural selection of the diffusion origin for manipulation tasks is the origin of the end-effector frame itself. However, this origin selection is not equivariant to the grasped object, making our diffusion kernel only left-equivariant and not right-equivariant. Another natural diffusion origin is the centroid of the point cloud, which was utilized by Yim et al. [84] and Corso et al. [17] for protein docking problems. Indeed, this is a special case of an equivariant origin selection mechanism that satisfies Eq. (19). However, as pointed out by Ryu et al. [61] and Kim et al. [37], centroids are often dominated by the global geometry rather than the critical sub-geometry of the target objects. Please recall that this is why R-NDFs suffer without object segmentation. While the protein-ligand interaction problem in Yim et al. [84] and Corso et al. [17] has additional torsional degrees of freedom to debias this centroid artifact, it won't translate to our problem since the points in o_e are only actuated by the end-effector pose g.

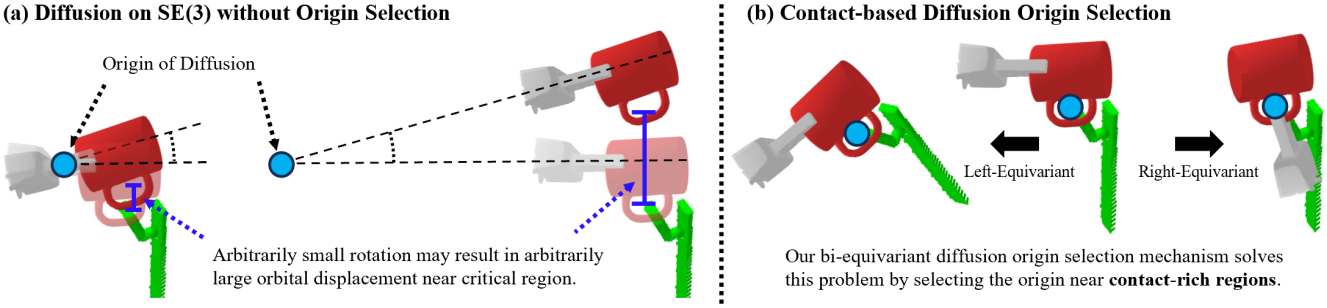

Figure 8. **Necessity of Diffusion Origin Selection Mechanism. (a)** A small rotational diffusion may result in arbitrarily large orbital displacement near the critical region depending on the diffusion origin. **(b)** We employ a contact-based diffusion origin selection mechanism. This not only allows bi-equivariant diffusion process but also stabilizes learning by minimizing the orbital impact of the rigid body rotation near the critical regions.

Equivariant Diffusion Origin Selection Mechanism with Contact Heuristics. An important quality of a good diffusion origin selection mechanism is that the selected origin should not be too far away from the critical contact-rich region. As illustrated in Fig. 8, even a small rotational diffusion may take the critical region of the grasped object (the handle of the mug) far away from the placement target (the tip of the hanger), making training unstable. Although this problem can be resolved by reducing the rotational noise scale of the diffusion process, it requires meticulous task-specific hyperparameter tuning. Furthermore, as can be seen in Eq. (61), the rotational score consists of the pure rotational term and the orbital term. By studying the orbital term, one may notice that this term is non-dimensionalized by the product of the displacement term $\Delta \boldsymbol{p}$, which is proportional to the length unit, and the translational score $\boldsymbol{s}_\nu$, which is reciprocal to the length unit. Although these two dimensionful quantities neatly cancel out each other's unit, this structure inevitably increases the variance of the score estimation when the displacement term $\Delta \boldsymbol{p}$ is too large. For instance, a small translational score term in the reference frame of the critical region may induce a large rotational score term in the end-effector frame if the displacement $\Delta \boldsymbol{p}$ between these two frames is large. This is natural because a small rotation in the end-effector frame can dramatically change the probability of the pose if $\Delta \boldsymbol{p}$ is large. Therefore, it is always optimal to work in a diffusion origin near the critical region, such that $\Delta \boldsymbol{p}$ is kept minimal. This is the reason why we propose a contact-based diffusion origin selection mechanism in Eq. (30), which selects the origin near the important contact-rich sub-geometries.

We find that this origin selection mechanism stabilizes training by enabling Diffusion-EDFs to correctly identify important contact rich sub-geometries from the grasp observation o_e. This can be verified by visualizing the strength of the query weight field. Fig. 9 illustrates the query points in colors according to their query weights. Query points with high weights are represented in cyan and those with near-zero weights are in black. As can be seen in the figure, the query weight field of the trained Diffusion-EDFs successfully assign high weight to the mug's handle, which is the most significant sub-geometry when placing it on a hanger.

E. Experiment Details

E.1. Simulation Experiment Details

In this section, we provide further details on the simulated benchmark experiments in Sec. 5.

E.1.1 Simulation Environment

Evaluations are performed in a simulated enviroment using SAPIEN [83] with nine ceiling-mounted depth cameras. We assume a perfect observation to remove the influence of point cloud processing pipelines, which is orthogonal to our research. We also remove the impact of robot's kinematic constraints by using a floating gripper-only robot instead of simulating the full robot. In addition, we turn off the collision between the environment and allow the robot to teleport to the pre-pick/place pose in order to get rid of failures related to motion planning. We evaluate the success of pick or place by turning off the collision between the environment (including the table) and the target object to manipulate, and measuring the object's z-axis position. If the object is not firmly grasped by the gripper or is not placed on the intended placement target, the object will fall after removing the environmental collision. Therefore, we measure the z-axis position to automatically assess whether the object has not fallen, meaning that the manipulation has succeeded.

(a) Query points of a real-world mug observation.

(b) Query points of a real-world bottle observation.

Figure 9. **Learned Query Points**. The figure depicts the point clouds of a real mug and bottle with their query points visualized in colors according to their weights. The query points with the highest weight values are illustrated in cyan. The query weight field of the trained Diffusion-EDFs assigns high weight to **(a)** the mug's handle, which is the most significant sub-geometry when placing it on a hanger, and **(b)** the bottom of the bottle, which is the most significant sub-geometry when placing it on a shelf.

E.1.2 Method Details

For each task, we train the models using ten human-generated demonstrations, in which five object instances in only upright poses are used. In other words, each of the five object instances is demonstrated for two different pick/place poses. In the training data set, we do not use distracting objects. We used a custom-built web-based GUI to collect human demonstrations.

Diffusion-EDFs. We only use ten human demonstrations to train Diffusion-EDFs in a fully end-to-end manner. No additional prior knowledge such as pre-training, object segmentation, pose estimation or data augmentation is used for Diffusion-EDFs. For preprocessing, we use simple voxel downsampling to reduce the number of points.

R-NDFs. For R-NDFs, we use the pre-trained weights from the original implementation of Simeonov et al. [68]. These weights were trained with a self-supervised learning method that relies on massive amount (150 gigabytes) object geometry that are specific to the target object categories (mug, bowl, bottle; 50 gigabytes for each). Although we do not use bowls in our experiment, we still use the weights trained from all three object categories, which achieve better performance than weights trained from only a single object category [67, 68]. Still, we observe that R-NDFs fail to place the mug on our mug hanger. We presume that this is due to the discrepancy of the hanger's shape in our experiment and the ones used for pre-training, which were procedurally generated [68]. Therefore, we do R-NDFs an additional favor of using the pre-trained hanger instances instead of our hanger for the evaluation. Lastly, we also tried to naively pre-train the NDFs using the reconstructed meshes from the point clouds in our ten task demonstrations, but resulted in suboptimal performance (less than 5% success rate). These attempts show the importance of the end-to-end trainability of EDFs [61] and Diffusion-EDFs. R-NDFs cannot be used for uncommon object categories, as they require immense amount of category-specific data for pre-training. Procedural generation has also turned out to be unable to resolve this problem because it cannot cover all variations in the category, which was evident in the case of the mug hanger mentioned above.

We also evaluate R-NDFs both with and without object segmentation. It should be noted that the ability to infer without object segmentation is important not only because of its convenience. As we have demonstrated in our real hardware experiments in Sec. 5, it allows the model to understand *scene-level contexts* beyond a single target object. The experimental results in Tab. 1 clearly show that R-NDFs are unable to make inference without object segmentation. As pointed out by Ryu et al. [61], we presume this is because of the violation of locality in R-NDFs, such as centroid subtraction.

SE(3)-Diffusion Fields. In contrast to R-NDFs, we train $SE(3)$-Diffusion Fields [75] using only the ten demonstrations as Diffusion-EDFs. Following Urain et al. [75], we jointly train the model to match both the signed distance function and the score function. We specifically use the *PoiNt-SE(3)-DiF* variant in the original paper [75]. Although this model utilizes $SO(3)$-equivariant point cloud encoder based on VN-PointNet [19], the overall architecture is not equivariant. Therefore, we use $SO(3)$ rotational data augmentation to complement the lack of equivariance.

Similar to R-NDFs, we evaluate $SE(3)$-Diffusion Fields both with and without object segmentation. With object segmentation, $SE(3)$-Diffusion Fields could learn to pick up the target object, although the success rates are much lower than Diffusion-EDFs. Without object segmentation, they achieve success rates lower than 15% for all scenarios.

E.2. Real Hardware Experiment Details

E.2.1 Experimental Setup

We use a Franka Emika Panda robot arm with two Intel RealSense D415 RGB-D cameras. The first camera is attached to the wrist of the robot. The robot moves around the workspace to observe RGB-D images of the scene from multiple viewpoints. We employ RTAB-Map [41], a 3D SLAM technique, to convert these observations into a point cloud of the scene. Rather than relying on visual odometry, we take advantage of the forward kinematics solution from the robot's joint encoders, which is more precise. Although we use 3D SLAM-based approach in our experiments, this procedure can be skipped if multiple well-calibrated external cameras are available. The second camera is installed on the table to observe the point cloud of the robot's gripper. This external camera is calibrated to the ArUco marker [28] frame attached to the robot's end-effector. All the point clouds are post-processed using Open3D [88], in which we remove statistical outliers and apply voxel filtering. We also apply hue and lightness augmentation for the training data to obtain robustness under light condition changes.

In our experimental procedure, the robot first moves along a predefined trajectory to observe the scene. RTAB-Map is used to convert these observations into the point cloud of the scene in real time. Diffusion-EDFs take this point cloud to generate the end-effector poses to pick the target object. After picking the object, the robot moves to the predefined grasp observation pose. The robot then rotates its grasped object by 360°, and the external camera observes it. These observations are then registered into the grasp point cloud. For the scene point cloud, we use the same one that we used to infer the pick pose. With these two point clouds, Diffusion-EDFs infer the end-effector poses to place the grasped object onto the placement target. For the collection of human demonstrations, we follow a procedure similar to that in the aforementioned inference pipeline. The only difference is that the target pose demonstration is manually provided by a human instead of Diffusion-EDFs.

E.2.2 System Engineering

Motion Primitives. While it is theoretically possible to generate a collision-free motion plan for any reachable goal pose, it is challenging in reality due to the imprecise nature of point cloud observations. Therefore, determining how to approach the target pose is also an important problem. As we focus only on the problem of inferring the target pose itself in this work, we simply assume that we already have task-specific motion primitives to approach the generated goal pose. In all three real-world tasks, we use a simple motion primitive of picking along the end-effector's z-axis direction (the direction in which the gripper is pointing), and placing the target object in the top-down direction. The robot first moves to the pre-pick/place pose by following the collision-free trajectory found by an off-the-shelf motion planner. The motion primitives are then used to approach the generated target pick/place pose from the previous pre-pick/place pose. After successful picking or placing, we initiate post-pick/place primitives. We simply lift up the end-effector for the post-pick primitive. For the post-place primitive, we retract the end-effector towards the opposite direction that was taken in the pre-pick manuever. We use MoveIt [16] for motion planning and use the TOPP-RA [57] algorithm to time-parameterize our waypoint-based motion primitives.

Although we use predefined motion primitives, not every problem can be solved in this way. Therefore, more general approach should also encompass learning not only the target pose but also the approach direction. We expect that our score model in Eq. (23) can be used for this purpose with slight modifications. The approach direction can be represented as the displacement between the pre-pick/place pose and the target pose. This displacement can be effectively expressed as an $\mathfrak{se}(3)$ Lie algebra vector. Therefore, our score model can be modified to equivariantly infer this Lie algebra vector that represents the approach direction. We leave this research for future studies.

Energy-based Critic. Due to the collision and kinematic constraint of the robot, not every pose generated by Diffusion-EDFs are feasible. Although we ignored this problem in our simulation experiment, this problem must be considered in real robot applications. Therefore, similar to Urain et al. [75] and Ryu et al. [61], we generate multiple samples in parallel and reject infeasible poses one by one until a reachable pose is found.

However, it is difficult to ensure convergence for every generated sample as we use a limited number of Langevin steps to achieve reasonable inference time (5$\sim$17 seconds). The number of unconverged samples tend to be larger in our real-world experiment with noisy observations than in the simulated ones with perfect observations. Furthermore, rejecting infeasible poses often leads to the elimination of correct poses and the selection of unconverged wrong poses. Urain et al. [75] and Ryu et al. [61] circumvented this problem by sorting the generated samples according to the learned energy function, which

evaluates the quality of the generated poses. In contrast to these works, however, our method does not have an explicit scalar function that can be utilized.

Therefore, we train an auxiliary energy function to sort the generated poses according to their quality. We first modify the bi-equivariant energy function of Ryu et al. [61] to allow diffusion time conditioning. We then take the Lie derivatives to obtain the energy-based score model similar to Urain et al. [75]. This energy-based score model is trained using the loss function in Eq. (21) with proper non-dimensionalization. Although this score-matching model is far less accurate than our original model in Eq. (23) due to the inflexible nature of energy-based diffusion models, the trained energy function is sufficient to distinguish between unconverged samples and converged samples.

With the learned energy function, we first sort the generated samples according to their energy value. If the energy function is well trained, lower-energy samples should be better than higher-energy samples. However, in contrast to the MCMC-based training of Ryu et al. [61], our diffusion-based energy function training does not have a contrastive mechanism to penalize the model for assigning low energy to outlier poses. Therefore, our energy function often assigns too low energy values to outlier poses, although the training is much faster. Nevertheless, we find that simply rejecting too-low-energy outliers effectively solves this problem. Therefore, we remove the first few samples from the sorted list and start from samples with moderately low energy. We then try motion planning for each sample until a feasible pose is found. This strategy drastically improves the success rate of pick-and-place tasks in our real-world tasks.

E.2.3 Experimental Results Details

Note that it is difficult to precisely measure the performance of Diffusion-EDFs for real-world tasks as the success rate is determined not only by the inference quality but also the quality of observation, localization, and motion planning. For instance, noisy observation and localization cause success rates to drop for subtasks that require high precision, such as mug placement and bottle picking, even though Diffusion-EDFs accurately generated correct target poses. Challenges associated with motion planning can also reduce the success rate, particularly for subtasks that require difficult 6-DoF manipulation, such as mug placement. We achieve over 90% success rate for all subtasks except the mug placement and bottle picking. For these two tasks, the success rates are roughly around 80%. The majority of the errors in these tasks were caused by a slight lack of accuracy in the position that was less than a centimeter. Note that these real hardware success rates may largely differ across systems, depending on the quality of observation, calibration, motion planning and control pipelines, which are orthogonal to our research. The video of the real robot manipulation experiments can be found in our project website (https://sites.google.com/view/diffusion-edfs). In the video, our robot performs 5 to 6 pick-and-place operations in one take without failure, showcasing that Diffusion-EDFs can solve all three real-world tasks with high success rates.

For more reproducible results, we also provide example input data and codes[4] that we used to generate end-effector poses for the three real-world tasks with Diffusion-EDFs. These supplementary materials can provide an idea of Diffusion-EDFs' pure inference performance for noisy real-world observations without the complications related to motion planning and localization. The samples generated by Diffusion-EDF for the mug-on-hanger and bowls-on-dishes tasks are illustrated in Figs. 10 and 11, respectively. The samples generated by Diffusion-EDF for the bottles-on-shelf task are illustrated in Figs. 12 and 13. Diffusion-EDFs combined with the energy-based critic in Sec. E.2.2 can successfully infer appropriate poses for all these tasks in more than 90% of the cases, although it is important to note that this success rate is subject to human evaluation and may vary based on the individual's criteria.

For mugs and bottles, it takes 5$\sim$6 seconds to generate 20 poses for picking, and 9$\sim$10 seconds to generate 10 poses for placing. For bowls, it takes 7 seconds to generate 20 poses for picking and 17 seconds to generate 10 poses for placing. The sampling is slower for the bowls-on-dishes task because the point clouds in this task have more points than in the other tasks. As mentioned in Sec. D.2, we use two different models for low-resolution and high-resolution denoising. In addition, we use the energy-based critic to sort the sampled poses according to their quality. Therefore, three different models must be trained for each pick and place tasks. It takes less than 24 minutes to train each model for mug-picking and less than 36 minutes for mug-placing with an RTX3090 GPU. The bottles-on-a-shelf task requires slightly longer training time, amounting to 27 minutes for picking and 43 minutes for placing with an RTX3090 GPU. The bowls-on-dishes task requires a much longer training time because it consists of three different subtasks. It takes less than 47 minutes of training for picking and less than 1.3 hours for the placing. Note that the three models can be trained in parallel. Therefore, it takes less than an hour with three RTX3090 GPU to train our method for all tasks except for the bowl-placing task.

[4]Data and codes can be found in https://github.com/tomato1mule/diffusion_edf

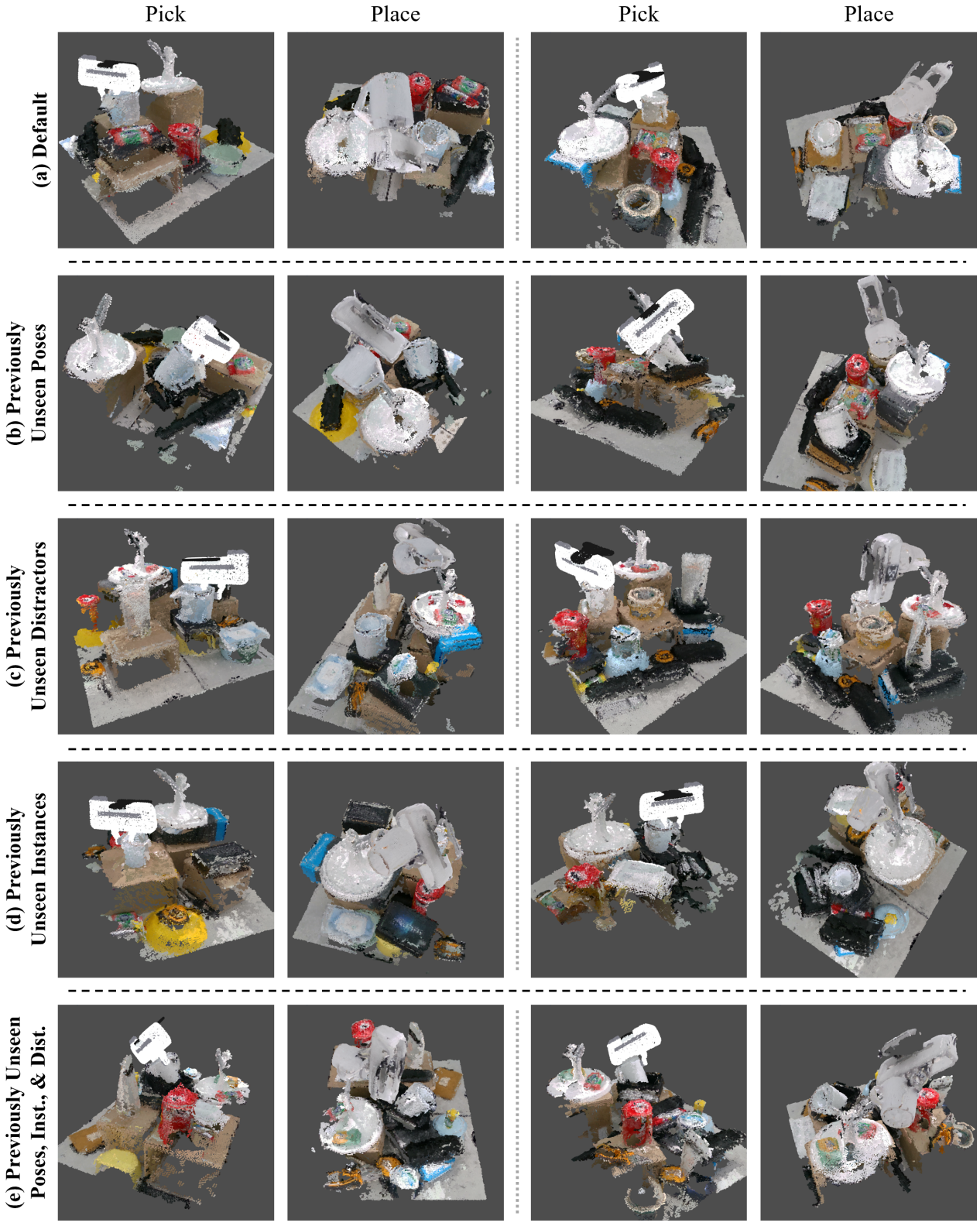

Figure 10. **Samples Generated by Diffusion-EDFs for Real-world Mug-on-a-hanger Task.** The figure depicts the end-effector pose samples for picking and placing a white mug on a white mug hanger. Diffusion-EDFs trained with only ten human demonstrations generated these samples from the real-world point cloud observations of the scene and grasp. Similar to our simulation experiments, we experiment for the **(a)** default scenario, **(b)** previously unseen target object poses (oblique; note that we only trained Diffusion-EDFs for upright poses) scenario, **(c)** previously unseen adversarial distractors (in white color) scenario, **(d)** previously unseen target object instances scenario, and **(e)** the all scenarios combined. The video of the denoising diffusion process can be found in https://sites.google.com/view/diffusion-edfs

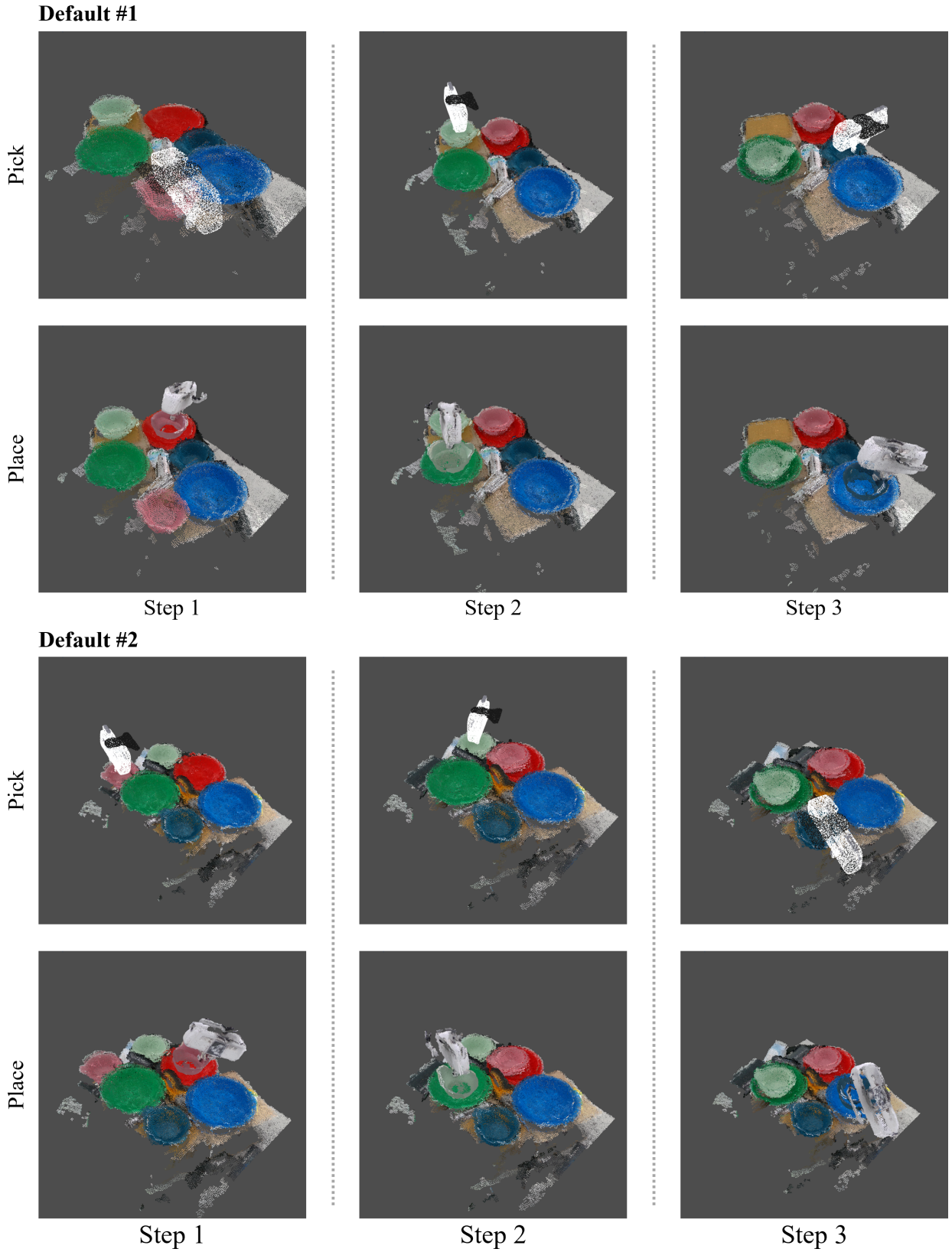

Figure 11. **Samples Generated by Diffusion-EDFs for Real-world Bowls-on-dishes Task.** The figure depicts the end-effector pose samples for picking and placing bowls on the dishes of matching colors in red-green-blue order. Diffusion-EDFs trained with only ten human demonstrations (three colored subtasks for each) generated these samples from the real-world point cloud observations of the scene and grasp. The video of the denoising diffusion process can be found in https://sites.google.com/view/diffusion-edfs

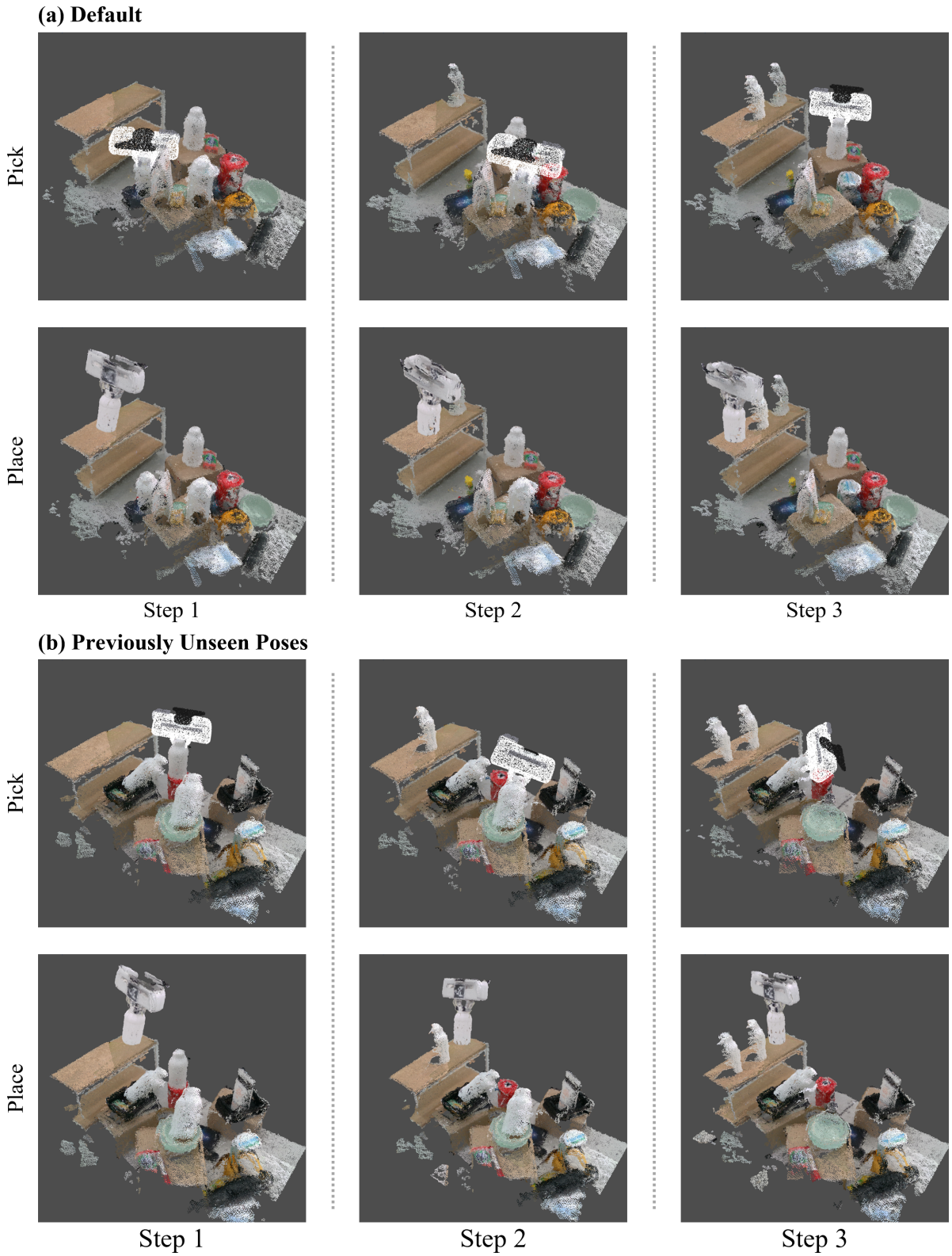

Figure 12. **Samples Generated by Diffusion-EDFs for Real-world Bottles-on-a-shelf Task.** The figure depicts the end-effector pose samples for picking and placing multiples bottles on a shelf. Diffusion-EDFs trained with only four human demonstrations (three sequential subtasks for each) generated these samples from the real-world point cloud observations of the scene and grasp. Similar to our simulation experiments, we experiment for the **(a)** default scenario and **(b)** previously unseen target object poses (oblique; note that we only trained Diffusion-EDFs for upright poses) sceneraio. The video of the denoising diffusion process can be found in https://sites.google.com/view/diffusion-edfs

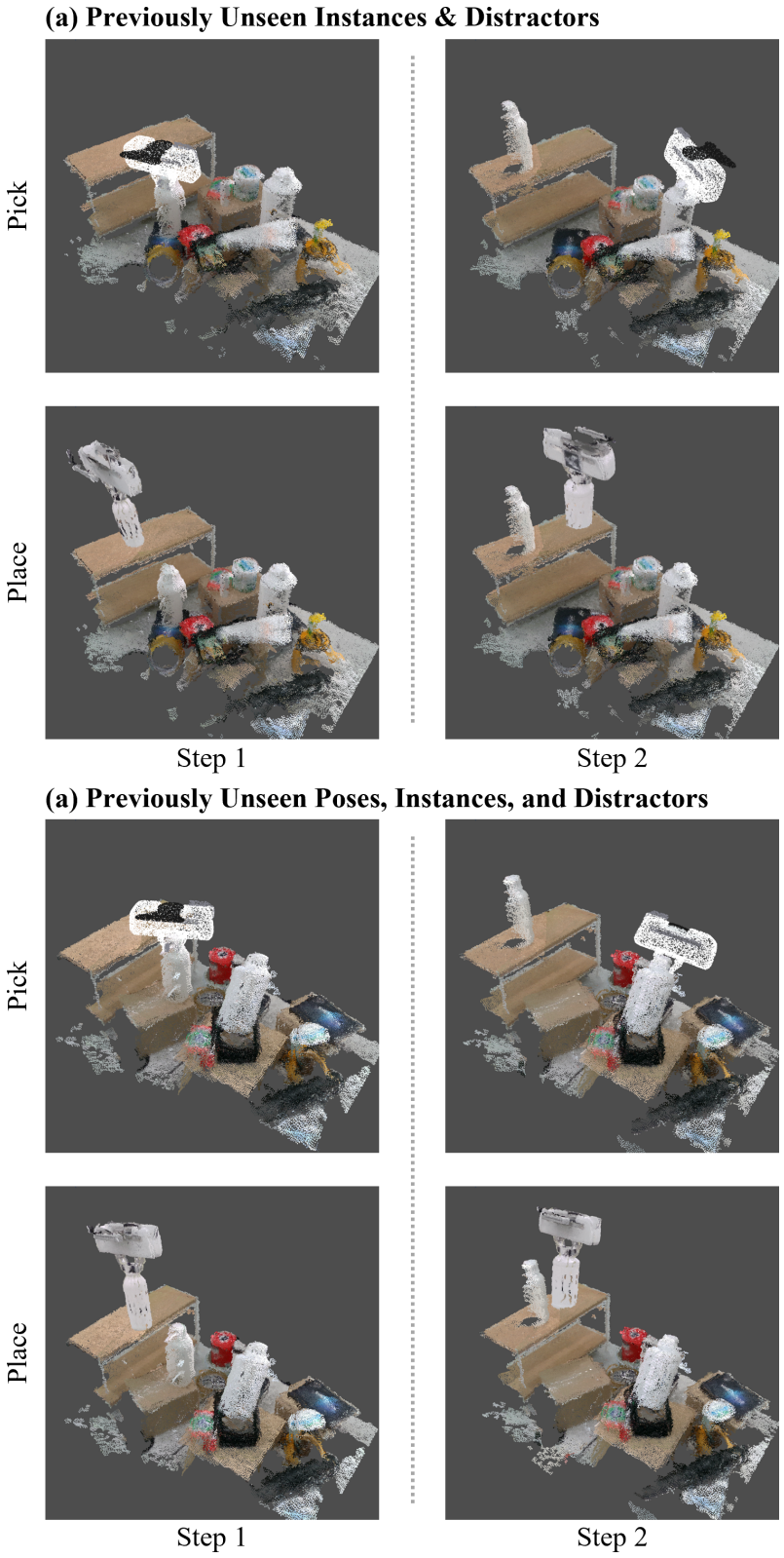

Figure 13. **Samples Generated by Diffusion-EDFs for Real-world Bottles-on-a-shelf Task (Previously Unseen Instances).** The figure depicts the end-effector pose samples for picking and placing multiples bottles on a shelf. In contrast to Fig. 12, we experiment with previously unseen bottle instances. Diffusion-EDFs trained with only four human demonstrations (three sequential subtasks for each) generated these samples from the real-world point cloud observations of the scene and grasp. Similar to Fig. 12, we experiment with both the **(a)** trained poses and **(b)** previously unseen poses (oblique; note that we only trained Diffusion-EDFs for upright poses). The video of the denoising diffusion process can be found in https://sites.google.com/view/diffusion-edfs