SELF: SE(3)-EQUIVARIANT LATENT FLOW POLICY FOR ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Equivariant visuomotor policies exploit geometric symmetries to improve data efficiency in robot manipulation, but repeatedly evaluating expressive equivariant networks over action sequences can make inference expensive. In this paper, we propose **SELF**, an **S**E(3)-**E**quivariant **L**atent **F**low policy that directly transports observation latents to action latents. SELF encodes observations and action chunks into the same representation of invariant scalars and equivariant vectors, enabling a lightweight velocity field to transport between endpoints with identical geometric transformation laws. To retain local geometry throughout transport, the velocity field accesses persistent scene context containing spatially anchored features and higher-order geometric information. This design reduces the cost of iterative generation while maintaining access to structured scene geometry. The latent transport is equivariant by construction under joint transformations of the source latent and scene context. The observation encoder, action autoencoder, and velocity field are trained jointly with reconstruction and flow-endpoint supervision. Experiments on eight simulated and three real-world manipulation tasks demonstrate that SELF matches strong equivariant policies in success rate, while achieving approximately 2.9× and 23.8× faster end-to-end inference than E3Flow and SDP-DDPM, respectively. Code will be released.

1 INTRODUCTION

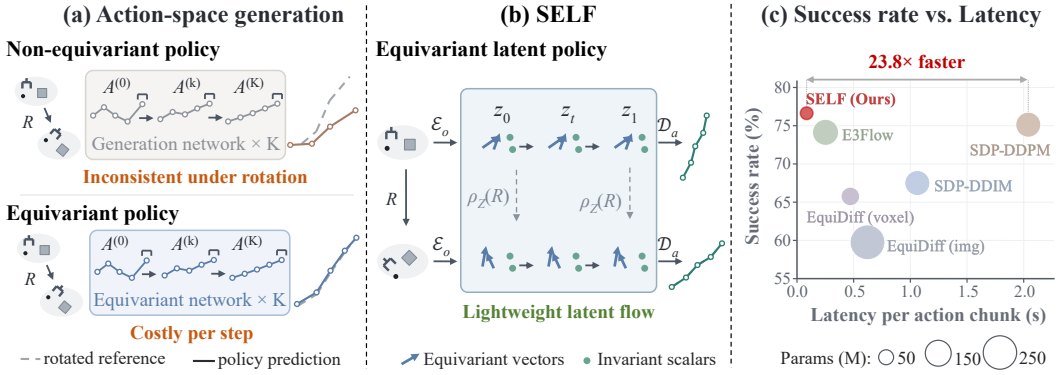


Figure 1: Comparison of SELF with action-space generative policies. (a) Non-equivariant policies do not enforce geometric consistency, while equivariant action-space policies repeatedly apply computationally heavy equivariant networks to action sequences, incurring high per-step computation. (b) Unlike action-space generation, SELF directly transports observation latents to action latents in a shared representation of invariant scalars and equivariant vectors, preserving geometric structure through lightweight latent updates. (c) SELF achieves competitive manipulation performance with fewer parameters and lower inference latency, including a 23.8× speedup over SDP-DDPM.

Learning generalizable manipulation skills from limited expert demonstrations is a central challenge in robot learning Hussein et al. (2017); Mandlekar et al. (2021); Zhao et al. (2023). Generative im-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

itation learning methods, including diffusion policies and flow matching models, have emerged as effective approaches to visuomotor control Lipman et al. (2022); Ze et al. (2024); Chi et al. (2025); Chisari et al. (2024); Zhang et al. (2025). By formulating action-sequence prediction as conditional generation from visual observations, these methods capture complex, multimodal action distributions. However, standard architectures do not explicitly enforce how predicted actions should transform under spatial rotations and translations of the scene. Learning these geometric relationships from demonstrations alone can require substantial data and limit generalization to spatial configurations beyond the training distribution Zhu et al. (2023).

Equivariant policies address this limitation by incorporating geometric symmetries directly into their architectures, providing an inductive bias for data-efficient learning and spatial generalization Cohen & Welling (2016); Fuchs et al. (2020); Deng et al. (2021); Wang et al. (2024); Yang et al. (2024a). Recent approaches, such as Spherical Diffusion Policy (SDP) Zhu et al. (2025) and E3Flow Zhang et al. (2026), combine geometric perception with equivariant action generators to preserve consistent transformation laws throughout generation. However, these methods repeatedly apply expressive equivariant networks to action sequences during denoising or flow integration, introducing substantial computational overhead. Even when observation features are computed once and reused, the generator must be evaluated at every sampling step, so inference cost depends on both the number of steps and the computation performed at each step. This raises a central question: can we retain the geometric structure of equivariant policies while reducing the cost of iterative action generation?

We argue that the computational cost of equivariant generation depends critically on the representation in which the generative process operates. Direct observation-to-action latent transport, as explored by VITA, offers a promising route to efficient generation Gao et al. (2026). However, equivariant observation-to-action latent transport requires geometric compatibility beyond matching observation and action latents dimensions. The two endpoints must share a consistent geometric transformation law so that interpolation and the learned transport preserve equivariance. Moreover, replacing an action-sequence generator with a lightweight latent velocity field should not sacrifice access to the local scene geometry needed for precise manipulation. These considerations motivate a shared equivariant latent representation with persistent geometric context throughout transport.

Building on these principles, we propose SELF, an SE(3)-Equivariant Latent Flow policy for robot manipulation. SELF encodes observations and action chunks into a shared representation of invariant scalars and equivariant vectors, and learns a lightweight equivariant velocity field that directly transports the observation latent to the action latent. An equivariant decoder then maps the transported latent to an action chunk. To preserve access to scene geometry throughout transport, SELF constructs persistent context from spatially anchored scalar, vector, and higher-order features. Invariant attention selects relevant scene information, while equivariant feature interactions incorporate it into the evolving latent. This construction preserves equivariance throughout latent transport under joint transformations of the latent state and scene context. Together, these components enable lightweight latent updates while retaining structured geometric information for action generation. The observation encoder, action autoencoder, and velocity field are trained jointly with reconstruction and flow-endpoint supervision.

We evaluate SELF on eight simulated and three real-world manipulation tasks to assess its manipulation performance and computational efficiency. Across the simulated tasks, SELF achieves a mean success rate of 76.6%, compared with 74.1% for E3Flow and 75.1% for SDP-DDPM, while delivering approximately $2.9\times$ and $23.8\times$ faster policy inference per action chunk, respectively. SELF also uses $4.2\times$ fewer parameters than E3Flow. In summary, our main contributions are as follows:

- We introduce SELF, a visuomotor policy that directly transports observation latents to action latents in a shared equivariant representation. A lightweight equivariant velocity field and action decoder preserve the geometric structure of this transport, with equivariance established for the context-conditioned latent transport.
- We develop a persistent equivariant scene-context mechanism that supplies spatially anchored and higher-order features throughout latent transport. Invariant attention and equivariant feature interactions maintain access to local scene geometry during generation.
- We conduct extensive experiments demonstrating that SELF achieves competitive success rates with substantially fewer parameters and lower inference latency than strong baselines.

2 Related Work

Generative Visuomotor Policies Generative imitation learning has achieved notable success in robotic manipulation by modeling complex, multimodal demonstration distributions. Frameworks such as Diffusion Policy (DP) Chi et al. (2025) and 3D Diffusion Policy (DP3) Ze et al. (2024) formulate policy learning as a conditional denoising process over action trajectories. To mitigate the computational burden of iterative diffusion sampling, recent works have explored flow matching and consistency-based approaches Lipman et al. (2022); Yang et al. (2024b); Zhang et al. (2025); Lu et al. (2024) to improve sampling efficiency. Despite their expressiveness and improved sampling speed, standard generative policies typically do not explicitly enforce continuous geometric symmetries. This can limit generalization to 3D rotations and translations outside the training distribution, motivating additional data augmentation or demonstration collection.

Equivariant Robot Learning Incorporating geometric symmetries, such as $SO(3)$ or $SE(3)$ equivariance, directly into policy architectures provides strong inductive biases that support spatial generalization and data-efficient learning Fuchs et al. (2020); Deng et al. (2021); Klein et al. (2023). In robotic manipulation, equivariance has been widely utilized for grasp pose estimation Ryu et al. (2024); Hu et al. (2024) and trajectory-level imitation learning Yang et al. (2024a); Wang et al. (2024). More recently, equivariant policies, including ET-SEED Tie et al. (2025), Spherical Diffusion Policy (SDP) Zhu et al. (2025), and E3Flow Zhang et al. (2026), have combined equivariant architectures with generative models.

Latent Generative Modeling and Cross-Modal Transport Operating generative models in learned latent spaces has proven effective for high-resolution image synthesis Rombach et al. (2022) and cross-modal generation Liu et al. (2025). In visuomotor control, VIsion-To-Action (VITA) Gao et al. (2026) recently introduced noise-free latent flow matching, transporting visual representations directly to action latents without requiring visual conditioning modules at every generation step. However, VITA operates in a latent space without explicitly enforced spatial transformation laws and therefore does not guarantee equivariance under 3D rotations and translations. SELF couples a lightweight scalar-vector transport state with persistent higher-order geometric context, preserving compatible transformation laws and access to local scene geometry throughout transport.

3 Preliminaries

3.1 Equivariance and Typed Features

Rigid transformations in three dimensions form the special Euclidean group $SE(3) = SO(3) \ltimes \mathbb{R}^3$, where

$$SO(3) = \{R \in \mathbb{R}^{3\times 3} \mid R^\top R = I_3,\ \det R = 1\}. \tag{1}$$

An element $g = (R, b) \in SE(3)$ acts on a point $p \in \mathbb{R}^3$ as $g \cdot p = Rp + b$. Given group actions on an input space $\mathcal{X}$ and an output space $\mathcal{Y}$, a mapping $f : \mathcal{X} \to \mathcal{Y}$ is equivariant if

$$f(g \cdot x) = g \cdot f(x), \qquad \forall g \in SE(3),\ x \in \mathcal{X}. \tag{2}$$

For a visuomotor policy, this property requires predicted actions to transform consistently with the observations under a joint rigid transformation Zhu et al. (2025); Zhang et al. (2026).

On feature spaces, a linear representation ρ specifies the group action and satisfies $\rho(g_1 g_2) = \rho(g_1)\rho(g_2)$. Equivariance then takes the form $f(\rho_{\text{in}}(g)x) = \rho_{\text{out}}(g)f(x)$. An invariant mapping is the special case in which the output representation is the identity, so that $f(\rho_{\text{in}}(g)x) = f(x)$.

Finite-dimensional continuous representations of $SO(3)$ decompose into irreducible representations (irreps) indexed by degree ℓ. We write a type-ℓ feature with C_ℓ channels as

$$F^{(\ell)} \in \mathbb{R}^{C_\ell \times (2\ell+1)}, \qquad F^{(\ell)} \mapsto F^{(\ell)} D^{(\ell)}(R)^\top, \tag{3}$$

where $D^{(\ell)}(R)$ is the degree-ℓ Wigner D-matrix in a real orthonormal basis. Type-0 features are invariant scalars. Type-1 features are three-dimensional vectors, with $D^{(1)}(R) = R$ in a Cartesian basis. Type-2 features have five components and are equivalent to symmetric traceless rank-two tensors in three dimensions. These feature types provide a structured representation of scene geometry Liao et al. (2023). Bias-free channel mixing within each type preserves its transformation law, while Clebsch–Gordan tensor products enable equivariant interactions across types. In particular, the $1 \otimes 2 \to 1$ component combines a vector and a type-2 feature to produce a vector.

3.2 Flow Matching

Flow matching learns a continuous-time velocity field that transports a source distribution p_0 to a target distribution p_1 Lipman et al. (2022). Let $(x_0, x_1) \sim \mu$ be paired endpoints in the same Euclidean space, where μ is a coupling with marginals p_0 and p_1. A linear conditional path is

$$x_t = (1-t)x_0 + tx_1, \qquad t \in [0,1], \tag{4}$$

with conditional target velocity $u_t = \frac{\mathrm{d}x_t}{\mathrm{d}t} = x_1 - x_0$. A neural velocity field f_ψ is trained to regress this velocity:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\substack{(x_0,x_1)\sim\mu \\ t\sim\mathcal{U}[0,1]}} \left[\| f_\psi(x_t, t; c) - (x_1 - x_0) \|_2^2 \right], \tag{5}$$

where c denotes optional conditioning associated with the endpoint pair. At inference, a source sample initializes the ODE

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f_\psi(x(t), t; c), \qquad x(0) = x_0, \tag{6}$$

which is numerically integrated from $t = 0$ to $t = 1$ with c held fixed. The coupling determines how endpoints are paired, and the source distribution need not be Gaussian. VITA applies this formulation to direct transport from observation latents to learned action latents in a non-equivariant visuomotor policy Gao et al. (2026).

4 Method

SELF reduces the cost of iterative action generation by transporting observations to actions in a shared equivariant latent space. Achieving this requires two conditions: the endpoints must obey compatible geometric transformation laws, while the evolving latent must retain access to scene geometry throughout transport. As illustrated in Fig. 2, to meet these requirements, an observation encoder supplies the source latent together with persistent scene context, whereas an action autoencoder provides the training target and maps the transported endpoint back to actions. A lightweight velocity field then connects the two endpoints through context-conditioned updates. Together, these components enable efficient latent transport without discarding the geometric information needed for action generation. We describe the shared representation and scene context below, followed by the transport mechanism and joint training objectives.

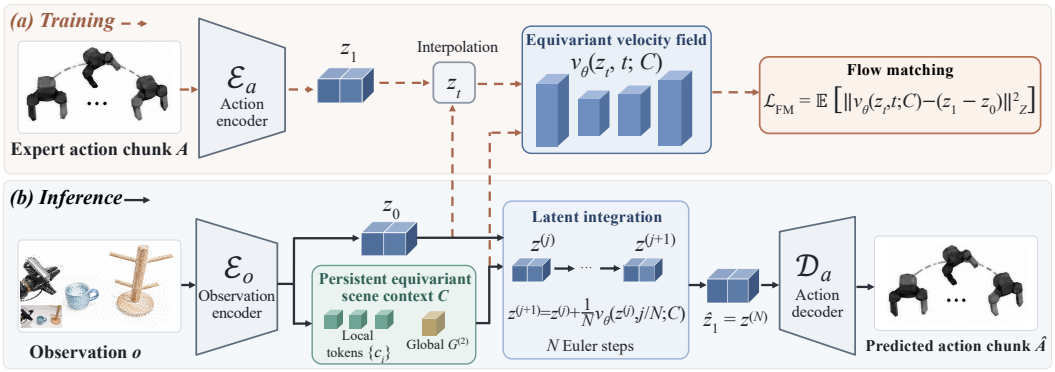


Figure 2: **Overview of SELF.** The observation encoder produces a source latent z_0 and persistent equivariant scene context C, comprising local tokens and global higher-order features. During training (dashed paths), an equivariant action encoder maps the demonstrated action chunk A to a target latent z_1 in the same scalar–vector representation. Interpolated states z_t supervise the context-conditioned velocity field through flow matching. At inference (solid paths), N Euler updates transport z_0 to $\hat{z}_1$, which the equivariant action decoder maps to the predicted action chunk $\hat{A}$. Context remains fixed throughout integration, providing geometric information at every update.

4.1 SHARED EQUIVARIANT LATENT REPRESENTATION

An observation o contains a colored point cloud, a wrist-camera image, and end-effector proprioception, and the policy predicts H absolute end-effector poses and gripper commands. To factor out global translations without rotating the coordinate axes, let $p_\star$ denote the current end-effector position and express each position as $\bar{p} = \gamma(p - p_\star)$, where $\gamma > 0$ is a common isotropic scale. Under a rigid transformation $g = (R, b)$, both p and $p_\star$ transform jointly, giving $\bar{p} \mapsto R\bar{p}$. We apply this centering and scaling convention to all positional inputs, including historical proprioception. Colors and image features are treated as invariant scalar side information under the joint transformation of the scene and the attached wrist camera.

We represent the normalized relative action chunk as $A = (a_h)_{h=1}^{H}$, where $a_h = (\bar{p}_h, r_{h,1}, r_{h,2}, u_h)$ consists of a relative position, two orientation vectors, and a scalar gripper command. Stacking these components across the action horizon gives the action autoencoder $3H$ vector channels and H scalar channels. Accordingly, we define $\rho_A(R)$ to rotate each vector channel while leaving the scalar channels unchanged. For this feature-space transformation to correspond to a physical rotation in the world frame, $r_{h,1}$ and $r_{h,2}$ are taken to be columns of the rotation matrix. When actions are stored using a row-based rotation-6D serialization, they must therefore be converted to this column-vector convention, and the vectors produced by the decoder must be orthogonalized. We detail these geometric conventions in Appendix A.

To make the two endpoints geometrically compatible, we assign them the same scalar–vector representation:

$$\mathcal{Z} = \mathbb{R}^{m\times 3} \oplus \mathbb{R}^n, \qquad z = (V, s), \qquad \rho_{\mathcal{Z}}(R)z = (VR^\top, s). \tag{7}$$

Following E3Flow Zhang et al. (2026), our observation encoder uses an EquiformerV2-based perception backbone Liao et al. (2023) to extract typed spherical features up to degree $\ell = 2$. The globally pooled type-0 and type-1 features are combined, within their respective types, with the gripper scalars and end-effector position and orientation vectors. An equivariant projection then maps the resulting features to the source latent $z_0 = (V_0, s_0)$. The pooled type-2 features are instead retained for the scene-context branch introduced in the next section.

Having constructed the observation endpoint z_0, we obtain the corresponding target endpoint by encoding the action chunk in the same representation. Specifically, the action encoder E_a and decoder D_a form an equivariant action autoencoder:

$$z_1 = E_a(A), \qquad \widetilde{A} = D_a(z_1). \tag{8}$$

Before encoding, the action horizon is folded into the channel dimension. Both modules are built from Vector Neuron residual blocks Deng et al. (2021), using channel mixing shared across spatial coordinates, invariant scalar gates, and no additive vector biases. Together with the action convention defined above, these operations ensure $E_a(\rho_A(R)A) = \rho_{\mathcal{Z}}(R)E_a(A)$ and $D_a(\rho_{\mathcal{Z}}(R)z) = \rho_A(R)D_a(z)$. Consequently, the repeated transport operates on a $(3m + n)$-coordinate latent state with no explicit temporal axis, while the action horizon is handled by the encoder and decoder. The resulting efficiency therefore comes from moving the iterative updates out of action-sequence space, rather than necessarily reducing the total number of latent coordinates.

4.2 PERSISTENT EQUIVARIANT SCENE CONTEXT

The shared representation makes the observation and action endpoints geometrically compatible, but compressing the observation into a single global latent can discard spatial details needed for precise manipulation. To preserve this information, we associate z_0 with persistent scene context $\mathcal{C} = (G^{(2)}, \{c_i\}_{i=1}^{K})$ that remains fixed and accessible throughout transport. Here, $G^{(2)}$ contains global type-2 features, while each local token $c_i = (S_i, U_i, T_i)$ contains scalar, vector, and type-2 channels. This separation allows the evolving state to remain a lightweight scalar–vector representation while the velocity field continues to access spatially localized and higher-order scene geometry.

We construct the local part of $\mathcal{C}$ from the backbone's final spatial feature map before global pooling. Let x_i denote the i-th spatial anchor in the centered and scaled coordinate system defined above, and let $F_i^{(\ell)}$ denote its type-ℓ feature under the convention of Eq. 3. We further denote the wrist-image embedding, shared across all anchors, by $\eta(I)$. The scalar, vector, and type-2 components of each

local token are then constructed as

$$\begin{aligned} S_i &= \phi_0\Big([F_i^{(0)}, \eta(I), \|x_i\|_2]\Big), \\ U_i &= [x_i^\top; W_1 F_i^{(1)}], \qquad T_i = W_2 F_i^{(2)}. \end{aligned} \tag{9}$$

The mappings in Eq. 9 are chosen to preserve the transformation law of each feature type. Specifically, ϕ_0 consists of LayerNorm followed by an MLP and acts only on invariant inputs, while W_1 and W_2 perform bias-free channel mixing without coupling the irrep coordinates. The type-1 features are expressed in the same Cartesian basis as the vector latent. In parallel with the local tokens, the global component $G^{(2)}$ is obtained through a separate bias-free channel projection of the pooled type-2 features. These constructions preserve the transformations in Eq. 3 and jointly define the context representation $\rho_{\mathcal{C}}(R)$. The local branch associates scene features and wrist-image information with individual spatial anchors, whereas the global branch provides a pooled summary of higher-order geometry. Together with the source latent, they form the complete observation map

$$(z_0, \mathcal{C}) = E_o(o). \tag{10}$$

Both outputs are computed once per action chunk: z_0 initializes the transport, while $\mathcal{C}$ remains available to the velocity field throughout its evolution.

4.3 Equivariant Observation-to-Action Latent Flow

With the endpoints and context defined, we instantiate the flow formulation of Section 3.2 using the observation–action pairing of each demonstration. For z_0 and z_1 from the same (o, A),

$$z_t = (1-t)z_0 + tz_1, \qquad u_t = z_1 - z_0. \tag{11}$$

The shared linear representation in Eq. 7 makes both the interpolated state and target velocity transform consistently under rotation. The source is observation-derived, with no added source noise. The velocity field $v_\theta(z, t; \mathcal{C})$ learns this transport while selecting context according to the current state.

For a type-ℓ feature X, define its channelwise invariant magnitude by

$$\nu_\ell(X)_c = \sqrt{\frac{1}{2\ell+1}\sum_{j=1}^{2\ell+1} X_{cj}^2 + \epsilon}, \tag{12}$$

where $\epsilon > 0$ ensures numerical stability. Equivariant input projections map z to hidden features (V, s). Each block uses the scalar time embedding $e(t)$ and invariant feature magnitudes to read local context. Suppressing block and head indices, attention uses

$$\begin{aligned} q &= f_q([\mathsf{s}, e(t), \nu_1(\mathsf{V})]), \\ k_i &= f_k([S_i, \log(1+\nu_1(U_i)), \log(1+\nu_2(T_i))]), \\ \alpha_i &= \frac{\exp(q^\top k_i/\sqrt{d})}{\sum_{j\in\mathcal{I}} \exp(q^\top k_j/\sqrt{d})}, \quad i \in \mathcal{I}, \end{aligned} \tag{13}$$

Here f_q and f_k are MLPs, d is the per-head query/key dimension, and $\mathcal{I}$ indexes valid tokens. Since the query and keys are invariant, so are the attention weights. The pooled values $(\bar{S}, \bar{U}, \bar{T}) = \sum_i \alpha_i (S_i, U_i, T_i)$ therefore retain their respective transformation laws. Multiple heads are concatenated along the channel dimension.

We normalize global type-2 channels as $\widehat{G}^{(2)} = G^{(2)}/\nu_2(G^{(2)})$ and retain their magnitudes as scalar inputs. A block updates its vector state through

$$\begin{aligned} \mathsf{V}^+ &= \mathsf{V} + \beta_g \mathcal{T}_g(\mathsf{V}, \widehat{G}^{(2)}) + \beta_v W_v \bar{U} + \beta_2 \mathcal{T}_2(\mathsf{V}, \bar{T}), \\ (\mathsf{V}', \mathsf{s}') &= \mathrm{VNBlock}\big(\mathsf{V}^+, [\mathsf{s}, e(t) + b_g + b_{\mathrm{loc}}]\big). \end{aligned} \tag{14}$$

The maps $\mathcal{T}_g$ and $\mathcal{T}_2$ are learned Clebsch–Gordan tensor products with the path $1 \otimes 2 \to 1$. W_v is a bias-free channel projection. b_g is an MLP of $\log(1+\nu_2(G^{(2)}))$. b_{loc} is an MLP of $[\bar{S}, \nu_1(\bar{U}), \nu_2(\bar{T})]$. Learned scalar gates $\beta_g, \beta_v, \beta_2$ control the context contributions. Stacking these blocks and applying

an equivariant output projection gives $v_\theta(z, t; \mathcal{C})$. Architecture widths and initialization settings are given in Appendix B. All context additions preserve feature types, yielding

$$v_\theta(\rho_{\mathcal{Z}}(R)z, t; \rho_{\mathcal{C}}(R)\mathcal{C}) = \rho_{\mathcal{Z}}(R)v_\theta(z, t; \mathcal{C}). \tag{15}$$

Thus, the field can adapt its use of local and higher-order information without introducing higher-order channels into the transported latent.

At inference, we encode the observation once and hold $\mathcal{C}$ fixed:

$$z^{(0)} = z_0, \qquad z^{(j+1)} = z^{(j)} + \frac{1}{N} v_\theta\left(z^{(j)}, \frac{j}{N}; \mathcal{C}\right), \quad j = 0, \dots, N-1. \tag{16}$$

We denote the integrated endpoint by $\widehat{z}_1 = \Phi_\theta^N(z_0; \mathcal{C}) = z^{(N)}$. The decoder produces the relative chunk $\widehat{A} = D_a(\widehat{z}_1)$. Positions are restored as $p_h = p_\star + \gamma^{-1}\widehat{p}_h$. The action encoder is used only during training. The following result establishes equivariance of the latent transport under joint transformations of its source and context.

Theorem 1 (Equivariance of context-conditioned latent transport)**.** *Suppose the velocity field satisfies Eq. 15 and D_a is equivariant. Then, for any fixed $N \geq 1$, the Euler-generated relative action chunk is rotationally equivariant under joint transformations of the source latent and scene context. If, additionally, the observation map satisfies $E_o(g \cdot o) = (\rho_{\mathcal{Z}}(R)z_0, \rho_{\mathcal{C}}(R)\mathcal{C})$ and the perception, sampling, and coordinate assumptions in Appendix A hold, the restored policy satisfies $\pi(g \cdot o) = g \cdot \pi(o)$.*

Because each Euler update is equivariant when the state and context transform jointly, the result holds for any fixed number of integration steps. We defer the proof and the required coordinate, orientation, and perception assumptions to Appendix A.

4.4 Training Objectives

We jointly optimize the encoders, velocity field, and decoder so that the shared latent space supports both transport and action reconstruction. For each demonstration pair $(o, A) \sim \mathcal{D}$, we sample $t \sim \mathcal{U}[0, 1]$ and evaluate the velocity field along the paired path defined in Eq. 11. For a latent difference $\Delta z = (\Delta V, \Delta s)$, we define

$$\|\Delta z\|_{\mathcal{Z}}^2 = \frac{\|\Delta V\|_F^2}{3m} + \frac{\|\Delta s\|_2^2}{n}. \tag{17}$$

This rotation-invariant metric averages the vector and scalar errors over their respective coordinates. The flow-matching objective is

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\substack{(o,A)\sim\mathcal{D} \\ t\sim\mathcal{U}[0,1]}} \left[\|v_\theta(z_t, t; \mathcal{C}) - (z_1 - z_0)\|_{\mathcal{Z}}^2 \right]. \tag{18}$$

Gradients from $\mathcal{L}_{\mathrm{FM}}$ are propagated through both endpoint encoders.

To ensure that both the learned action latents and the transported endpoints can be decoded into actions, we follow VITA Gao et al. (2026) and use action reconstruction (AE), flow-latent decoding (FLD), and flow-latent consistency (FLC) objectives. We differentiate through the N-step Euler rollout in Eq. 16 to obtain $\widehat{z}_1$, and define the normalized action error as $\ell_1(B, A) = \|B - A\|_1/(10H)$. Let sg denote the stop-gradient operator. The auxiliary objectives and the complete training loss are

$$\begin{aligned} \mathcal{L}_{\mathrm{AE}} &= \mathbb{E}_{(o,A)\sim\mathcal{D}} \left[\ell_1(D_a(z_1), A) \right], \\ \mathcal{L}_{\mathrm{FLD}} &= \mathbb{E}_{(o,A)\sim\mathcal{D}} \left[\ell_1(D_a(\widehat{z}_1), A) \right], \\ \mathcal{L}_{\mathrm{FLC}} &= \mathbb{E}_{(o,A)\sim\mathcal{D}} \left[\|\widehat{z}_1 - \mathrm{sg}(z_1)\|_{\mathcal{Z}}^2 \right], \\ \mathcal{L} &= \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{AE}}\mathcal{L}_{\mathrm{AE}} + \lambda_{\mathrm{FLD}}\mathcal{L}_{\mathrm{FLD}} + \lambda_{\mathrm{FLC}}\mathcal{L}_{\mathrm{FLC}}. \end{aligned} \tag{19}$$

The AE objective reconstructs actions directly from the target latent, whereas FLD and FLC supervise the transported endpoint in the action and latent spaces, respectively. Loss weights, endpoint supervision, and training settings are provided in Appendix B.

Table 1: Comparison of success rates (%) between SELF and state-of-the-art non-equivariant and equivariant policies on eight MimicGen tasks. For each task, results are averaged over three random seeds using the best periodic evaluation success rates. **Bold** and underline indicate the best and second-best results, respectively.

Method	Coffee D2	Coffee Preparation D1	Hammer Cleanup D1	Mug Cleanup D1	Nut Assembly D0	Square D2	Stack D1	Stack Three D1	Average
BC-RNN	42	16	20	22	32	6	48	10	24.50
ACT	22	34	32	30	38	4	38	4	25.25
DP	48	72	52	40	57	8	74	34	48.13
DP3	48	34	58	32	36	6	58	12	35.50
VITA	60	68	64	52	72	26	90	60	61.50
EquiBot	0	0	8	14	4	2	38	0	8.25
EquiDiff (img)	62	76	60	52	74	14	88	52	59.75
EquiDiff (voxel)	**68**	**80**	80	40	58	30	94	76	65.75
SDP (DDIM)	50	60	80	48	68	46	96	92	67.50
SDP (DDPM)	60	66	**86**	**56**	89	50	**100**	94	75.13
E3Flow	56	58	**86**	**56**	**91**	48	**100**	**98**	74.13
SELF (Ours)	64	68	82	**56**	83	**66**	**100**	94	**76.63**

5 EXPERIMENTS

5.1 DATASET AND IMPLEMENTATION DETAILS

Simulation Benchmarks We evaluate on eight MimicGen tasks Mandlekar et al. (2023) (as shown in Fig. 3): Coffee D2, Coffee Preparation D1, Hammer Cleanup D1, Mug Cleanup D1, Nut Assembly D0, Square D2, Stack D1, and Stack Three D1. These tasks span object placement, assembly, stacking, and multi-stage manipulation. We use 100 expert demonstrations per task and follow the E3Flow evaluation protocol Zhang et al. (2026): training for 500 epochs with three random seeds and evaluating every 20 epochs over 50 episodes. We report the best periodic evaluation success rate and the unweighted mean across the eight tasks.

Baselines We compare SELF with state-of-the-art visuomotor policies, including non-equivariant policies: BC-RNN Mandlekar et al. (2021), ACT Zhao et al. (2023), Diffusion Policy (DP) Chi et al. (2025), DP3 Ze et al. (2024), and VITA Gao et al. (2026). We also compare against equivariant policies, including EquiBot Yang et al. (2024a), Equivariant Diffusion Policy (EquiDiff) Wang et al. (2024) with image or voxel inputs, Spherical Diffusion Policy (SDP) Zhu et al. (2025) with DDIM or DDPM sampling, and E3Flow Zhang et al. (2026).

Implementation Details SELF is implemented in PyTorch. We jointly train the observation encoder, action autoencoder, and latent velocity field for 500 epochs using AdamW with $\beta_1 = 0.95$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and weight decay 10^{-6}. The learning rate is set to 10^{-4} and follows a cosine decay schedule after 500 warmup steps. We use a batch size of 16 with gradient accumulation over four batches and maintain an exponential moving average of the model parameters. The latent flow uses six Euler steps for both training-time endpoint supervision and inference. Policy inference latency per action chunk is measured on an NVIDIA RTX 4090 GPU. Additional architecture, training, and evaluation details are provided in Appendix B.

5.2 SIMULATION EXPERIMENTS

Manipulation Performance. Table 1 compares success rates across eight MimicGen tasks. As can be seen, equivariant policies generally achieve higher average success rates than non-equivariant baselines. For example, most equivariant policies achieve higher average success rates than even the strongest non-equivariant baseline, VITA. This overall advantage supports the value of geometric inductive biases when learning manipulation skills from limited demonstrations. Meanwhile, performance within the equivariant group varies across architectures and tasks. Voxel-based EquiDiff reaches 65.75% on average and leads the baselines on Coffee D2 and Coffee Preparation D1,

Table 2: Comparison of model size and inference efficiency between SELF and state-of-the-art equivariant policies on eight MimicGen tasks. We report parameter counts (M), FLOPs (G), and policy inference latency per action chunk (ms) on a single NVIDIA RTX 4090 GPU. The Average column reports the mean latency across tasks. **bold** and underline indicate the best and second-best results, respectively.

Method	Params (M) ↓	FLOPs (G) ↓	Policy Inference Latency per Action Chunk (ms) ↓								
			Coffee D2	Coffee Preparation D1	Hammer Cleanup D1	Mug Cleanup D1	Nut Assembly D0	Square D2	Stack D1	Stack Three D1	**Average**
EquiBot	142.59	124.60	1033.2	1055.5	979.8	1019.0	1045.8	987.6	1011.0	987.9	1015.0
EquiDiff (img)	271.18	965.05	612.6	611.8	615.7	616.6	635.0	631.0	622.2	633.4	622.3
EquiDiff (voxel)	77.58	260.16	457.2	466.2	458.6	459.7	470.8	455.1	524.8	477.3	471.2
SDP (DDIM)	140.29	92.79	1007.8	1045.5	1054.0	1100.0	1069.1	1022.7	1143.6	1052.4	1061.9
SDP (DDPM)	140.29	181.60	1970.3	2059.7	1978.0	2019.3	2005.0	2081.0	2081.8	2127.8	2040.4
E3Flow	151.62	21.99	253.3	242.3	233.3	264.9	238.0	262.9	268.8	253.0	252.1
SELF (Ours)	**35.92**	**5.49**	**85.0**	**80.9**	**84.6**	**86.7**	**86.3**	**83.9**	**93.1**	**84.3**	**85.6**

whereas SDP (DDPM) and E3Flow achieve stronger overall results of 75.13% and 74.13%, respectively. Their strengths are particularly evident in assembly and stacking: both achieve 100% on Stack D1, and E3Flow reaches 91% on Nut Assembly D0 and 98% on Stack Three D1. The gap between SDP's DDPM and DDIM variants, 75.13% versus 67.50%, also highlights the influence of the sampling configuration on task performance. SELF achieves the highest average success rate of 76.63%, exceeding SDP (DDPM) and E3Flow by 1.50 and 2.50 percentage points, respectively. Notably, SELF outperforms VITA by 15.13 percentage points, highlighting the effectiveness of our equivariant formulation with persistent geometric context within the observation-to-action latent transport framework.

Inference Efficiency. Table 2 reports the parameters, FLOPs, and policy inference latency per action chunk of the compared equivariant policies. Although most equivariant methods outperform non-equivariant baselines with limited demonstrations, their expressive equivariant architectures incur substantial computational costs, and repeated action-sequence updates lead to high inference latency that limits rapid closed-loop control. E3Flow accelerates generation relative to the diffusion-based equivariant baselines, yet still requires 151.62M parameters and 252.1 ms per action chunk on average. SELF addresses this computational burden by performing iterative generation with a lightweight scalar–vector latent field. With this design, SELF achieves the lowest parameter count and reported FLOPs among the compared methods, at 35.92M and 5.49G, respectively, and the lowest inference latency on every evaluated task, averaging 85.6 ms. Specifically, SELF uses 4.2× fewer parameters and runs 2.9× faster than E3Flow. Relative to SDP, SELF uses 3.9× fewer parameters and achieves 23.8× and 12.4× speedups over its DDPM and DDIM variants, respectively.

Due to space limitations, we provide real-world validation results, extensive ablation studies, and detailed analyses in Appendix C. We also provide video recordings of the real-world experiments in the supplementary materials.

6 Conclusion

We have presented SELF, a visuomotor policy that generates action chunks through direct observation-to-action transport in a shared equivariant latent representation. SELF combines geometrically compatible observation and action endpoints with a lightweight equivariant velocity field, while persistent scene context supplies spatially anchored local features and higher-order geometric information throughout transport. Our analysis establishes equivariance of the latent transport and identifies sufficient conditions for extending this property to the complete policy. Evaluations across eight simulated manipulation tasks and three real-robot tasks demonstrate competitive success rates. The computational comparison shows that SELF uses substantially fewer parameters and achieves lower inference latency than existing state-of-the-art equivariant policies. These results support shared equivariant latent transport as an effective approach to efficient action generation.

AI USE STATEMENT

We used generative AI tools to assist with implementing and debugging the SELF method, refining its mathematical formulation and proof drafts, providing feedback on experimental and ablation design, and polishing the language of the paper. All AI-assisted code was reviewed and tested by the authors, all mathematical claims and proofs were independently checked, and all text, citations, and empirical statements were verified against the implementation, experimental records, and original sources. The reported results were obtained from author-controlled experiments and were not generated by AI. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

All experiments were conducted either in simulation or on standard robotic platforms in controlled laboratory environments. This study did not involve human-subject research or the collection of personal, sensitive, or identifiable information. The simulation experiments use publicly available benchmark data, while the real-robot demonstrations were collected by the authors solely for training and evaluating the manipulation tasks considered in this work. Physical experiments were performed by trained researchers following standard laboratory safety procedures. The authors declare no conflicts of interest related to this work.

REPRODUCIBILITY STATEMENT

We support reproducibility by describing the model architecture, equivariant representations, training objectives, and inference procedure in the main text. The appendices provide the assumptions and complete proof of our equivariance result, together with architecture settings, hyperparameters, training configurations, and evaluation and ablation protocols. Our simulation experiments use the publicly available MimicGen benchmark, while the real-robot demonstrations were collected by the authors. The corresponding platform, data-collection procedure, and evaluation criteria are documented in the appendix. Upon acceptance, we will release the source code together with the training and evaluation scripts.

REFERENCES

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Eugenio Chisari, Nick Heppert, Max Argus, Tim Welschehold, Thomas Brox, and Abhinav Valada. Learning robotic manipulation policies from point clouds with conditional flow matching. In *Conference on Robot Learning (CoRL)*, 2024.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning (ICML)*, pp. 2990–2999. PMLR, 2016.

Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *ICCV*, pp. 12180–12189. IEEE, 2021.

Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *NeurIPS*, 33:1970–1981, 2020.

Dechen Gao, Boqi Zhao, Andrew Lee, Ian Chuang, Hanchu Zhou, Hang Wang, Zhe Zhao, Junshan Zhang, and Iman Soltani. Vita: Vision-to-action flow matching policy. In *ICLR*, volume 2026, pp. 100903–100933, 2026.

Boce Hu, Xupeng Zhu, Dian Wang, Zihao Dong, Haojie Huang, Chenghao Wang, Robin Walters, and Robert Platt. Orbitgrasp: $se(3)$-equivariant grasp learning. *arXiv preprint arXiv:2407.03531*, 2024.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. *NeurIPS*, 36:59886–59910, 2023.

Yi-Lun Liao, Brandon Wood, Abhishek Das, and Tess Smidt. Equiformerv2: Improved equivariant transformer for scaling to higher-degree representations. In *International Conference on Learning Representations (ICLR)*, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2022.

Qihao Liu, Xi Yin, Alan Yuille, Andrew Brown, and Mannat Singh. Flowing from words to pixels: A noise-free framework for cross-modality evolution. In *CVPR*, pp. 2755–2765. IEEE, 2025.

Guanxing Lu, Zifeng Gao, Tianxing Chen, Wenxun Dai, Ziwei Wang, and Yansong Tang. Manicm: Real-time 3d diffusion policy via consistency model for robotic manipulation. *arXiv preprint arXiv:2406.01586*, 2024.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on robot learning (CoRL)*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10674–10685. ieee, 2022.

Hyunwoo Ryu, Jiwoo Kim, Hyunseok An, Junwoo Chang, Joohwan Seo, Taehan Kim, Yubin Kim, Chaewon Hwang, Jongeun Choi, and Roberto Horowitz. Diffusion-edfs: Bi-equivariant denoising generative modeling on se (3) for visual robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 18007–18018, 2024.

Chenrui Tie, Yue Chen, Ruihai Wu, Boxuan Dong, Zeyi Li, Chongkai Gao, and Hao Dong. Etseed: Efficient trajectory-level se (3) equivariant diffusion policy. In *ICLR*, volume 2025, pp. 60114–60132, 2025.

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 *Conference on Robot Learning (CoRL)*, 2024.

Jingyun Yang, Zi-ang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning. In *Conference on Robot Learning (CoRL)*, 2024a.

Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, and Bin Cui. Consistency flow matching: Defining straight flows with velocity consistency. *arXiv preprint arXiv:2407.02398*, 2024b.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, pp. 14754–14762, 2025.

Qinglun Zhang, Shen Cheng, Tian Dan, Haoqiang Fan, Guanghui Liu, and Shuaicheng Liu. Efficient hybrid se (3)-equivariant visuomotor flow policy via spherical harmonics for robot manipulation. *arXiv preprint arXiv:2603.23227*, 2026.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

Xupeng Zhu, Dian Wang, Guanang Su, Ondrej Biza, Robin Walters, and Robert Platt. On robot grasp learning using equivariant models. *Autonomous Robots*, 47(8):1175–1193, 2023.

Xupeng Zhu, Fan Wang, Robin Walters, and Jane Shi. Se (3)-equivariant diffusion policy in spherical fourier space. In *International conference on machine learning (ICML)*, 2025.

A EQUIVARIANCE OF LATENT TRANSPORT

We prove Theorem 1 using the coordinate conventions and observation map described in Section 4. The full-policy result is conditional on the assumptions stated below.

Positions and translation. All point-cloud, proprioceptive, and action positions use the same reference $p_\star$ and isotropic scale γ, so that $\bar{p} = \gamma(p - p_\star)$. The reference transforms jointly with the scene. Historical proprioceptive positions must use the same convention, and decoded positions are restored by $p_h = p_\star + \gamma^{-1}\bar{p}_h$. A fixed affine normalizer $p \mapsto \gamma p + d$ is compatible only when the common offset is canceled for every positional feature: $(\gamma p + d) - (\gamma p_\star + d) = \gamma(p - p_\star)$.

Orientations. The two vectors in the rotation-6D action representation are interpreted as rotation-matrix columns. For a nonzero first vector a and a second vector b not parallel to a, the decoder constructs

$$e_1 = \frac{a}{\|a\|_2}, \qquad w = b - \langle e_1, b\rangle e_1, \qquad e_2 = \frac{w}{\|w\|_2}, \qquad e_3 = e_1 \times e_2. \tag{20}$$

Norms and inner products are invariant and $(Rx) \times (Ry) = R(x \times y)$ for $R \in SO(3)$. Hence $Q(a, b) = [e_1, e_2, e_3]$ satisfies $Q(Ra, Rb) = RQ(a, b)$. A row-based rotation-6D serialization must be converted to and from this column convention. The orientation result is restricted to nondegenerate vector pairs. A fallback that inserts a fixed world-axis direction is not covered by the theorem.

Observation encoder. The observation map is assumed equivariant under the joint transformation of the scene, robot state, and attached camera. Colors and image embeddings are scalar side information. Sampling must return corresponding anchors and neighborhoods under the transformation, up to a common permutation of anchors, features, and masks. Distance-based sampling requires consistent initialization and tie handling. Random sampling requires coupled randomness for a samplewise comparison. The implementation uses random FPS initialization and 10^{-6} random perturbations of graph edge vectors. Thus, the exact proof assumes corresponding samples and rotated edge perturbations. Independent random draws do not imply samplewise equivariance. The spherical features and Cartesian vectors are also assumed to use compatible irrep bases.

Latent interpolation. Rotations are represented by orthogonal matrices on each irrep, while scalar side information is unchanged. For any $R \in SO(3)$, linearity of $\rho_{\mathcal{Z}}(R)$ gives

$$(1 - t)\rho_{\mathcal{Z}}(R)z_0 + t\rho_{\mathcal{Z}}(R)z_1 = \rho_{\mathcal{Z}}(R)((1 - t)z_0 + tz_1). \tag{21}$$

The endpoint difference obeys the same identity. Thus, assigning the paired endpoints the same representation is sufficient for the interpolation and its regression target to transform equivariantly.

Velocity field. For $X^{(\ell)} \mapsto X^{(\ell)}D^{(\ell)}(R)^\top$, orthogonality gives $\nu_\ell(X^{(\ell)}D^{(\ell)}(R)^\top) = \nu_\ell(X^{(\ell)})$. Therefore, the queries, keys, and weights in Eq. 13 are invariant. The attended value of a rotated token set satisfies

$$\sum_i \alpha_i X_i^{(\ell)} D^{(\ell)}(R)^\top = \left(\sum_i \alpha_i X_i^{(\ell)}\right) D^{(\ell)}(R)^\top. \tag{22}$$

The same result holds under a common permutation of tokens. Bias-free channel projections commute with the group action, and the tensor-product paths obey

$$\mathcal{T}(VR^\top, TD^{(2)}(R)^\top) = \mathcal{T}(V, T)R^\top. \tag{23}$$

Scalar gates and invariant scalar conditions preserve these identities. Consequently, every block in Eq. 14, and hence their composition, satisfies Eq. 15.

Euler integration and action decoding. Let $z^{(j)}$ be generated from $(z_0, \mathcal{C})$ and let $z_R^{(j)}$ be generated from $(\rho_{\mathcal{Z}}(R)z_0, \rho_{\mathcal{C}}(R)\mathcal{C})$. The initialization obeys $z_R^{(0)} = \rho_{\mathcal{Z}}(R)z^{(0)}$. Assuming this relation at step j and writing $h = 1/N$,

$$z_R^{(j+1)} = \rho_{\mathcal{Z}}(R)z^{(j)} + hv_\theta(\rho_{\mathcal{Z}}(R)z^{(j)}, jh; \rho_{\mathcal{C}}(R)\mathcal{C}) \tag{24}$$

$$= \rho_{\mathcal{Z}}(R)\left(z^{(j)} + hv_\theta(z^{(j)}, jh; \mathcal{C})\right) = \rho_{\mathcal{Z}}(R)z^{(j+1)}. \tag{25}$$

Thus, the endpoint is equivariant for any fixed number of Euler steps. Applying the equivariant decoder D_a gives the same property for the relative action chunk.

For translation, the transformed reference is $Rp_\star + b$ and the relative prediction is $R\bar{p}_h$. Therefore,

$$(Rp_\star + b) + \gamma^{-1} R\bar{p}_h = R(p_\star + \gamma^{-1}\bar{p}_h) + b. \tag{26}$$

Together with the orientation convention above and an invariant gripper command, this proves the SE(3)-equivariance of the decoded actions under the stated assumptions, completing the proof of Theorem 1.

Numerical verification of latent transport. To complement the proof, we numerically evaluate the latent-transport component of Theorem 1 using trained models. This test starts after observation encoding and ends before action decoding, isolating the velocity field and its six-step Euler transport. We encode eight held-out observations per task once and apply 16 random $\mathrm{SO}(3)$ rotations to each resulting latent–context pair, yielding 128 comparisons per task. All vector features and type-2 context features transform according to R and $D^{(2)}(R)$, respectively, while scalar features and token masks remain unchanged. For each evaluated map F, we compare $F(\rho_{\mathcal{Z}}(R)z; \rho_{\mathcal{C}}(R)\mathcal{C})$ with $\rho_{\mathcal{Z}}(R)F(z;\mathcal{C})$, keeping time fixed between the two paths for velocity-field evaluations. Writing the original and transformed-input outputs as (V, s) and (V_R, s_R), respectively, we measure relative RMS errors:

$$e_V = \frac{\|V_R - VR^\top\|_F}{\|VR^\top\|_F + \epsilon}, \qquad e_s = \frac{\|s_R - s\|_2}{\|s\|_2 + \epsilon}, \qquad \epsilon = 10^{-12}.$$

Table 3 reports the mean, 95th-percentile, and maximum errors. Across both tasks, the maximum six-step endpoint errors are 1.51×10^{-7} for vector channels and 1.25×10^{-7} for scalar channels. These small residuals provide numerical evidence that the trained latent transport preserves the prescribed transformation laws, consistent with the theoretical result above.

Table 3: Numerical $\mathrm{SO}(3)$ equivariance of the trained latent velocity field and six-step Euler transport. Entries report mean / 95th percentile / maximum relative RMS error.

Task	Evaluated map	Vector error e_V ($\times 10^{-7}$)	Scalar error e_s ($\times 10^{-7}$)
Stack Three D1	Velocity field	2.44/3.52/5.20	1.54/2.08/2.30
	Six-step endpoint	1.07/1.22/1.34	0.929/1.11/1.25
Mug Cleanup D1	Velocity field	3.19/6.40/10.3	1.33/1.71/1.82
	Six-step endpoint	1.09/1.29/1.51	0.800/0.975/1.17

B Implementation Details

Architecture. The latent width is fixed across integration steps, whereas the action horizon is folded into action-encoder and decoder channels. The action encoder is a training-only component. The observation encoder and persistent context are computed once per action chunk. Only the velocity field is repeatedly evaluated during integration.

Input and output. The observation encoder uses the latest colored point cloud and wrist image, together with two proprioceptive steps. Each point cloud contains 1,024 XYZ–RGB points, obtained by depth observations from the external camera, cropping to the workspace, and farthest-point sampling. The Equiformer backbone uses FPS–kNN pooling with $\ell_{\max} = m_{\max} = 2$, channel widths $(16, 32, 64, 128, 256)$, four pooling ratios of 0.5, neighbor limits $(16, 16, 16, 16, 128)$, radius caps of 3 in normalized coordinates, and grid resolution 12. For each task, let d_x, d_y, d_z be the half-ranges of the dataset's point-cloud coordinates. We use the isotropic scale $\gamma = 1/\max(d_x, d_y, d_z)$ for point-cloud, end-effector, and action positions, with the current end-effector position as the reference. Point-cloud and wrist RGB values are normalized to $[-1, 1]$. The wrist image is cropped from 84×84 to 76×76 and encoded by a ResNet-18 trained from scratch with its normalization layers removed. A 32-keypoint SpatialSoftmax produces a 64-dimensional feature, which enters the scalar channels of the local context. Each proprioceptive step contains end-effector position and

Table 4: SELF architecture. Channel counts denote irrep multiplicities. An order-ℓ channel contains $2\ell + 1$ coordinates. In the autoencoder widths, v and s denote vector and scalar channels.

Component	Setting
Shared latent	$m = n = 128$: 128 vector channels and 128 scalar channels (512 real coordinates in total).
Local context	The local context has $K = 64$ tokens. Each token has 32 scalar, 16 vector, and 16 type-2 channels.
Global context	32 type-2 channels.
Action autoencoder	Two Vector Neuron residual blocks in both the encoder and decoder.
Autoencoder widths	The encoder maps $(48v, 16s) \to (128v, 256s) \to (128v, 128s)$. The decoder uses the reverse mapping.
Latent velocity field	The field uses four residual blocks with 128 vector and 512 scalar hidden channels. The sinusoidal time embedding has 64 dimensions.
Local attention	Four heads with per-head dimension 16.
Action chunking	The policy uses two proprioceptive observation steps and predicts $H = 16$ actions. Eight actions at zero-based indices $1, \ldots, 8$ are executed.
Flow integration	Six Euler steps for both differentiable endpoint supervision and inference.
Initialization	Vector/type-2 gates are zero-initialized. The final layers of the global and local scalar-bias MLPs are also zero-initialized.

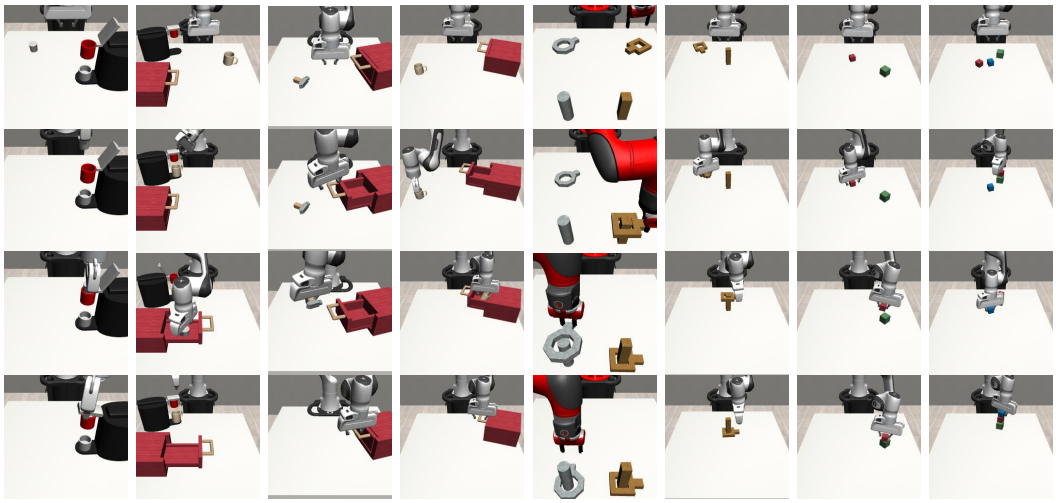


Figure 3: Visualization of the execution process for eight MimicGen manipulation tasks. Each column presents a representative episode, with snapshots ordered temporally from the initial scene configuration at the top to successful task completion at the bottom.

orientation and two gripper joint positions. Orientation is converted to two three-dimensional vectors, and gripper joint positions are range-normalized. Each predicted action contains position, two orientation vectors, and a gripper scalar. Action orientation and gripper values are not normalized. After restoring the position reference and undoing normalization, the output is converted to XYZ, axis-angle rotation, and gripper commands for absolute operational-space control at 60 Hz.

Training. All eight tasks use the same model and training hyperparameters. We train for 500 epochs with seeds 42, 43, and 44 using AdamW with learning rate 10^{-4}, betas $(0.95, 0.999)$, epsilon 10^{-8}, and weight decay 10^{-6}. The minibatch size is 16 with gradient accumulation over four batches, giving an effective batch size of 64. The learning rate follows a cosine schedule to zero

after 500 warmup steps. EMA uses an inverse-gamma schedule with inverse gamma 1, power 0.75, and decay bounds $[0, 0.9999]$, starting without an update delay. Following VITA Gao et al. (2026), the auxiliary loss weights are $(\lambda_{\mathrm{AE}}, \lambda_{\mathrm{FLD}}, \lambda_{\mathrm{FLC}}) = (0.5, 0.5, 1)$. AE and FLD use mean coordinatewise L_1 error on normalized action chunks. FLC uses the latent metric in Eq. 17. FLD and FLC differentiate through all six Euler steps, with stop-gradient applied only to the FLC target. FM updates both endpoint encoders. The latent metric is rotation-invariant. Coordinatewise action L_1 is a reconstruction objective and does not change architectural equivariance. Training uses random image crops, without rotation augmentation, added observation or point-cloud noise, gradient clipping, or grasp reweighting.

Validation. For each task, 2% of the 100 demonstrations are held out at the episode level for offline validation, which uses raw model weights. Online evaluation uses EMA weights and deterministic center image crops. Following E3Flow Zhang et al. (2026), we evaluate and save checkpoints every 20 epochs over the same 50 episodes, with seeds 100000–100049. We first select the best periodic evaluation within each training seed and then average success rates across the three seeds. The overall average is the unweighted mean over the eight tasks. Inference uses six Euler steps and executes eight actions per chunk. Rollouts use ten asynchronous environments with the same fixed episode set. Table 2 reports parameter counts, FLOPs, and policy inference latency per action chunk on a single NVIDIA GeForce RTX 4090 GPU. Each method–task pair is loaded and measured independently and serially, with batch size one and native FP32 precision, without autocast or compilation. Measurements cover one complete fixed episode with test seed 100000. Ten untimed warmup calls precede each pair's measurement, after which policy state and CPU/CUDA random-number-generator states are restored. Latency is measured as CUDA-synchronized wall time. The timed region covers only `policy.predict_action` with inputs already on the GPU, excluding simulator steps, rendering, point-cloud generation, inter-process communication, video processing, and CPU–GPU transfers.

C Additional Experiments

C.1 Real-Robot Experiments

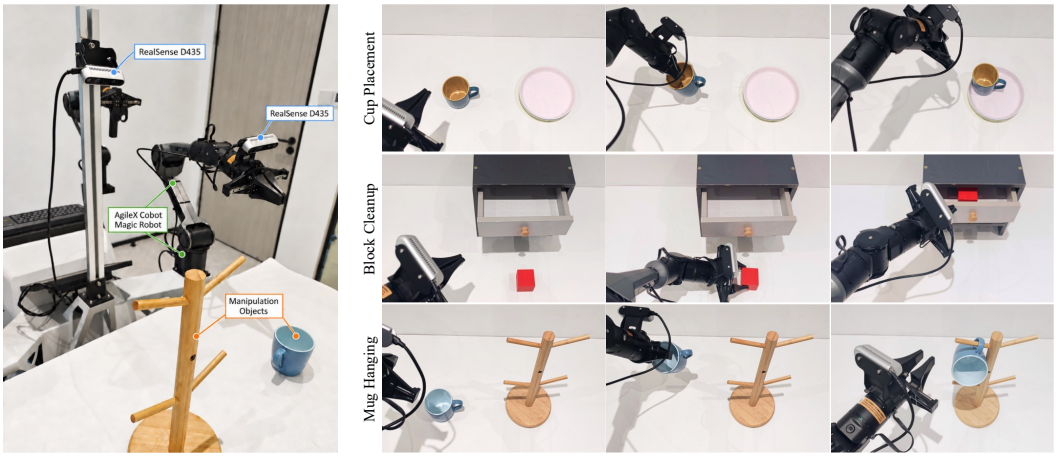
Experimental setup. We conduct real-robot experiments on an AgileX Cobot Magic platform, using the left 6-DOF Piper arm with a parallel gripper for policy execution. Expert demonstrations are collected through master–slave teleoperation. Two Intel RealSense D435i cameras provide head-mounted RGB-D observations and wrist-mounted RGB observations. The head-camera point cloud is transformed into the robot base frame, cropped to the workspace, and sampled to 1,024 colored points. Wrist images are resized to 84×84. The policy additionally receives end-effector position, orientation, and gripper state.

We collect 50 successful demonstrations per task at 30 Hz and train a separate SELF policy for each task. Each policy is trained for 500 epochs using AdamW with a learning rate of 10^{-4} and an effective batch size of 64. Deployment uses EMA weights and six Euler integration steps. The policy predicts 16-step action chunks and executes eight actions before replanning. Inference runs on an NVIDIA RTX 4090 GPU. Target actions are specified at 30 Hz and interpolated by the low-level controller at 60 Hz.

We compare SELF with E3Flow, DP, and SDP over 20 trials per task and method. All methods use the same hardware, calibration, initial-state distributions, and low-level control and safety constraints. Each trial begins from the corresponding task's training initial-state distribution and ends upon success or a clear failure. We report success rate as the fraction of successful trials.

Task details.

- **Cup Placement.** The robot picks up a cup and places it on a target plate. This task requires locating both objects and controlling the cup during transport and placement.
- **Block Cleanup.** The robot picks up a block and places it inside a drawer. This task requires guiding the grasped block into the drawer opening.



(a) Real-world robotic platform

(b) Execution process of real-world robotic tasks.

Figure 4: Illustration of (a) the robotic platform used in our real-world experiments and (b) representative execution sequences on three real-world manipulation tasks.

Table 5: Success rates (%) of different methods on our three real-world tasks. Best results are shown in bold.

Method	NFE	Cup Placement	Block Cleanup	Mug Hanging	Average
DP	100	30	0	5	11.67
SDP	100	**95**	30	20	48.33
E3Flow	10	90	50	**45**	61.67
SELF (Ours)	6	**95**	**65**	40	**66.67**

- **Mug Hanging.** The robot picks up a mug and hangs it by its handle on a mug rack. This task requires aligning the handle with the target hook and releasing the mug so that it remains supported by the rack.

Results analysis. Table 5 reports success rates on the three real-world tasks. SELF achieves an average success rate of 66.67%, compared with 61.67% for E3Flow, 48.33% for SDP, and 11.67% for DP. On Cup Placement, SELF and SDP both achieve 95% success. On Block Cleanup, SELF reaches 65%, compared with 50% for E3Flow and 30% for SDP. Mug Hanging remains challenging for all methods, with SELF achieving 40% success compared with E3Flow's 45%. Overall, SELF achieves competitive or better manipulation performance using only 6 function evaluations, compared with 10 for E3Flow and 100 for the diffusion baselines. These results demonstrate that lightweight latent transport can reduce the number of iterative network evaluations while maintaining strong real-world performance.

C.2 Ablation Studies

Shared Equivariant Latent Representation. We first evaluate the shared equivariant representation introduced in Section 4.1 by replacing the equivariant action autoencoder and latent velocity field with an MLP-based backend. This variant retains observation-to-action latent generation but does not enforce the common geometric transformation law used by SELF. Comparing rows 1 and 7 of Table 6, the MLP backend achieves 59.50% average success, compared with 77.75% for full SELF. The equivariant formulation improves performance on all four tasks, with particularly large gains on Square D2 and Coffee Preparation D1, where success increases from 32% to 66% and from 48% to 68%, respectively. These results support our design of geometrically compatible observation and action endpoints connected by an equivariant velocity field: preserving geometric structure within latent generation contributes to manipulation performance beyond the use of observation-to-action transport alone.

Table 6: Success rates (%) of our ablation variants on four representative manipulation tasks. **Bold** and underline indicate the best and second-best results in each column, respectively.

Variant	Stack Three D1	Square D2	Coffee Preparation D1	Nut Assembly D0	Average
1) MLP backend	86	32	48	72	59.50
2) Global context	**94**	40	64	83	70.25
3) Source only	92	48	52	79	67.75
4) w/o global $l = 2$ context	88	46	58	**84**	69.00
5) w/o local $l = 2$ context	92	48	60	77	69.25
6) w/o $l = 2$ context	90	38	52	81	65.25
7) SELF (Full)	**94**	**66**	**68**	83	**77.75**

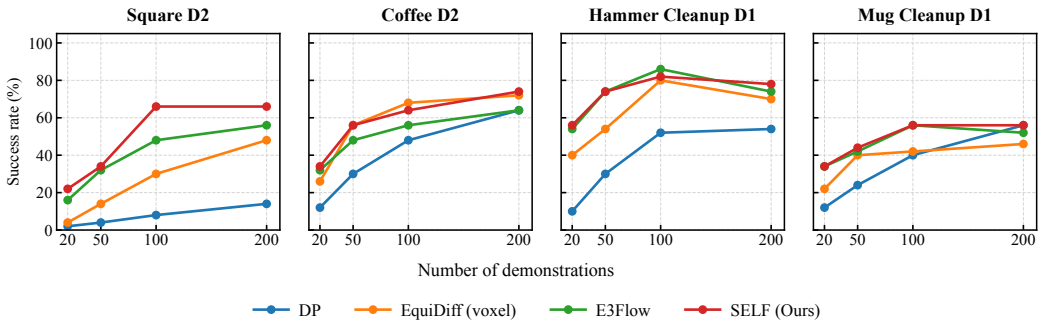


Figure 5: Data efficiency on four MimicGen tasks with varying numbers of expert demonstrations. Equivariant policies perform better than DP when demonstrations are scarce, while SELF maintains comparable success rates to the equivariant baselines.

Persistent Equivariant Scene Context. We next examine the role of preserving spatially anchored scene information throughout transport, as proposed in Section 4.2. Row 2 replaces spatial tokens with globally pooled scene features, while row 3 supplies scene information through the initial latent without additional context during transport. Their average success rates decrease to 70.25% and 67.75%, respectively, compared with 77.75% for full SELF in row 7. Comparing rows 2 and 7, retaining spatial context improves Square D2 from 40% to 66%. Comparing rows 3 and 7, full SELF improves performance on all four tasks. Although globally pooled features recover part of the average performance, row 2 does not consistently outperform row 3 across tasks. These results support the separation of the evolving latent from persistent spatial context: the initial global representation establishes the transport source, while spatially anchored features provide additional scene information that remains accessible as the latent evolves. This allows lightweight latent generation to retain the local geometric information needed for manipulation.

Higher-Order Geometric Context. Finally, we evaluate the design of retaining higher-order geometry in the context while keeping the transported latent restricted to invariant scalars and equivariant vectors. Rows 4 and 5 remove the global type-2 features $G^{(2)}$ and local type-2 features $\{T_i\}$, respectively, while row 6 removes both. Compared with full SELF in row 7, removing all higher-order context reduces average success from 77.75% to 65.25%. Retaining only the local branch (row 4) or the global branch (row 5) improves this to 69.00% and 69.25%, respectively. Using both branches in row 7 achieves the highest average success and outperforms row 6 on every task. These results support the complementary roles of global and spatially localized higher-order features. Through the context-conditioned interactions described in Section 4.3, SELF benefits from this richer geo-

metric information without introducing higher-order channels into the evolving latent, supporting the proposed separation between a lightweight transport state and structured geometric context.

Data Efficiency We also evaluate the data efficiency of SELF against DP, EquiDiff, and E3Flow on four MimicGen tasks. As shown in Fig. 5, DP benefits substantially from additional demonstrations but achieves relatively low success rates when demonstrations are scarce. Equivariant methods perform better in these low-data settings, consistent with the benefits of geometric inductive biases. SELF maintains comparable performance to the equivariant baselines across demonstration budgets, including when only 20 or 50 demonstrations are available. Together with the efficiency results in Table 2, these results show that SELF retains comparable data efficiency while substantially reducing computational cost.