ACTIONFLOW: Equivariant, Accurate, and Efficient Manipulation Policies with Flow Matching

Niklas Funk, Julen Urain, Joao Carvalho, Vignesh Prasad, Georgia Chalvatzaki, and Jan Peters
Department of Computer Science, Technical University of Darmstadt, Germany
niklas@robot-learning.de

Abstract: Spatial understanding is a critical aspect of most robotic tasks, particularly when generalization is important. Despite the impressive results of deep generative models in complex manipulation tasks, the absence of a representation that encodes intricate spatial relationships between observations and actions often limits spatial generalization, necessitating large amounts of demonstrations. To tackle this problem, we introduce a novel policy class, **ActionFlow**. ActionFlow integrates spatial symmetry inductive biases while generating expressive action sequences. On the representation level, ActionFlow introduces an SE(3) Invariant Transformer architecture, which enables informed spatial reasoning based on the relative SE(3) poses between observations and actions. For action generation, ActionFlow leverages Flow Matching, a state-of-the-art deep generative model known for generating high-quality samples with fast inference – an essential property for feedback control. In combination, ActionFlow policies exhibit strong spatial and locality biases and SE(3)-equivariant action generation. Our experiments demonstrate the effectiveness of ActionFlow and its two main components on simulated and real-world robotic manipulation tasks and confirm that ActionFlow yields equivariant, accurate, and efficient policies. Project website: https://flowbasedpolicies.github.io/

Keywords: Learning from Demonstrations, Behavioral Cloning, Deep Generative Models, Flow Matching, Robot Manipulation

1 Introduction

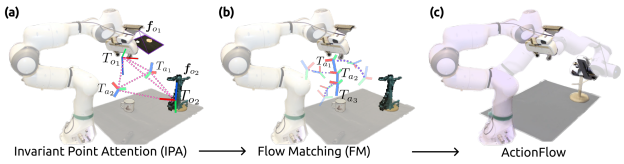

Figure 1: **Overview of ActionFlow.** ActionFlow represents both observations & actions in one common space and describes every token by its pose T and features $\boldsymbol{f}$. **(a)** The scene consists of two observations and a randomly initialized action sequence. **(b)** Given the scene, a geometric transformer computes the spatial attention between the tokens considering their relative SE(3) poses. Its output defines a flow in the action space, resulting in local robot trajectories. **(c)** When using the procedure of scene encoding & action generation as a policy, we get accurate, efficient & SE(3) equivariant action sequences at low inference times.

Recently, we observed impressive results in using deep generative models for solving complex manipulation tasks [1, 2, 3, 4, 5]. Yet, it is well known that models that naively integrate observations and actions usually require copious amounts of demonstrations for good task performance. In this direction, there has been a collection of research that explored how to exploit the spatial relations between observations and actions [6, 7, 8, 9, 10, 11] to learn more sample efficient policies. **Equivariant** policies generalize the policy's behavior under global translations or rotations [12, 7, 6, 13, 14, 15, 16, 17], thereby adding an effective inductive bias.
Herein, we are not only interested in equivariant policies, but also in capturing **local spatial relations** [18, 19, 7]. Consider, for example, picking a mug and hanging it (cf. Fig. 1). When the robot is approaching to pick up the mug, the robot should be capable of reasoning based on the relative poses between its own pose, the mug, and its next actions. But when hanging the mug, the robot

should also focus on the relative poses between the mug and the hanger. Thus, equipping the policy with the ability to reason based on the spatial relation between the different observations and actions might be essential to learning policies efficiently. How can we integrate all these desiderata and still learn **dexterous, fast, and expressive policies from demonstrations**?

Inspired by the recent successes from the protein folding community [20, 21, 22, 23, 24], in which SE(3) symmetric models are integrated with highly-expressive deep generative models, we introduce **ActionFlow**, a novel policy class for robotics, suitable for learning dexterous manipulation skills while integrating geometric notions for sample efficient learning. In essence, ActionFlow is composed of two main elements: **(1)** a state-of-the-art [25] highly-expressive deep generative model (Flow Matching) [26, 27] that has been shown capable of generating high-quality samples within very small inference times, and, **(2)** an SE(3) Invariant Transformer network [20, 22] that frames a *relative positional encoding* [20, 28] based on the tokens' relative SE(3) poses (Fig. 1). Combining those components results in several interesting properties that make ActionFlow an appealing candidate for learning robot policies and, in particular, robotic manipulation from demonstrations:

Fast and accurate action sequence generation. Given the underlying Flow Matching generative model, we can generate precise action trajectories with very low inference times [29, 25].

SE(3) equivariant action generation. ActionFlow inherently preserves the tasks' spatial structure and naturally adapts the predicted actions to the observations. Given a translation or a rotation in the observations, the actions are equally transformed, thereby providing SE(3) equivariant generation. While ActionFlow's underlying transformer network is invariant, we achieve global SE(3) equivariance by applying the flow matching updates w.r.t. the actions' local frame [20, 22].

Relative Pose Aware Attention. The SE(3) Invariant Transformer allows the actions to attend to the different observation tokens based on their relative poses. This allows the system to find correlations based on the spatial relative information between the tokens and enhances the generalization to scenes where objects are arranged differently.

In summary, our main contributions are: **(a)** we investigate Flow Matching for fast and precise robotic action generation, **(b)** we introduce an SE(3) Invariant Transformer architecture for geometry-aware robot learning. Our experiments in simulated and real robot environments underline the effectiveness of both components and showcase that their combination, i.e., our proposed ActionFlow, yields accurate and fast manipulation policies while showcasing sample efficiency.

2 ActionFlow

ActionFlow policies should be fast, accurate, expressive, and sample-efficient. Therefore, they are built on two core elements: a Flow Matching-based generative model that generates action sequences quickly and a geometrically grounded transformer model capable of capturing the intricate spatial relations between observations and actions in the SE(3) space. Before introducing both components in detail, we introduce ActionFlow's observation and action space. ActionFlow relies on a geometrically grounded and highly flexible scene representation. Both observations $\boldsymbol{O} : (\boldsymbol{T}_o, \boldsymbol{F}_o)$ and actions $\boldsymbol{A} : (\boldsymbol{T}_a, \boldsymbol{F}_a)$ are represented through a sequence of SE(3) poses $\boldsymbol{T} = (T^1, \dots, T^N)$ and associated features $\boldsymbol{F} = (\boldsymbol{f}^1, \dots, \boldsymbol{f}^N)$ representing semantic information related to the specific action/observation (cf. Fig. 2). Action poses correspond to the desired target poses to be reached, while the features represent at what instant of time the target pose should be reached. Given this representation, our goal is to learn ActionFlow policies $\pi_{\boldsymbol{\theta}}(\boldsymbol{T}_a|\boldsymbol{O})$ that generate action pose sequences $\boldsymbol{T}_a$ given an observation $\boldsymbol{O}$.

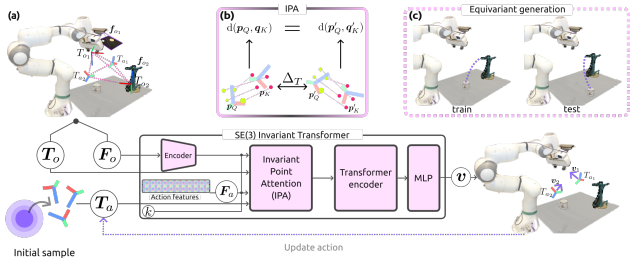

Figure 2: **Spatial Symmetries in ActionFlow**. **(a)**, Visual representation of the SE(3) Invariant Transformer mapping from observations $\boldsymbol{F_o}$ with their poses $\boldsymbol{T}_o$ & candidate actions $\boldsymbol{T}_a$, to action update vectors $\boldsymbol{v}$. **(b)** Visualization of Invariant Point Attention (IPA) which is based on generating points $\boldsymbol{p}_Q$ and $\boldsymbol{p}_K$ in the queries' and keys' local frames resulting in equivariance. **(c)** Illustrating ActionFlow's SE(3) equivariant action generation.

2.1 Flow Matching for SE(3) Action Generation

Similarly to diffusion models [30, 31], Flow Matching models generate samples by iteratively calling a learned model. Yet, as shown in [29, 25], Flow Matching models require fewer model calls, reducing the inference time for sample generation. Herein, we apply (conditioned) Flow Matching to generate N SE(3) action poses $\boldsymbol{T}_a=(T_a^1,\dots,T_a^N)\in SE(3)^N$. Similarly to [21], we adapt a common Flow Matching method (Rectified Linear Flow [29, 25, 26]) to the Lie Group SE(3). We define a decoupled flow between the rotation ($\boldsymbol{r}$) and the translation ($\boldsymbol{p}$), allowing to represent the distribution path and the vector fields independently. Rectified Linear Flow proposes representing the data point conditioned flow $\boldsymbol{\phi}_t(\boldsymbol{a}|\boldsymbol{a}_1)$ with a straight line from a noisy sample $\boldsymbol{a}_0\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$ at $t=0$ to the datapoint $\boldsymbol{a}_1\in\mathcal{D}$ at $t=1$. For our case, we define a flow $T_t=\boldsymbol{\phi}_t(T_0|T_1)$ that moves a noisy initial pose $(\boldsymbol{p}_0,\boldsymbol{r}_0)$ to a pose sampled from the dataset, i.e., $(\boldsymbol{p}_1,\boldsymbol{r}_1)\sim\mathcal{D}$, through

$$\text{Translation: } \boldsymbol{p}_t=\boldsymbol{\phi}_t(\boldsymbol{p}_0|\boldsymbol{p}_1)=t\boldsymbol{p}_1+(1-t)\boldsymbol{p}_0 \quad \text{Rotation: } \boldsymbol{r}_t=\boldsymbol{\phi}_t(\boldsymbol{r}_0|\boldsymbol{r}_1)=\boldsymbol{r}_0\text{Exp}\left(t\text{Log}(\boldsymbol{r}_0^{-1}\boldsymbol{r}_1)\right) \quad (1)$$

with Log and Exp, the logarithmic map and the exponential map respectively for the SO(3) manifold [32]. Notice that Equation (1) represents a path through the geodesic on SO(3) from $\boldsymbol{r}_0$ to $\boldsymbol{r}_1$. Given the decoupled flows, the vector fields, and in particular, the translation velocity $\dot{\boldsymbol{p}}_t\in\mathbb{R}^3$ and the rotation velocity $\dot{\boldsymbol{r}}_t\in\mathbb{R}^3$ equate to $\dot{\boldsymbol{p}}_t=\boldsymbol{r}_t^\intercal(\boldsymbol{p}_t-\boldsymbol{p}_1)/(1-t)$ & $\dot{\boldsymbol{r}}_t=(\text{Log}(\boldsymbol{r}_t^{-1}\boldsymbol{r}_1))/(1-t)$. Notice that even if rotations are represented in $\boldsymbol{r}\in SO(3)$, the velocity vector for the rotations $\dot{\boldsymbol{r}}_t\in\mathbb{R}^3$ is a 3D vector (axis-angle representation) represented in the tangent space centered around $\boldsymbol{r}_t$.

Training. Our parameterized model $(\boldsymbol{v}_{\boldsymbol{p}},\boldsymbol{v}_{\boldsymbol{r}})=\boldsymbol{v}_{\boldsymbol{\theta}}(T,\boldsymbol{O},t)$ outputs both a translation vector $\boldsymbol{v}_{\boldsymbol{p}}\in\mathbb{R}^3$ and a rotation vector $\boldsymbol{v}_{\boldsymbol{r}}\in\mathbb{R}^3$. Given a dataset $\mathcal{D}:\{T^i,\boldsymbol{O}^i\}_{i=0}^I$, the training objective is to minimize the mean-squared error between the model outputs $(\boldsymbol{v}_{\boldsymbol{p}},\boldsymbol{v}_{\boldsymbol{r}})$ and the velocity vectors $\boldsymbol{u}_t=(\dot{\boldsymbol{p}}_t,\dot{\boldsymbol{r}}_t)$.

Sampling in SE(3). To generate an action pose $T=(\boldsymbol{p},\boldsymbol{r})$, we sample a rotation $\boldsymbol{r}_0\sim\mathcal{U}(SO(3))$ and translation $\boldsymbol{p}_0\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$ and iteratively run Euler-discretization for K inference steps, i.e., $\boldsymbol{p}_{k+1}=\boldsymbol{p}_k+\boldsymbol{r}_k\boldsymbol{v}_{\boldsymbol{\theta}}(T_k,\boldsymbol{O},t)\Delta t$ & $\boldsymbol{r}_{k+1}=\boldsymbol{r}_k\text{Exp}(\Delta t\boldsymbol{v}_{\boldsymbol{\theta}}(T_k,\boldsymbol{O},t))$, with $\Delta t=1/K$ and $t=k\Delta t$.

2.2 SE(3) Invariant Transformer

As model architecture we propose an SE(3) Invariant Transformer (cf. Fig. 2). At the core of this architecture is a geometry-aware attention layer, known as Invariant Point Attention (IPA) [20, 22]. The IPA layer augments the queries, keys, and values of classical attention [33] with a set of 3D points that are generated in the local frames of the query $\boldsymbol{T}_Q$ and key $\boldsymbol{T}_K$ poses. The layer is designed such that the output is invariant to global rotations and translations (cf. Fig. 2 (b)). If we apply a transformation $\Delta_T\in SE(3)$ over both observation poses $\boldsymbol{T}_o'=\Delta_T\boldsymbol{T}_o$ and action poses $\boldsymbol{T}_a'=\Delta_T\boldsymbol{T}_a$, the network generates the same output. Moreover, with the IPA layer, the network can reason about all the relative poses between the entities in the scene. We hypothesize that the invariant and object-centric nature of the network will lead to more data-efficient policies.

3 Experimental Results

We focus on two experiments. First, we evaluate the impact of the SE(3) Invariant Transformer w.r.t. data-efficient policy learning. Second, we evaluate ActionFlow for real robot manipulation.

3.1 SE(3) Invariant Transformer Evaluation

This section evaluates the proposed SE(3) Invariant Transformer (cf. Section 2.2) w.r.t. two design choices: (**1**) Does a **multi-token representation**, in which each object is treated as a single token, enhance policy performance? (**2**) Does the **IPA** layer, which allows computing the relative poses between tokens, help find informative features to improve policy performance?

Evaluation Environment. We consider a subset of Mimicgen environments [34]. We slightly adapt the data to be compatible with our model and represent it as object T_o^w & action poses T_a^w in the world frame.

Models. We consider three ActionFlow variations: (**A.1**) The original ActionFlow (cf. Sec-

tion 2.2). (**A.2**) We eliminate the IPA layer but keep each object as an independent token. (**A.3**) We eliminate the IPA layer and represent all observations as a single token. Additionally, we consider (**B**) a Diffusion Policy (DP) model [2], and (**C**) a RNN-GMM model [35, 34].

Results. We train the models with different amounts of demonstrations $(20, 50, 200, 1000)$ for 3500 epochs and evaluate their performance in 50 randomly sampled test environments. The results in Fig. 3 reveal that the original ActionFlow consistently outperforms the variations and baselines across the tasks, specifically in low demonstration regimes. This indicates that IPA is beneficial for learning policies in a sample-efficient way, while with large datasets, all models appear to converge to similar performances. We also observe that representing the observations with multiple tokens shows performance benefits compared to representing the whole observation as a single token. Moreover, the Diffusion Policy (DP) baseline performs very similar to our model ablation A.3 with the single token. One potential explanation for this finding is that the DP's underlying transformer model also flattens the observations of each timestep into a single token, thereby resembling a similar network architecture as the single token ablation.

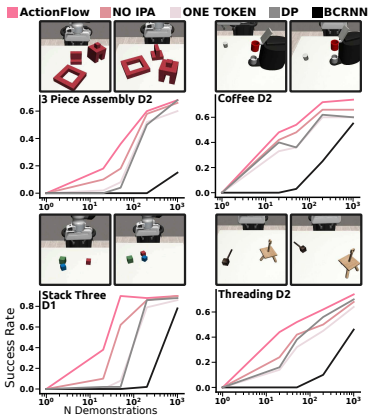

Figure 3: **Success rate of models trained on different number of demonstrations** $(20, 50, 200, 1000)$ on a subset of Mimicgen tasks [34].

3.2 Evaluating Equivariance - Real Robot Mug Hanging Experiment

We evaluate ActionFlow on the task of placing a mug onto a hanger.

Setup. The experimental platform consists of a 7DoF Franka Panda with a RealSense at its end-effector (cf. Fig. 1). The end-effector is described by its pose T, and the features $\boldsymbol{f}$ contain the encoded RGB images (using a ResNet18 [36]) and the gripper's opening width. Another observation is the point cloud of the hanger, which is obtained by the camera's depth readings. The point cloud features are obtained using a PointNetEncoder [37]. Both encoders are trained from scratch. For policy training, we collect 50 demonstrations using variations as shown in Fig. 4. Notably, the demonstrations only include slight variations of the mug poses, while the hanger always stays in the same pose.

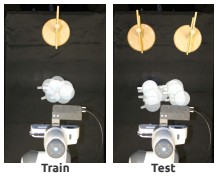

Figure 4: Initial configurations for train and test evaluations.

Initialization	Success Rate
Train	9/10
Test	7/10

Figure 5: Mug hanging results.

Results. The results in Fig. 5 reveal that the ActionFlow policy achieves high success rates of 90% upon evaluating in similar scenarios as those encountered during training. The evaluation in previously unseen test scenarios (cf. Fig. 4) shows that our policy can still handle these novel test scenarios well, achieving 70% success. This good generalization to the previously unseen testing scenarios highlights ActionFlow's equivariance properties. Moreover, the policies run online in real-time as action generation takes $0.04\,\mathrm{s}$ on an NVIDIA RTX 3090.

Figure 6: Successful ActionFlow policy rollout.

4 Conclusion

We presented ActionFlow, a new policy class for robot learning from demonstrations. On the representation level, ActionFlow consists of an SE(3) Invariant Transformer equipped with geometry-aware Invariant Point Attention. Actions are generated using Flow Matching, a new generative model capable of obtaining high-quality samples with low inference times. The resulting policies are fast, represent both actions and observations in one common space, and yield SE(3) equivariant action generation. Our experiments underline the effectiveness of ActionFlow's individual components and demonstrate its capabilities for efficiently solving real robotic manipulation tasks.

Acknowledgments

This work has received funding from the German Research Foundation (DFG) Emmy Noether Programme (CH 2676/1-1), the EU's Horizon Europe projects MANiBOT (Grant no.: 101120823) and ARISE (Grant no.: 101135959), the AICO grant by the Nexplore/Hochtief Collaboration with TU Darmstadt and from the German Federal Ministry of Education and Research (BMBF) project IKIDA (01IS20045).

References

[1] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. In K. Liu, D. Kulic, and J. Ichnowski, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1199–1210. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/zhu23a.html.

[2] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[3] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[4] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[5] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, S. K. S. Ghasemipour, C. Finn, and A. Wahid. ALOHA unleashed: A simple recipe for robot dexterity. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=gvdXE7ikHI.

[6] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[7] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki. Act3d: Infinite resolution action detection transformer for robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023.

[8] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023.

[9] V. Vosylius, Y. Seo, J. Uruç, and S. James. Render and diffuse: Aligning image and action spaces for diffusion-based behaviour cloning. *arXiv preprint arXiv:2405.18196*, 2024.

[10] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[11] S. James, K. Wada, T. Laidlow, and A. J. Davison. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13739–13748, 2022.

[12] J. Yang, C. Deng, J. Wu, R. Antonova, L. Guibas, and J. Bohg. Equivact: Sim (3)-equivariant visuomotor policies beyond rigid object manipulation. *arXiv preprint arXiv:2310.16050*, 2023.

[13] H. Ryu, J. Kim, J. Chang, H. S. Ahn, J. Seo, T. Kim, J. Choi, and R. Horowitz. Diffusion-edfs: Bi-equivariant denoising generative modeling on se (3) for visual robotic manipulation. *arXiv preprint arXiv:2309.02685*, 2023.

[14] H. Huang, D. Wang, A. Tangri, R. Walters, and R. Platt. Leveraging symmetries in pick and place. *The International Journal of Robotics Research*, page 02783649231225775, 2024.

[15] C. Gao, Z. Xue, S. Deng, T. Liang, S. Yang, L. Shao, and H. Xu. Riemann: Near real-time se (3)-equivariant robot manipulation without point cloud segmentation. *arXiv preprint arXiv:2403.19460*, 2024.

[16] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022.

[17] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5923–5930. IEEE, 2023.

[18] S. Calinon. A tutorial on task-parameterized movement learning and retrieval. *Intelligent service robotics*, 9:1–29, 2016.

[19] C. R. Dreher, M. Wächter, and T. Asfour. Learning object-action relations from bimanual human demonstration using graph networks. *IEEE Robotics and Automation Letters*, 5(1): 187–194, 2019.

[20] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[21] J. Yim, A. Campbell, A. Y. Foong, M. Gastegger, J. Jiménez-Luna, S. Lewis, V. G. Satorras, B. S. Veeling, R. Barzilay, T. Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023.

[22] J. Yim, B. L. Trippe, V. De Bortoli, E. Mathieu, A. Doucet, R. Barzilay, and T. Jaakkola. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023.

[23] B. Jing, B. Berger, and T. Jaakkola. Alphafold meets flow matching for generating protein ensembles. *arXiv preprint arXiv:2402.04845*, 2024.

[24] G. Huguet, J. Vuckovic, K. Fatras, E. Thibodeau-Laufer, P. Lemos, R. Islam, C.-H. Liu, J. Rector-Brooks, T. Akhound-Sadegh, M. Bronstein, et al. Sequence-augmented se (3)-flow matching for conditional protein backbone generation. *arXiv preprint arXiv:2405.20313*, 2024.

[25] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024.

[26] R. T. Chen, M. FAIR, and Y. Lipman. Flow matching on general geometries.

[27] R. T. Chen and Y. Lipman. Flow matching on general geometries. In *The Twelfth International Conference on Learning Representations*, 2024.

[28] A. Liutkus, O. Cıfka, S.-L. Wu, U. Simsekli, Y.-H. Yang, and G. Richard. Relative positional encoding for transformers with linear complexity. In *International Conference on Machine Learning*, pages 7067–7079. PMLR, 2021.

[29] X. Liu, C. Gong, and Q. Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[30] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[31] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[32] J. Sola, J. Deray, and D. Atchuthan. A micro lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[34] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

[35] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, pages 1678–1690. PMLR, 2022.

[36] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi:10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

[37] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. J. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12200–12209, October 2021.