2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Flow-Enabled Generalization to Human Demonstrations in Few-Shot Imitation Learning

Runze Tang[1] and Penny Sweetser[1]

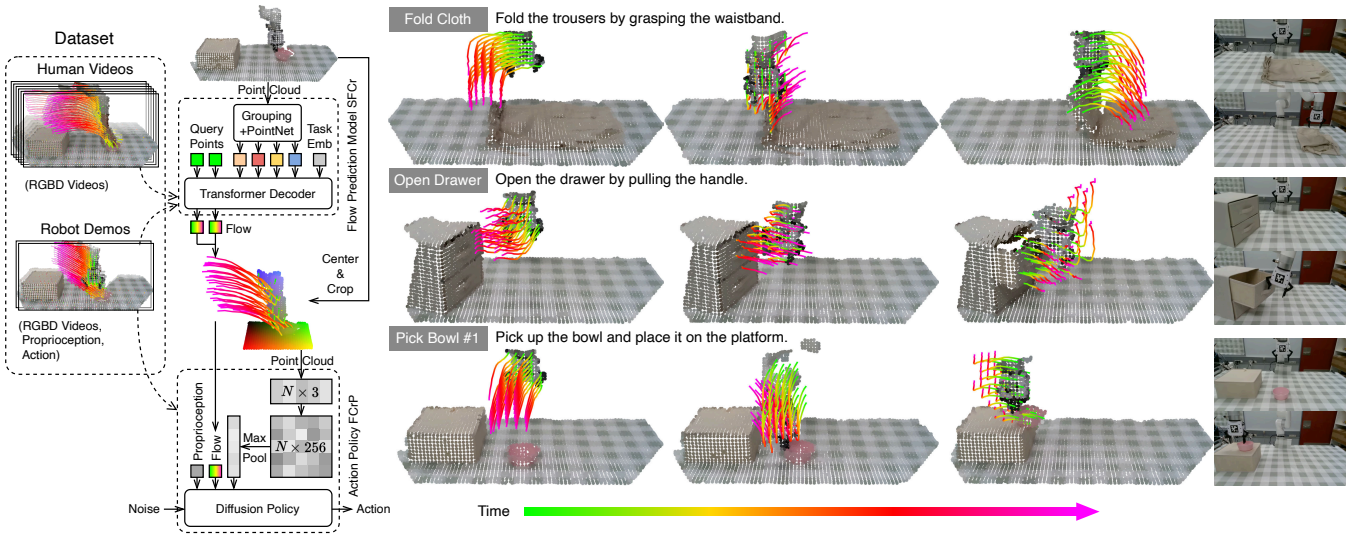


Fig. 1: **Overview of our work.** Left: We use 30 human videos and 10 robot demonstrations to train the cross-embodiment flow prediction model SFCr and the flow-conditioned policy FCrP. Right: The point cloud observation from a single third-person-view camera and the predicted flow during execution. The images on the right are the beginning and success states of each task.

***Abstract*— Imitation Learning (IL) enables robots to learn complex skills from demonstrations without explicit task modeling, but it typically requires large amounts of demonstrations, creating significant collection costs. Prior work has investigated using flow as an intermediate representation to enable the use of human videos as a substitute, thereby reducing the amount of required robot demonstrations. However, most prior work has focused on the flow, either on the object or on specific points of the robot/hand, which cannot describe the motion of interaction. Meanwhile, relying on flow to achieve generalization to scenarios observed only in human videos remains limited, as flow alone cannot capture precise motion details. Furthermore, conditioning on scene observation to produce precise actions may cause the flow-conditioned policy to overfit to training tasks and weaken the generalization indicated by the flow. To address these gaps, we propose SFCrP, which includes a Scene Flow prediction model for Cross-embodiment learning (SFCr) and a Flow and Cropped point cloud conditioned Policy (FCrP). SFCr learns from both robot and human videos and predicts any point trajectories. FCrP follows the general flow motion and adjusts the action based on observations for precision tasks. Our method outperforms SOTA baselines across various real-world task settings, while also exhibiting strong spatial and instance generalization to scenarios seen only in human videos.**

I. INTRODUCTION

As a fundamental subclass of Imitation Learning (IL), behavior cloning (BC) is a widely adopted strategy for offline policy learning from demonstrations [1]. While BC can address tasks with complex dynamics, it usually requires dozens of demonstrations for simple tasks [2–4], and thousands to achieve robust generalization [5, 6]. However, collecting large-scale datasets is often cost-prohibitive given the specialized equipment for each demonstrator [5–7]. Therefore, many studies focus on using videos of human manipulation as a substitute for robot demonstrations [8–10].

With 3D spatial information, point cloud-based methods generally have better data efficiency and generalization capabilities than image-based methods [11]. The image convolution features of the robot and human hand regions are inevitably different. Therefore, distribution adaptation methods are required to align image representations [9, 12]. While in a point cloud, the robot or human hand is represented by points in the air. Combined with segmentation methods, point clouds become a well-suited representation for cross-embodiment learning [10]. Flow, that is, the trajectories of points, has been confirmed to be an effective representation that describes motions to bridge human and robot demonstrations [9, 10, 13–15]. However, most prior work has focused on flow on the object or robot arm solely to achieve cross-embodiment learning. Focusing solely on the object's motion often overlooks the robot's pre-grasp motion [10]. Conversely, only considering the motion of the robot omits the details of the interaction with the object [9]. In this work, we present a point cloud-based method that enables any-point

[1]The Australian National University {runze.tang, penny.kyburz}@anu.edu.au

flow prediction and cross-embodiment learning.

For point cloud scene perception, using a Transformer model to process the point cloud represented by tokens could effectively capture the spatial information of objects in the scene [2, 4], but lacks point-level details. Point-level perception methods, such as DP3 [16], often struggle in identifying scene changes, resulting in limited generalization [17]. Moreover, diffusion policy is shown to tend to overfit the training tasks and lacks generalization ability [18, 19]. To address these issues, we use flow as an intermediate representation to bridge a Transformer-based flow prediction model and a flow-conditioned action policy with point-level perception. We crop the point cloud observation of the policy and balance the reliance between the point cloud and the flow to reduce overfitting. In summary, our main contributions are:

- A flow prediction model SFCr that predicts any point trajectories with high cross-embodiment data efficiency.
- A flow and cropped point cloud conditioned policy FCrP that achieves spatial and instance generalization.
- Comprehensive experiments that demonstrate flow is a representation that can (1) bridge group-level spatial relationship perception and point-level detail recognition, (2) align robot demonstrations and human videos, and (3) significantly reduce overfitting of diffusion policy to achieve stable generalization.

Furthermore, we discuss our method, including the underlying mechanisms, and the aforementioned issues in the broader field through four research questions: RQ1: How effectively does segmentation narrow the cross-embodiment appearance gap? RQ2: Through what mechanism does cropping the point cloud enhance the policy performance in precision tasks? RQ3: How does conditioning on flow enhance policy generalization? RQ4: How effective is balancing the reliance in alleviating overfitting of the diffusion policy?

II. Related Work

Behavior cloning in imitation learning is a framework where agents acquire skills by observing and replicating expert behaviors [20, 21]. In behavior cloning for robot manipulation, RGB images [3, 22–25] and point clouds [2, 4, 16, 24, 26–28] are two typical types of observations. Some previous work has combined RGB image features and point clouds to obtain both pixel-level perception and depth information [17, 29–32], but it required multiple cameras to mitigate the impact of convolution background blending and view-dependent features. In this work, we focus on the point cloud obtained from a single third-person view camera.

To enhance performance, many papers went beyond using manipulation task data. Some leveraged large-scale image or point cloud datasets for pretraining [4, 5, 31, 33–36], while others directly used pre-trained visual models as feature extractors [6, 14, 25, 28–30, 37, 38]. Alternatively, some papers relied on in-domain manipulation task data but trained on more diverse forms, such as data collected from different robot embodiments [5, 39–41] or human videos [42] through visual embeddings [8, 34, 35, 43, 44], object flow [10, 13–15, 45], or human hand trajectory [9, 12, 46–48].

Existing approaches to predict flow can be classified into three main categories: predicting flow on the object [10, 13–15, 45, 49, 50], predicting the flow on the robot arm [9, 12, 39, 51], and predicting trajectories of any points in the scene [52–54]. Considering only the flow of the object could easily achieve cross-embodiment learning, but it lacks information about the robot motion, especially before grasping [10]. While the trajectory of points only on the robot cannot capture the interaction with objects. Although predicting the flow of the entire scene could capture any motion in the scene, it has a larger cross-embodiment gap [13]. Prior work has shown that using large-scale video datasets with diverse embodiments could address the cross-embodiment gap in predicting any-point trajectories [53]. In this paper, we address this gap with a small-scale dataset using a well-designed flow prediction method.

The choice of observation can be critical for the action policy [55]. Relying solely on the predicted flow as observation, some researchers compute actions in a heuristic way [9, 10, 15, 45, 49, 50]. In contrast, other methods used a policy network to produce actions conditioned on the predicted flow and scene observation [12, 13, 52, 53]. Policies that use both flow and scene observation as conditions are capable of performing more precise actions and correcting potential inaccuracies in the input flow. In comparison, heuristic methods highly rely on the accuracy of the predicted flow. However, the scene observation can undermine the generalization ability of the action policy. In contrast, many flow-conditioned heuristic action policies that are not conditioned on the scene observation show a stronger generalization ability [9, 10, 15, 45, 49].

Most of the flow-conditioned policy models are actually conditioned on the flow feature vector rather than the raw flow [12, 13, 52]. They use flow as a training target of the flow prediction model, rather than a representation that truly bridges the flow prediction and the action policy. This could further mitigate the failure caused by inaccurate flow, but could lead to potential overfitting to the training task and a strong binding between the flow model and policy model. Our approach uses the raw flow and a local cropped point cloud as the condition for the action policy. In this way, we ensure heuristic-level generalization based on the flow and obtain enough observations for precision-demanding tasks.

III. Approach

Our method consists of two parts: the flow prediction model SFCr and the flow-conditioned policy FCrP. SFCr can learn from human videos, while FCrP is trained solely on robot demonstrations. Each robot demonstration includes an RGBD video, the corresponding robot proprioception, and the associated actions. We use the position of the gripper and its two fingers as proprioception data $g \in \mathbb{R}^{3\times3}$. The human demonstrations contain only the RGBD videos. We first build the raw point cloud of each RGBD image, then use voxel downsampling to reduce the number of points. To obtain the ground truth flow, we use CoTracker [56] to track grid-sampled query points in the RGB video and map them to the

raw point cloud to obtain 3D point trajectories $F_{0:T} \in \mathbb{R}^{T\times 3}$ to form the flow $\mathcal{F}$ of the whole scene. To segment the robot and human hand, we apply FastSam [57] to each frame of the RGB videos. We segment human videos using a language prompt at the first frame and bounding boxes for the rest. The robot video is segmented using bounding boxes based on the April tag on the robot gripper as prompts.

A. SFCr: Cross-Embodiment Scene Flow Prediction Model

The general architecture of our flow prediction network is shown in Fig. 1. We use a Transformer decoder [58] as its main component. The input tokens include point cloud tokens, task embedding, and flow query tokens. Each point cloud token is the PointNet [59] features of a local group of points, adding the spatial encoding. We sample the center x_0 of each group by farthest point sampling and select points near the center to form the group $x_{0:k}$. Each flow query token is the spatial encoding of the starting point F_0 of the corresponding trajectory $F_{0:T}$. The point cloud group center x_0 and the flow query points F_0 share the same spatial encoder. The output of each flow query token is fed into a shared multi-layer perceptron (MLP) to obtain the predicted trajectory. By using the Transformer decoder to process all these inputs, we expect the model to match the task embedding and the query points with the point cloud group tokens, learn a rough motion for each group, align across flow tokens, and finally provide a refined query point motion represented by trajectories.

We evaluate the predicted flow based on the absolute position. But for each trajectory point, we use the position related to its query point as the predicting target $F_i - F_0$ and minimize the L1-norm loss. Empirically, this representation has a lower prediction error than position related to the previous point $F_i - F_{i-1}$ [10] or absolute position F_i.

To prevent the model from overfitting to the spatial distribution of query points, our flow prediction model is trained on a trajectory subset of $N_q = 64$ query points for each point cloud observation. However, more than half of the points in the scene remain static. Randomly sampling from all trajectories will result in an imbalanced distribution of trajectory lengths. Therefore, for each sample, we first sample a moving ratio $p_m \sim \mathcal{U}(0, 1)$, then select $p_m N_q$ points whose future trajectory is not static and $(1 - p_m)N_q$ static points as query points. We determine whether a query point is static or not based on the width of a trajectory, which is defined as $\max_{i,j\in[0,T)} \|F_i - F_j\|_2$, the largest distance between any two points F_i, F_j on trajectory F. This metric is more noise-robust than the accumulated displacement. During execution, the query points are selected via grid-based sampling from the box-shaped cropped point cloud centered at the robot gripper. We feed all the in-box query points in at once to maintain consistency among trajectories.

To minimize the visual difference between the robot and the hand and enable cross-embodiment flow prediction, we segment the image to obtain robot/hand segmentation. We replace the point cloud color in the robot/hand region with (1,0,1) and add a dimension after the XYZRGB values to indicate whether a point belongs to the robot/hand or not. Moreover, we randomly remove a fraction of point cloud group tokens where most points are marked as robot/hand. This aims to train the flow prediction model not to remember the exact shape of the robot/hand, but to make inferences based on their approximate position.

B. FCrP: Flow and Croped Point Cloud Conditioned Policy

Our diffusion-based [3, 60] action policy produces actions by progressive denoising, conditioned [61] on the predicted flow $\mathcal{F}$ and state observations $\{s_f, s_{t-1}, s_t\}$, as Fig. 1 shows. The observation horizon includes three states: (1) the flow state s_f, at which the flow $\mathcal{F}$ is predicted, (2) the state before the current state s_{t-1}, and (3) the current state s_t, where $t \geq f$. Each state observation includes the local cropped point cloud X and proprioception data g. We use the DP3 [16] encoder as the point cloud perception model, which first calculates the features of each point using shared MLPs, then applies max pooling to the feature dimension, followed by another MLP layer to obtain a compact representation.

Instead of using the point cloud observation of the whole scene, we crop the point cloud observation X to keep only a box-shaped region around the robot gripper and center it with the robot gripper as the origin for each state $\{s_f, s_{t-1}, s_t\}$. For the proprioception points $\{g_f, g_{t-1}, g_t\}$ at each state and the flow $\mathcal{F}$ at the flow state s_f, we center them around the gripper position at the flow state s_f, to keep the related spatial information within the observation horizon. In this way, the observation of our action policy is fully localized with the robot gripper as origin without any absolute spatial information, which enables the generalization following the flow and action adjustment conditioned on the local point cloud. In execution, we select query points within the box-shaped cropped point cloud for flow prediction.

To decouple our action policy from the flow prediction network in terms of inference frequency, we introduce a flow-state-action alignment mechanism. Our action policy predicts a sequence of actions starting from the flow state s_f with an execution mask that matches the motion between the flow state s_f and the current state s_t with the flow $\mathcal{F}$. The execution mask indicates which actions were performed from the flow state s_f to reach the current state s_t, and which actions should be performed next. This enables the prediction of an arbitrary number of actions from the same predicted flow. Predicting actions starting from the flow state temporally aligns the motion in flow and the actions, enabling our action policy to produce actions that follow the same motion as the flow to enable generalization.

Apart from enabling parallel inference and asynchronous flow condition updating by conditioning on the previous flow and replacing it whenever the new one is predicted, our flow-state-action alignment mechanism could also enable heuristic flow prediction skipping at error-prone states. For example, when the gripper is close to the object before grasping, if the flow prediction model mistakenly thinks that the robot has already grasped the object and gives a flow with an upward direction, the action policy may fail to grasp the object.

TABLE I: Task Properties

Key Property	Fold Cloth	Open Drawer	Pick Bowl
Multi-Task	No	No	Yes
Object Type	Deformable	Articulated	Rigid
Include Grasp Action	Yes	No	Yes
Action Accuracy Req.	Medium	High	Low

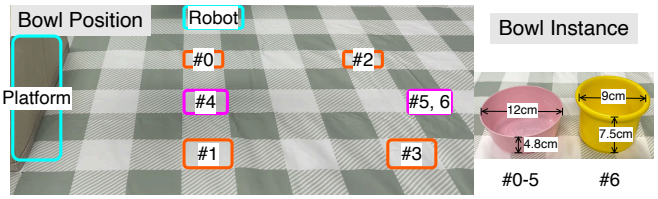


Fig. 2: The bowl position (warm color rectangles) and instances of Pick Bowl tasks. There are no robot demonstrations for #4-6.

Many prior approaches use the predicted flow embeddings as the condition of the policy network. In this way, the policy network becomes more robust to the inaccurately predicted flow. To achieve a similar goal using the raw flow, we let a trained flow prediction model predict the flow for each robot demonstration, to replace the ground-truth flow during action policy training. Empirically, we do not observe a significant performance drop using different flow prediction models for predicted flow generation and execution. Moreover, we balance the condition reliance between the point cloud and the flow to reduce overfitting caused by point cloud conditioning. We randomly mask the point cloud (MP) by replacing the entire point cloud with zero with a probability of 0.5. Thus, the policy is forced to rely more on the flow.

IV. Real-World Evaluation

Fig. 1 and Table I list the settings and description of the real-world tasks. Fig. 2 shows seven versions #0-6 of the Pick Bowl task. For each task, we collect 10 robot demonstrations (R10) and 30 human videos (H30) for training, except Pick Bowl #4-6, which have 30 human videos only.

We compare the flow prediction accuracy of ours with ScaleFlow-L [10], which is a 3D trajectory prediction model based on PointNeXt [62] and VAE [63, 64]. We compare the manipulation task success rate of our method with DP3 [16], RISE [2], and SUGAR [4]. DP3 uses a well-designed point cloud encoder followed by a diffusion policy [3] to predict actions. RISE uses spatial convolution [65] followed by a Transformer [58] to extract point cloud features for the final diffusion policy. SUGAR samples and groups the point cloud, extracting the group features as tokens for Transformer calculations. The decoder of SUGAR was designed for keypose action prediction. We adopted the action head of ACT [22] that is also based on the Transformer decoder, to enable SUGAR for micro-step action prediction. SUGAR pretrains the Transformer encoder on five pretraining tasks with four datasets (∼940K samples). We evaluate SUGAR pre-trained using the ensemble of four datasets and from scratch. We also assess the ablated variants of our method, where the flow prediction network does not have robot/hand segmentation (w/o SG), the action policy has no point cloud observation (w/o PC), is not trained on the predicted flows (w/o PF), and does not mask the point cloud (w/o MP).

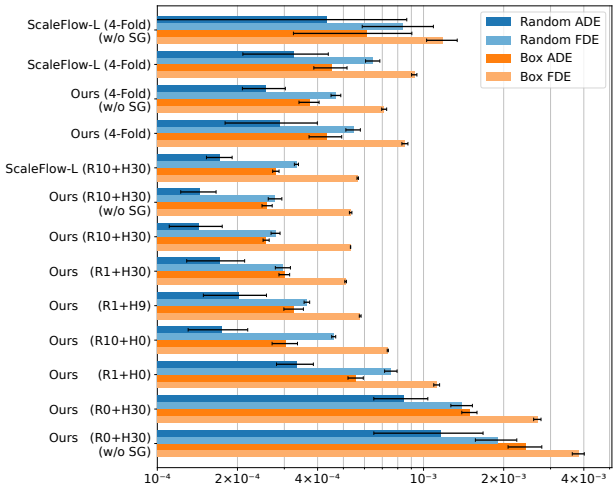


Fig. 3: Logarithmic scale flow ADE and FDE over five seeds.

A. Flow Prediction Evaluation

Our predicted flow during execution shows correct motion information. In the middle column of the flow plots shown in Fig. 1, we can observe that the trajectory of points on the object remains static when the robot is approaching (green to red) and then starts moving with the same motion as the robot (red to magenta). Fig. 3 shows the average displacement error (ADE) and final displacement error (FDE) of the predicted flow of randomly selected query points (Random) and in-box around the robot gripper (Box) on the test robot demonstrations. The 4-Fold validation involves sequentially excluding robot demonstrations from one of the four Pick Bowl tasks (#0–3) in each fold, evaluating the model on the task that does not have robot demonstrations. Compared to ScaleFlow-L [10], our method shows a lower error in both the full dataset (R10+H30) and the 4-Fold settings. Moreover, our method trained with R10+H0, R1+H9, and R1+H30 achieves similarly low errors, comparable to using the full dataset (R10+H30). These results underscore the high cross-embodiment data efficiency of our method.

RQ1: How effectively does segmentation narrow the cross-embodiment appearance gap? When no robot data is available (R0+H30), our method shows a substantial error drop compared to the version not using robot/hand segmentation (w/o SG). We note that the higher flow error of our method without robot data (R0+H30) is primarily due to human videos having a higher speed and longer flow, rather than incorrect motion in the predicted flow. However, the difference between with and without segmentation is not notable when robot data are available (R10+H30 and 4-Fold). We attribute Ours (4-Fold) having a slightly higher error than without segmentation (w/o SG) to the information loss caused by the segmentation-based point cloud removal augmentation. In summary, (1) our flow prediction model could generalize to scenarios only seen in human videos (4-Fold) without the need for segmentation, (2) with segmentation narrowing the appearance gap, our model could predict the flow of unseen embodiment with correct motion.

TABLE II: **Success rate with full dataset (R10+H30).** The seen average success rate does not include Pick Bowl #4-6. We conduct 20 trials for each task with different seeds, except for tasks that are almost impossible to complete, which have 10 trials. The figures in parentheses are the number of failures in which the first stage was completed but failed the final stage.

Method	Seen	Fold	Open	Pick Bowl						
	Succ %.	Cloth	Drawer	#0	#1	#2	#3	#4	#5	#6
DP3 [16]	74.17	11/20	15/20	12/20 (+5)	**20/20**	**20/20**	11/20	0/10	0/10	0/10
RISE [2]	50.00	3/20 (+14)	9/20 (+2)	3/20 (+15)	17/20 (+2)	10/20 (+3)	18/20	0/10	0/10	0/10
SUGAR [4]	75.83	7/20 (+1)	8/20 (+1)	19/20	**20/20**	19/20	18/20	**19/20** (+1)	9/20	0/10
SUGAR (pre-trained)	88.33	**18/20**	9/20	**20/20**	**20/20**	**20/20**	19/20	**19/20** (+1)	8/20	0/10
Ours (w/o PC)	63.33	2/20 (+15)	1/20	**20/20**	19/20 (+1)	16/20	18/20 (+2)	**19/20**	17/20	18/20
Ours (w/o PF&MP)	90.00	13/20 (+7)	**19/20** (+1)	**20/20**	19/20	18/20	19/20	**19/20**	17/20	13/20 (+1)
Ours	**96.67**	**18/20** (+1)	18/20 (+1)*	**20/20**	**20/20**	**20/20**	**20/20**	**19/20** (+1)	**20/20**	**20/20**

* This task requires high action precision, which highly relies on the point cloud observation. We did not apply the MP augmentation.

TABLE III: **Success rate of Pick Bowl #0-3 with limited demonstrations.** We conduct 10 trials for each task with different seeds. The figures in parentheses are the number of failures in which the task was completed halfway but failed at the final stage.

Method	# Demo	Pick Bowl				Avg.
	per Task	#0	#1	#2	#3	
DP3 [16]	R1+H0	1 (+3)	0 (+3)	3	0	10%
RISE [2]	R1+H0	0	0	0	0	0%
SUGAR [4]	R1+H0	7 (+2)	0 (+1)	6	3	40%
Ours (w/o MP)	R1+H0	6	6 (+1)	4	2	45%
Ours (w/o MP)	R1+H30	7	5	6	4	55%
Ours	R1+H0	**9** (+1)	2 (+5)	8	**9**	70%
Ours	R1+H30	7 (+2)	**6** (+1)	**10**	7	**75%**

TABLE IV: **First stage success rate of Open Drawer.** The first-try and retry success rate of the first stage of the Open Drawer task, that is, hooking the drawer handle.

Method	# First-Try Success	# Retry Success	First-try Succ %.	Retry Succ %.
DP3 [16]	11	4	55.0	44.4
RISE [2]	10	1	50.0	10.0
SUGAR [4]	9	0	45.0	0.0
SUGAR (pre-trained)	9	0	45.0	0.0
Ours (w/o PC)	0	1	0.0	5.0
Ours (w/o PF&MP)	17	3	**85.0**	**100.0**
Ours (w/o MP)	17	2	**85.0**	66.7

B. Real-World Robot Manipulation

Table II lists the success rate of all tasks using the full dataset (R10+H30), except for Pick Bowl #4-6 tasks (R0+H30). Our method achieves the highest success rate compared to baseline methods across all tasks, and demonstrates strong spatial and instance generalization to scenarios that are only seen in human videos (Pick Bowl #4-6).

Table III lists the success rate of Pick Bowl #0-3 tasks with limited robot demonstrations. The corresponding flow prediction error is shown in Fig. 3. Our method achieves a 70% average success rate with only one robot demonstration per task. For flow prediction, our training flow sampling method exploits the trajectories even in a single demonstration to increase the amount of training samples indirectly. For our policy network, the conditioned flow guides the approximate motion when the gripper is far from the target object. Moreover, the cropped point cloud when the robot gripper approaches the bowl in Pick Bowl #0 and #1 is very similar, which also holds for #2 and #3. The similarity of observations across different tasks allows the policy to learn a shared representation. All of these enable our method to achieve high data efficiency. Moreover, Fig. 3 shows that the in-box flow prediction error of R1+H0 is much higher than R1+H30, while in Table III the success rate difference between R1+H0 and R1+H30 is not significant. This suggests that our action policy is robust to imprecise flow, due to using predicted flow for training.

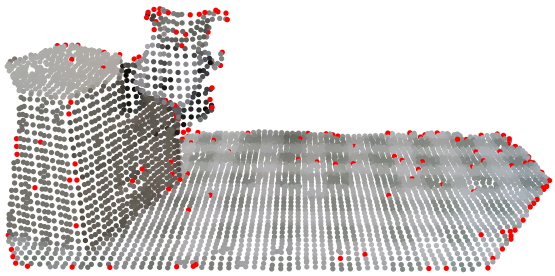
C. Failure Modes Analyze

DP3 shows a high success rate on single tasks but struggles to distinguish between the bowl positions in Pick Bowl #0-3 (Table II), resulting in occasionally moving to incorrect positions that correspond to another Pick Bowl task. This issue becomes more severe with fewer demonstrations (Table III).

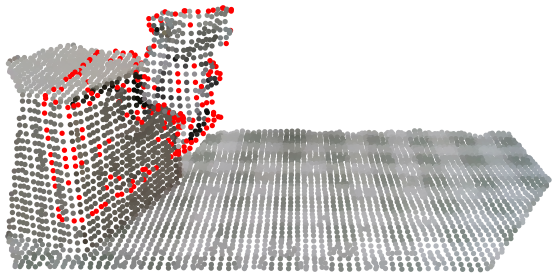
RISE, which also uses the diffusion policy as DP3, does not experience similar failures in Pick Bowl #0-3. However, both RISE and DP3 failed to show generalization in #4-6 but consistently moved to the bowl's position as in training. Furthermore, RISE occasionally triggers safety violations by crashing into the table in the Pick Bowl and Fold Cloth tasks. RISE also tends to lift the bowl with insufficient height, causing failures in Pick Bowl #0. We attribute these issues to the insufficient number of demonstrations for RISE, considering that the original work employed 50 demonstrations per task.

SUGAR, using a Transformer decoder as the action head, shows good spatial generalization in Pick Bowl #4,5 even without pretraining. The pretraining of SUGAR increases the general success rate, especially for the deformable object task Fold Cloth. However, SUGAR consistently struggles in the Open Drawer task, which requires a detailed point cloud perception to produce accurate actions. Table IV shows the first-stage success rate of the Open Drawer task, where RISE and SUGAR have a lower retry success rate than Ours and DP3, which have point-level perception.

Ours (w/o PC) consistently fails to hook the drawer handle in the Open Drawer task and fails put the trousers down in the Fold Cloth task. Ours (w/o PC) also occasionally triggers safety violations by colliding with the table. These issues highlight the importance of incorporating point cloud observations to adjust actions, rather than relying solely on the flow. Ours (w/o PF&MP) has point cloud observation as a condition. Therefore, it shows a high success rate in the precision-demanding tasks: Open Drawer and Fold Cloth. However, we note a drop in the success rate in Pick Bowl #4-6 caused by moving to the bowl positions as in training. This training-task overfitting is also observed in DP3 and RISE. Nonetheless, our method exhibits stronger generalization as a result of following the general motion provided by the flow.



(a) DP3 (Full Scene Point Cloud)



(b) Ours (Local Cropped Point Cloud)

Fig. 4: Max-pooling referenced points (red) in Open Drawer.

V. Discussion

In this section, we discuss the underlying mechanisms of our method and how they address the issues in the prior work by answering the following research questions.

RQ2: Through what mechanism does cropping the point cloud enhance the policy performance in precision tasks? Compared to the DP3 point cloud encoder that provides point-level perception, RISE and SUGAR's group-level method is coarser-grained. Although this improves their overall scene understanding, they remain incapable of discerning small changes, which prevents them from completing tasks that require more than just coarse-level actions, such as Open Drawer. DP3 encoder uses max-pooling to extract features from the referencing points, providing precise point-level descriptions of the scene. However, selecting referencing points from the point cloud of the entire scene results in sparse, redundant, or even uninformative points. In Pick Bowl tasks, sometimes there are even no referencing points on the bowl. Fig. 4 compares the referencing points chosen from the whole scene and a local cropped point cloud. This explains why the original work of DP3 crops the table point cloud. The cropped point cloud has fewer points, resulting in more concentrated referencing points. Although the DP3 encoder still tends to select edge points as reference points in the cropped point cloud, the internal reference points become more densely distributed. The more reference points on the target object, the better its position and shape can be represented, especially for a noisy point cloud.

RQ3: How does conditioning on flow enhance policy generalization? Diffusion policy tends to overfit the training tasks [18, 19]. Therefore, DP3 and RISE always move to a position associated with the training tasks and fail to generalize to unseen bowl positions in Pick Bowl #4-6. The DP3 encoder with sparse referencing points is insensitive to variations in the scene, producing similar embeddings despite changes in the position of the bowl. This aggravates the overfitting of the diffusion policy. As a result, DP3 occasionally moves to a wrong position that is associated with another training task in Pick Bowl #0-3, while RISE does not. Although our action policy also employs the diffusion policy, enabled by the flow, our method achieves both spatial and instance generalization. Our action policy without point cloud observation (w/o PC) shows generalization in Pick Bowl #4-6 following the flow condition. These findings suggest that, rather than treating the flow as a dense label-like condition, the action is calculated based on the flow to follow the same motion, thereby achieving the generalization.

RQ4: How effective is balancing the reliance in alleviating overfitting of the diffusion policy? The ablated version of our flow-condition policy with point cloud observation but without PF and MP (Ours w/o PF&MP) tends to move towards an incorrect bowl position corresponding to the training Pick Bowl tasks. While the ablated version with PF but without point cloud observation (Ours w/o PC) and the full version of our action policy that uses MP do not have such an overfitting problem. These results indicate that (1) training the policy with predicted flow (PF) does not introduce notable overfitting, (2) the primary cause of the overfitting is the reliance on point cloud observations. Therefore, it is necessary to undermine the point cloud by random masking (MP), thereby balancing the reliance between the point cloud and the flow to reduce overfitting.

In summary, for the policy network, it is essential to (1) leverage flow to guide the motion for generalization, (2) apply centering and cropping to the point cloud to enable fine-grained action adjustment based on the concentrated observation, and (3) randomly mask out the point cloud to achieve a balanced conditioning and reduced overfitting.

VI. Conclusion

We propose a cross-embodiment scene flow prediction model SFCr, which could generalize to unseen embodiments with segmentation and achieve high cross-embodiment data efficiency without segmentation. We propose a flow and cropped point cloud conditioned action policy, FCrP, that achieves high data efficiency and both spatial and instance generalization. The overall system SFCrP reduces the IL data requirement, making it possible to generalize to scenarios only seen in human videos. We empirically analyze group-level and point-level perception of point clouds, as well as the source of diffusion policy overfitting. We thoroughly ablated our method to illustrate how our method addresses the issues in the prior work, providing valuable insights for future research. However, we did not consider the flow-length variations caused by demonstrations having different speeds and motions. We train the flow prediction model SFCr using random query points, but execute it with in-box queries. The resulting difference in query point distribution causes an increase in the flow prediction error. Moreover, we mask the point cloud to balance the condition, while this method could reduce the policy performance on precision tasks. These problems remain open challenges for future research.

ACKNOWLEDGMENT

We would like to thank the Robotics@ANUComputing group for providing the experimental equipment and workspace. We also thank Hanna Kurniawati, Miaomiao Liu, and Rahul Shome for their helpful feedback and valuable discussions.

REFERENCES

[1] Maryam Zare, Parham M Kebria, Abbas Khosravi, and Saeid Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Transactions on Cybernetics*, 2024.

[2] Chenxi Wang, Hongjie Fang, Hao-Shu Fang, and Cewu Lu. Rise: 3d perception makes real-world robot imitation simple and effective. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2870–2877. IEEE, 2024.

[3] 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*, page 02783649241273668, 2023.

[4] Shizhe Chen, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Sugar: Pre-training 3d visual representations for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18049–18060, 2024.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[6] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[7] Michael Hagenow, Dimosthenis Kontogiorgos, Yanwei Wang, and Julie Shah. Versatile demonstration interface: Toward more flexible robot demonstration collection. *arXiv preprint arXiv:2410.19141*, 2024.

[8] Homanga Bharadhwaj, Abhinav Gupta, Vikash Kumar, and Shubham Tulsiani. Towards generalizable zero-shot manipulation via translating human interaction plans. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6904–6911. IEEE, 2024.

[9] Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. *arXiv preprint arXiv:2501.06994*, 2025.

[10] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024.

[11] Haoyi Zhu, Yating Wang, Di Huang, Weicai Ye, Wanli Ouyang, and Tong He. Point cloud matters: Rethinking the impact of different observation spaces on robot learning. *Advances in Neural Information Processing Systems*, 37:77799–77830, 2024.

[12] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.

[13] Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. *arXiv preprint arXiv:2407.15208*, 2024.

[14] Shengjie Wang, Jiacheng You, Yihang Hu, Jiongye Li, and Yang Gao. Skil: Semantic keypoint imitation learning for generalizable data-efficient manipulation. *arXiv preprint arXiv:2501.14400*, 2025.

[15] Hongyan Zhi, Peihao Chen, Siyuan Zhou, Yubo Dong, Quanxi Wu, Lei Han, and Mingkui Tan. 3dflowaction: Learning cross-embodiment manipulation from 3d flow world model. *arXiv preprint arXiv:2506.06199*, 2025.

[16] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

[17] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

[18] Chengyang He, Xu Liu, Gadiel Sznaier Camps, Guillaume Sartoretti, and Mac Schwager. Demystifying diffusion policies: Action memorization and simple lookup table alternatives. *arXiv preprint arXiv:2505.05787*, 2025.

[19] Shijie Wu, Yihang Zhu, Yunao Huang, Kaizhen Zhu, Jiayuan Gu, Jingyi Yu, Ye Shi, and Jingya Wang. Afforddp: Generalizable diffusion policy with transferable affordance. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6971–6980, 2025.

[20] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine intelligence 15*, pages 103–129, 1995.

[21] Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In *ICML*, volume 97, pages 12–20, 1997.

[22] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[23] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023.

[24] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023.

[25] Siyuan Huang, Haonan Chang, Yuhan Liu, Yimeng Zhu, Hao Dong, Peng Gao, Abdeslam Boularias, and Hongsheng Li. A3vlm: Actionable articulation-aware vision language model. *arXiv preprint arXiv:2406.07549*, 2024.

[26] Shizhe Chen, Ricardo Garcia, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. *arXiv preprint arXiv:2309.15596*, 2023.

[27] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[28] Adam Rashid, Satvik Sharma, Chung Min Kim, Justin Kerr, Lawrence Yunliang Chen, Angjoo Kanazawa, and Ken Goldberg. Language embedded radiance fields for zero-shot task-oriented grasping. In *7th Annual Conference on Robot Learning*, 2023.

[29] Yixuan Wang, Guang Yin, Binghao Huang, Tarik Kelestemur, Jiuguang Wang, and Yunzhu Li. Gendp: 3d semantic fields for category-level generalizable diffusion policy. In *8th Annual Conference on Robot Learning*, volume 2, 2024.

[30] Qianxu Wang, Haotong Zhang, Congyue Deng, Yang You, Hao Dong, Yixin Zhu, and Leonidas Guibas. Sparsedff: Sparse-view feature distillation for one-shot dexterous manipulation. *arXiv preprint arXiv:2310.16838*, 2023.

[31] Tong Zhang, Yingdong Hu, Hanchen Cui, Hang Zhao, and Yang Gao. A universal semantic-geometric representation for robotic manipulation. *arXiv preprint arXiv:2306.10474*, 2023.

[32] Tong Zhang, Yingdong Hu, Jiacheng You, and Yang Gao. Leveraging locality to boost sample efficiency in robotic

manipulation. *arXiv preprint arXiv:2406.10615*, 2024.

[33] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36:655–677, 2023.

[34] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.

[35] Ya Jing, Xuelin Zhu, Xingbin Liu, Qie Sima, Taozheng Yang, Yunhai Feng, and Tao Kong. Exploring visual pre-training for robot manipulation: Datasets, models and methods. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11390–11395. IEEE, 2023.

[36] Shengyi Qian, Kaichun Mo, Valts Blukis, David F Fouhey, Dieter Fox, and Ankit Goyal. 3d-mvp: 3d multiview pretraining for robotic manipulation. *arXiv preprint arXiv:2406.18158*, 2024.

[37] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[38] Nur Muhammad Mahi Shafiullah, Chris Paxton, Lerrel Pinto, Soumith Chintala, and Arthur Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022.

[39] Xinyu Zhang, Yuhan Liu, Haonan Chang, and Abdeslam Boularias. Scaling manipulation learning with visual kinematic chain prediction. *arXiv preprint arXiv:2406.07837*, 2024.

[40] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

[41] Guangqi Jiang, Yifei Sun, Tao Huang, Huanyu Li, Yongyuan Liang, and Huazhe Xu. Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets. *arXiv preprint arXiv:2410.22325*, 2024.

[42] Robert McCarthy, Daniel CH Tan, Dominik Schmidt, Fernando Acero, Nathan Herr, Yilun Du, Thomas G Thuruthel, and Zhibin Li. Towards generalist robot learning from internet video: A survey. *Journal of Artificial Intelligence Research*, 83, 2025.

[43] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[44] Jiange Yang, Bei Liu, Jianlong Fu, Bocheng Pan, Gangshan Wu, and Limin Wang. Spatiotemporal predictive pre-training for robotic motor control. *arXiv preprint arXiv:2403.05304*, 2024.

[45] Ruihai Wu, Yan Zhao, Kaichun Mo, Zizheng Guo, Yian Wang, Tianhao Wu, Qingnan Fan, Xuelin Chen, Leonidas Guibas, and Hao Dong. Vat-mart: Learning visual action trajectory proposals for manipulating 3d articulated objects. *arXiv preprint arXiv:2106.14440*, 2021.

[46] Homanga Bharadhwaj, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Zero-shot robot manipulation from passive human videos. *arXiv preprint arXiv:2302.02011*, 2023.

[47] Kenneth Shaw, Shikhar Bahl, Aravind Sivakumar, Aditya Kannan, and Deepak Pathak. Learning dexterity from human hand motion in internet videos. *The International Journal of Robotics Research*, 43(4):513–532, 2024.

[48] Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J Yoon, Ryan Hoque, Lars Paulsen, et al. Humanoid policy˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[49] Ben Eisner, Harry Zhang, and David Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. *arXiv preprint arXiv:2205.04382*, 2022.

[50] Daniel Seita, Yufei Wang, Sarthak J Shetty, Edward Yao Li, Zackory Erickson, and David Held. Toolflownet: Robotic manipulation with tools via predicting tool flow from point clouds. In *Conference on Robot Learning*, pages 1038–1049. PMLR, 2023.

[51] Yixiang Chen, Peiyan Li, Yan Huang, Jiabing Yang, Kehan Chen, and Liang Wang. Ec-flow: Enabling versatile robotic manipulation from action-unlabeled videos via embodiment-centric flow. *arXiv preprint arXiv:2507.06224*, 2025.

[52] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

[53] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation. *CoRR*, 2024.

[54] Chongkai Gao, Haozhuo Zhang, Zhixuan Xu, Zhehao Cai, and Lin Shao. Flip: Flow-centric generative planning as general-purpose manipulation world model. *arXiv preprint arXiv:2412.08261*, 2024.

[55] 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.

[56] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *Proc. ECCV*, 2024.

[57] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[59] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[60] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[61] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[62] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in neural information processing systems*, 35:23192–23204, 2022.

[63] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

[64] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[65] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019.