POLITECNICO DI TORINO
Repository ISTITUZIONALE

GAP: Geometric Anchor Pre-training for Data-Efficient Visuomotor Learning of Manipulation Tasks

Original
GAP: Geometric Anchor Pre-training for Data-Efficient Visuomotor Learning of Manipulation Tasks / Buoso, D., Protopapa, A., Di Carlo, S., Pistilli, F., Averta, G.. - (In corso di stampa). (IEEE/RSJ International Conference on Intelligent Robots and Systems Pittsburgh (USA)) [10.48550/ARXIV.2605.15836].

Availability:
This version is available at: 11583/3016994 since: 2026-10-26T09:39:21Z

Publisher:
Institute of Electrical and Electronics Engineers (IEEE)

Published
DOI:10.48550/ARXIV.2605.15836

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©9999 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

19 November 2026

arXiv:2605.15836v1 [cs.RO] 15 May 2026

GAP: Geometric Anchor Pre-training for Data-Efficient Visuomotor Learning of Manipulation Tasks

Davide Buoso, Andrea Protopapa, Stefano Di Carlo, Francesca Pistilli, and Giuseppe Averta
Department of Control and Computer Engineering, Polytechnic University of Turin, Italy

***Abstract*— Learning visuomotor policies from scarce expert demonstrations remains a core challenge in robotic manipulation. A primary hurdle lies in distilling high-dimensional RGB representations into control-relevant geometry without overfitting. While using frozen pretrained Vision Foundation Models (VFMs) improves data efficiency, it also shifts most task adaptation onto a small spatial pooling module, which can latch onto task-irrelevant shortcuts and lose geometric grounding when finetuned with few data samples. More broadly, pretrained visual representations used for policy learning have been observed to struggle under even minor scene perturbations, highlighting the need for robustness-oriented inductive biases. We propose Geometric Anchor Pre-training (GAP), a simple, action-free warm-up stage that regularizes the spatial adapter *before* downstream imitation learning. GAP pre-trains the pooling layer on a lightweight simulated proxy task where object masks are available at no cost, encouraging the adapter to produce keypoints that lie on the object, cover its spatial extent (instead of collapsing), and remain sharp and repeatable over time. This yields stable *geometric anchors* that provide a reliable coordinate interface for few-shot policy learning, while keeping the VFM frozen. We evaluate GAP on RoboMimic and ManiSkill under severe data scarcity (15–50 demonstrations) and domain shift. A simple adapter regularized with GAP consistently outperforms stronger attention-based poolers and end-to-end fine-tuning, achieving 62% success on RoboMimic `Can` with 15 demonstrations (+16% over AFA), 63% on the long-horizon high-precision `Tool Hang` task with 50 demonstrations (+13% over the best competitor based on R3M with Spatial Softmax) , and 61% on ManiSkill `StackCube` with 30 demonstrations (+11% over full fine-tuning). The proxy stage is lightweight (about 40 minutes on a single consumer GPU) and fully decoupled from downstream tasks, making it practical to reuse across environments and manipulation skills. Project page: https://lambdavi.github.io/gap/**

I. Introduction

Imitation Learning (IL) has become a strong paradigm for robotic manipulation, enabling robots to acquire complex, contact-rich behaviors directly from expert demonstrations. Recent diffusion-based policies further improved performance by modeling multimodal action distributions and have become a common solution for visuomotor IL [1]. However, in many practical settings, demonstrations are scarce, and learning reliable visuomotor policies remains challenging because the agent must extract task-relevant geometry from a few high-dimensional RGB observations without overfitting.

Corresponding author: Davide Buoso (e-mail: davide.buoso@polito.it).

This work was carried out within the Future Artificial Intelligence Research (FAIR) and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support.

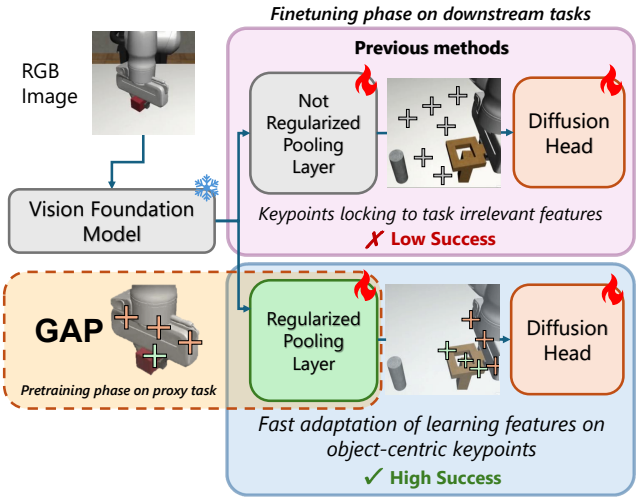


Fig. 1: **Geometric Anchor Pretraining.** We introduce GAP, a pretraining strategy applied to the spatial pooling layer on a cheap proxy task. When only a few demonstrations are available for the target task ($N \leq 50$), regularizing the spatial bottleneck with GAP consistently and largely outperforms other pooling techniques and end-to-end fine-tuning.

A common recipe to improve data efficiency is to (i) freeze a pretrained Vision Foundation Model (VFM) as the visual backbone (e.g., VC-1 [2] or DINOv2 [3]) and (ii) train a lightweight spatial bottleneck that compresses dense feature maps into a compact representation for control. Methods using Diffusion Policies often instantiate this bottleneck with Spatial Softmax, yielding a set of 2D keypoints [1], [4]. More semantic alternatives aim to learn which visual elements to retain: TokenLearner adaptively selects a small set of informative tokens [5], while Attentive Feature Aggregation (AFA) is a lightweight trainable pooling mechanism designed to attend to task-relevant cues and suppress distractors without fine-tuning the frozen backbone [6]. These modular setups are appealing because they preserve the semantic knowledge of the pretrained VFM while keeping the task-specific learnable component small.

Yet, in the low-data regime, the spatial bottleneck is also the main point of failure. With only a handful of demonstrations, the pooling module can lock onto easy-to-fit visual shortcuts (often in the background) instead of learning stable,

object-centric geometry, producing keypoints that are poorly localized and brittle to minor test-time changes [7]. We refer to this failure mode as *bottleneck collapse*: the adapter loses geometric grounding, and the downstream policy becomes unreliable under even small distribution shifts.

To address this, we propose **Geometric Anchor Pre-training (GAP)**, a simple yet particularly effective strategy that prevents bottleneck collapse by regularizing the spatial adapter *before* downstream policy (see Figure 1). The key intuition behind GAP is that many geometric cues needed for contact-rich manipulation—object extent, salient extremities, and stable spatial support—are transferable and can be learned without access to downstream actions.

GAP adds a short, action-free warm-up stage on a cheap simulated proxy task, where object masks are available at no cost. During this warm-up, the adapter is encouraged to produce keypoints that lie on the object, cover it rather than collapsing to a single location, and remain sharp and repeatable across time. This produces stable *geometric anchors* that serve as a reliable coordinate interface when learning the downstream policy from a few demonstrations.

After this warm-up, we train the downstream policy in a few-shot setting (15–50 demonstrations), using the GAP-initialized adapter as the visual bottleneck. We validate GAP on RoboMimic [8] and ManiSkill [9] under severe data scarcity and domain shift. Our results show satisfactory performance, with GAP achieving 62% success on `Can` (RoboMimic) with only 15 demonstrations ($+16\%$ over AFA, $+7\%$ over full finetuning), 63% on the high-precision long-horizon `Tool Hang` (RoboMimic) with 50 demonstrations ($+13\%$ over the best competitor R3M [10] + Spatial Softmax [4]), and scoring 61% on `StackCube` (Maniskill) with 30 demonstrations ($+11\%$ over full finetuning). Notably, the proxy stage is lightweight (approximately 40 minutes on a single consumer GPU) and is fully decoupled from the downstream task, making it practical to reuse across tasks and environments.

To summarize, this paper contributes the following:

- We introduce **GAP**, an action-free pretraining strategy for spatial pooling layers that injects geometric priors via a cheap mask-supervised proxy task, preventing bottleneck collapse in few-shot visuomotor IL.
- We demonstrate strong empirical gains in two different benchmarks (RoboMimic and ManiSkill) in extremely low-data regimes (15–50 demonstrations) under domain and tasks shift.
- We provide experimental proofs that explicit geometric regularization is necessary, and that simply exposing the adapter to additional proxy data is insufficient and can cause negative transfer.
- We demonstrate that GAP learns transferable geometric anchors that improve robustness and data efficiency across tasks, visual domain shifts, and across simulation environments (pretrain on RoboMimic, transfer to ManiSkill).

II. Related Work

Visual Foundation Models for Robotics. The adoption of large-scale pre-trained visual backbones has heavily driven recent advances in robot learning. Models such as MVP [11], VC-1 [2], and R3M [10] have demonstrated that representations learned from internet-scale video datasets (e.g., Ego4D [12]) or image-text pairs [13] can provide rich semantic features for downstream manipulation tasks. This paradigm is dominant because internet-scale pre-training provides robust generalization across novel object categories and diverse semantic environments.

However, this semantic focus introduces a fundamental trade-off: pre-training objectives that maximize semantic invariance (e.g., contrastive learning on image classification) inherently suppress high-frequency spatial and geometric details. In short, VFMs learn to recognize *what* an object is, often at the expense of precisely locating *where* its extremities are—a critical requirement for contact-rich manipulation. This comes with two major limitations. First, fine-grained tasks (e.g., `Tool Hang`, `Square Nut Assembly` [8]) require sub-centimeter geometric precision that global semantic embeddings simply cannot resolve. Second, as demonstrated in our experiments, VFM-based policies frequently suffer visual distribution shifts [14] because they tend to rely on spurious correlations (e.g., background textures, lighting) rather than invariant geometric structures.

Keypoint-Based and Structured Representations. Condensing high-dimensional visual states into sparse coordinate representations has a long history in robotics. Foundational architectures like Transporter Nets [15] and KeypointNet [16] extract low-dimensional structures via template matching or heuristic geometric reasoning. While sample-efficient, these early approaches often rely on reconstruction objectives that do not guarantee control-relevant disentanglement.

More recently, two divergent strategies have emerged. The first leverages large generative models—such as extracting features from diffusion U-Nets [17] or utilizing VFMs for semantic correspondence [18]. While semantically rich, these methods introduce large computational overhead during inference or introduce dependencies on external reference images. The second strategy relies on structured geometric pipelines, pairing semantic keypoints with shape completion for category-level planning [19], [20], or projecting depth into structured point maps [21].

These diverse approaches confirm a broader consensus: visuomotor control fundamentally benefits from explicit spatial structure. However, existing methods force a compromise, requiring either heavy inference-time computation, external model dependencies, or dense manual supervision. GAP avoids this compromise. By enforcing geometric consistency through a synthetic, mask-supervised proxy task, GAP makes spatial anchor extraction lightweight, reference-free, and seamlessly scalable to standard end-to-end Diffusion Policies.

Pooling strategies for Vision Foundation Models As Vision Foundation Models (VFMs) output dense, high-dimensional feature maps, compressing this spatial informa-

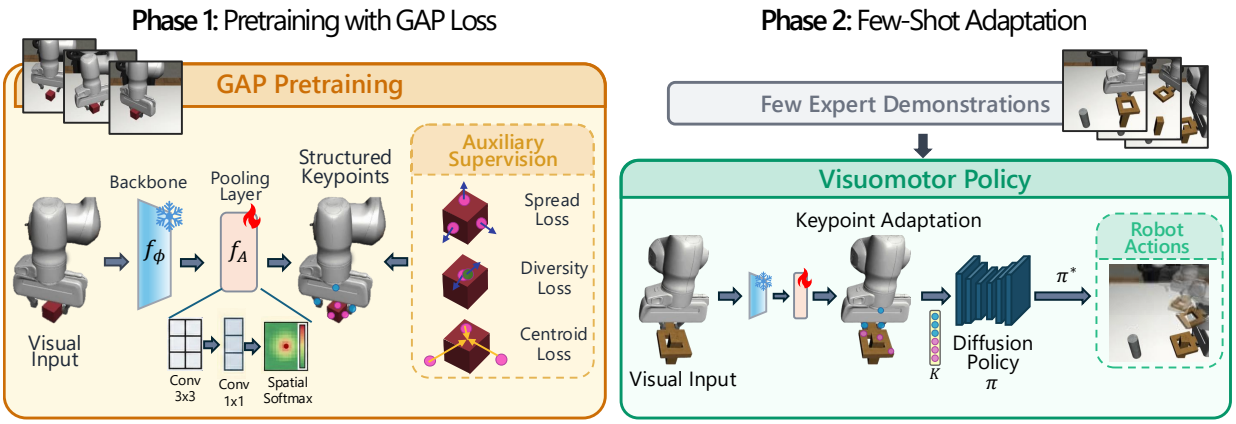


Fig. 2: **Method Overview.** 1. The spatial pooling layer extracts keypoints from the semantic pretrained backbone (frozen). GAP supervises this layer with the proposed loss, providing geometric grounding for policy learning. 2. Backbone and warmed-up pooling layer are then used to generate the input for the Diffusion Policy. During downstream training, the pooling layer is fine-tuned per task, to adapt object keypoints placement to the objects present in novel scenes.

tion for downstream policy learning has become a critical architectural focus. Recent approaches leverage attention mechanisms to dynamically aggregate visual information into compact token representations. Methods such as TokenLearner [5] and Perceiver IO [22] reduce spatial dimensionality by computing attention between learned latent queries and the input feature map, theoretically allowing the network to focus only on task-relevant semantic regions. Building on these, in the context of robotic manipulation, frameworks employing Attention Feature Aggregation (AFA) [6] and similar transformer-based bottlenecks have been proposed to filter visual distractors and improve robustness by explicitly learning only semantic features that are important for the manipulation tasks. However, these semantic attention mechanisms rely heavily on larger demonstration corpora to learn robust query-key mappings. As we demonstrate empirically, this high capacity becomes a critical liability in severe low-data regimes (e.g., 15–50 demonstrations), leading to severe representation collapse. Rather than isolating precise object coordinates, highly parameterized attention poolers frequently overfit to spurious transient features, such as background textures or specific lighting conditions. GAP explicitly counters this by trading semantic flexibility for strict geometric equivariance. By forcing the VFM features through a low-parameter, rigidly regularized spatial adapter, GAP extracts stable (x, y) keypoints instead of diffuse attention maps. This strict structural constraint prevents the texture-hijacking typical of high-capacity poolers, providing robust and noise-free spatial priors that largely accelerate convergence in data-starved environments while providing a simpler representation to the policy .

III. Methodology

Geometric Anchor Pretraining (GAP) is a pretraining strategy designed to extract object-centric geometric priors for visuomotor imitation learning directly from dense VFM's embeddings. GAP addresses the spatial overfitting commonly observed in data-scarce regimes ($N \leq 50$) by explicitly supervising a coordinate-based adapter via a masked proxy task. We begin by formalizing the imitation learning setting and defining the spatial adapter architecture. We then introduce our method (Figure 2), focusing on the proxy task and losses, followed by a description of the downstream policy adaptation procedure.

A. Imitation Learning with Diffusion Policies

We focus on visuomotor policy learning in the Imitation Learning setting, where an agent learns control behaviors from expert demonstrations. The agent is provided with a dataset $\mathcal{D} = \{\tau_i\}_{i=1}^{N}$, where each trajectory $\tau_i = \{(o_t, a_t)\}_{t=0}^{T}$ consists of visual observations o_t and corresponding actions a_t. The objective is to infer a policy $\pi_\theta(a_t|o_t)$ that reproduces the expert behavior.

In this work, we adopt **Diffusion Policies** [1], which parameterize the action distribution $\pi_\theta(a_t|o_t)$ as a conditional denoising diffusion probabilistic model. The training objective minimizes the noise prediction error:

$$\mathcal{L}_{diff} = \mathbb{E}_{\epsilon\sim\mathcal{N},t\sim\mathcal{U},\tau\sim\mathcal{D}}\left[\|\epsilon - \epsilon_\theta(a_t^{(k)}, k, E(o_t))\|_2^2\right] \quad (1)$$

where $E(o_t)$ is the visual embedding used to condition the denoising network ϵ_θ at diffusion step k. While a wide range of Imitation Learning algorithms have been introduced, with no loss of generality, in this work, we experiment with [1] as the policy learning method for all the experiments to isolate the impact of the conditioning representation. GAP focuses entirely on the pre-training of the adapter between the frozen vision encoder and the policy π_θ to provide a robust, geometry-aware conditioning signal.

B. The Spatial Adapter

To extract robust semantic features, we use a frozen pretrained backbone f_ϕ (e.g., ResNet-50, ViT-S, or ViT-B). To map visual features into precise spatial embeddings, f_ϕ is followed by a lightweight adapter module, which we denote as f_A (see Figure 2). Specifically, f_A first applies a shallow convolutional network (a 3×3 followed by a 1×1 convolution) to project the high-dimensional backbone features into K spatial activation maps, denoted as $\Phi_t \in$

$\mathbb{R}^{K\times h\times w}$. Finally, f_A applies a Spatial Softmax (SS) [4] module to convert these maps into K 2D spatial coordinates, which we define as our candidate *keypoints* $P_t = \{p_{k,t}\}_{k=1}^{K}$:

$$p_{k,t} = \sum_{x=1}^{w}\sum_{y=1}^{h}\begin{bmatrix}x\\y\end{bmatrix}\frac{\exp(\Phi_{t,k,x,y})}{\sum_{x',y'}\exp(\Phi_{t,k,x',y'})} \tag{2}$$

where $\Phi_{t,k,x,y}$ is the activation of the k-th feature channel at spatial location (x, y).

This mapping converts the visual learning paradigm into a state-based one, compressing dense feature maps into K sparse 2D coordinates. When training policies for 200+ demonstrations, we observe that the learned keypoints naturally stick to specific objects and become reliable "semantic trackers". However, under extremely low-data regime, without explicit supervision, these keypoints tend to latch onto spurious visual cues rather than geometrically meaningful locations. This occurs because the downstream action regression loss ($\mathcal{L}_{diff}$) provides only weak, indirect spatial supervision. Forced to minimize training error with minimal data, the network takes a shortcut, anchoring to high-contrast static distractors (e.g., table textures) instead of complex object geometry.

C. Geometric Anchor Pretraining (GAP)

To address this, GAP pretrains the spatial bottleneck f_A on a single, cheap simulated proxy task. This aims to decouple geometric feature learning from action-mapping.

Proxy Task. The objective of our training is to align geometric keypoints with task-relevant objects in the scene, without any task-specific knowledge. Therefore, in principle we can use for pretraining any simple manipulation task or contact-rich motor babbling. In this paper, with no loss of generality, we experiment with the `LiftCube` task from Robomimic [8]— the simplest task available in the benchmark—in which a Franka Emika Panda robot is tasked to reach, grasp, and lift a randomly positioned cube on a plain *white-background* table (see Figure 3-a). To evaluate proxy task invariance, we also experiment with a different pretraining task (PlaceSphere from ManiSkill). Trajectories are generated automatically via a scripted controller, requiring no human teleoperation. We use 100 demonstrations of this proxy task, leveraging ground-truth object segmentation masks $\mathcal{M}_t$ provided by the simulator to supervise our loss. Crucially, no expert action labels a_t are needed at any point during this phase.

GAP Spatial Objectives. We supervise the adapter f_A using a multi-objective spatial loss that enforces object-centric, spatially distributed, and non-redundant keypoints over the robot gripper and interacting objects. This is achieved through three components, depicted in Figure 2 and detailed below.

1) Centroid Alignment ($\mathcal{L}_{center}$): To ensure keypoints ground themselves on the target object rather than background distractors, we minimize the distance between the predicted keypoint centroid $\bar{p}_t$ and the ground-truth mask centroid c_t:

$$\mathcal{L}_{center} = \|\bar{p}_t - c_t\|_2^2 \quad \text{where} \quad \bar{p}_t = \frac{1}{K}\sum_{k=1}^{K} p_{k,t} \tag{3}$$

and c_t is computed via the spatial moments of the binary mask $\mathcal{M}_t$.

2) Geometric Spread ($\mathcal{L}_{spread}$): To prevent the degenerate solution where all keypoints collapse precisely in the centroid (which drops orientation information), we enforce the spatial variance of the keypoints σ_p to match the normalized object scale σ_{target}:

$$\mathcal{L}_{spread} = \|\sigma_p - \sigma_{target}\|_2^2 \quad \text{where} \quad \sigma_p = \frac{1}{K}\sum_{k=1}^{K}\|p_{k,t} - \bar{p}_t\|_2 \tag{4}$$

The target scale is derived from the mask area $A_t = \sum \mathcal{M}_t$, approximated as $\sigma_{target} = 0.8 \times \sqrt{A_t/\pi}$, representing a proportional bounding radius.

3) Keypoint Diversity ($\mathcal{L}_{div}$): Lastly, to maximize the structural information captured by the bottleneck, we penalize redundancy by enforcing a minimum separation margin

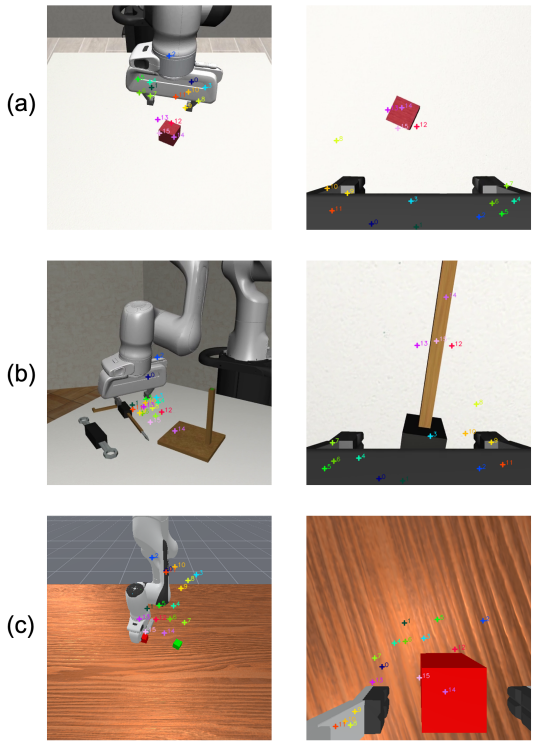


Fig. 3: **Qualitative Keypoint Transfer.** When pre-trained on a proxy task and then transferred to a new task and simulator, GAP allows for keeping a favorable geometric grounding even in zero-shot. **(a)** shows the keypoints placement on the Robomimic `LiftCube` task after pretraining. **(b)** and **(c)** show keypoints positioning when using the pre-trained visual encoder in zero-shot on a different task (of the same simulator) and on a different task and simulator (Maniskill), respectively. We show a third-person view on the left and a first-person view on the right.

δ_{min} between any pair of keypoints:

$$\mathcal{L}_{div} = \frac{1}{K}\sum_{k=1}^{K}\left[\max\left(0, \delta_{min} - \min_{j\neq k}\|p_{k,t} - p_{j,t}\|_2\right)\right]^2 \tag{5}$$

This term encourages the network to discover the object's distinct geometric extremities, thereby generating highly informative *Geometric Anchors*.

The final pre-training objective combines these terms to create a "push-pull" dynamic: keypoints are pulled onto the object ($\mathcal{L}_{center}$), but pushed outward to span its geometry ($\mathcal{L}_{spread}$) and away from one another ($\mathcal{L}_{div}$).

$$\mathcal{L}_{GAP} = \lambda_c\mathcal{L}_{center} + \lambda_s\mathcal{L}_{spread} + \lambda_d\mathcal{L}_{div} \tag{6}$$

In our implementation, we prioritize spatial coverage over strict centering by setting $\lambda_c = 0.3$, $\lambda_s = 0.5$, and $\lambda_d = 2.0$, with a diversity margin $\delta_{min} = 0.15$ (normalized image coordinates). We report an extensive ablation on the different objective components in Sec. IV-D.

Object-Centric Keypoint Allocation. While end-to-end policies require massive datasets to naturally develop entity-centric keypoints, GAP explicitly enforces this optimal behavior in low-data regimes. Given a pre-training scene with M semantic entities, we partition the K available keypoints into M disjoint subsets: $P_t = \bigcup_{m=1}^{M} P_{t,m}$. The spatial regularization objectives ($\mathcal{L}_{GAP}$) are then applied independently to each subset using its corresponding mask. This M-way partition bridges the abstract semantic expressivity of Vision Foundation Models (VFMs) with the strict, object-centric geometric priors required for physical manipulation. Consequently, downstream fine-tuning does not need to learn object separation from scratch. When transitioning from a proxy task to novel, multi-object environments (e.g., `StackCube`, `SquareNut`), these pre-trained subsets deploy as independent semantic trackers. Having already internalized priors of centroid alignment, spatial spread, and extremity-seeking diversity, they rapidly re-anchor to novel geometries with minimal adaptation. Figure 3 qualitatively demonstrates this robust, entity-separated initialization prior to any policy training.

D. Downstream Policy Adaptation

Following GAP, the regularized adapter and frozen encoder E are transferred to the downstream tasks. The M pre-trained keypoint subsets drastically reduce the policy's learning burden: subsets tracking persistent elements (e.g., the manipulator) require light adaptation, allowing the few-shot demonstrations to focus entirely on grounding the remaining keypoints to novel target objects. This preserves learned spatial priors and enables highly sample-efficient convergence. For fair comparison, **all evaluated baselines**—including end-to-end models and VFMs (R3M, DINOv2, VC-1)—employ an identical architecture. Models denoted with "SS" utilize our convolutional pooler and Spatial Softmax. During downstream fine-tuning, privileged segmentation masks $\mathcal{M}_t$ are strictly discarded. The vision backbone remains frozen, and we fine-tune only the lightweight adapter f_A alongside the diffusion head via the action-prediction objective $\mathcal{L}_{diff}$.

IV. Experiments

We evaluate GAP on two simulation benchmarks: Robomimic [8] and ManiSkill3 [9]. For all tasks, our observations consist of two camera views (i.e., an agent-centric and a wrist-mounted camera). Each camera stream is processed independently through its own instantiated visual backbone and GAP-regularized adapter f_A. The resulting keypoints are then concatenated before being passed to the diffusion policy. Noteworthy, we measure the number of trainable parameters of the methods: End-to-end training of the Resnet50 requires updating about 56M of parameters, while AFA and GAP require about 3M and 2M, respectively. Our experimental evaluation is designed to answer three research questions: 1) Does explicit geometric pre-training of the spatial bottleneck (GAP) improve data efficiency in the $N \leq 50$ regime, and does the learned prior transfer across simulator and tasks? 2) Is GAP appropriate to mitigate the "bottleneck" problem across backbones architecture and VFMs? 3) What drives GAP's performance gain: the proxy data, the geometric loss, or the disentangled keypoint structure?

A. Experimental Setup

We evaluate on four tasks of increasing difficulty: `PickAndPlace Can`, `Square Nut Assembly`, and `Tool Hang` from Robomimic [8], and `StackCube` from ManiSkill3 [9]. `PickAndPlace Can` is a relatively simple pick-and-place task; `Square Nut Assembly` requires precise peg insertion; `Tool Hang` is a long-horizon, multi-step assembly task; `StackCube` requires stacking one cube precisely above another. The difficulty of the task comes from the heavy randomization of object positioning over the whole table, which represents a non-trivial problem in a data-scarce regime. All tasks are evaluated over different settings, using 15, 20, 30, and 50 expert demonstrations available for policy training. Results are averaged over three seeds.

Baselines. For each task and setting we compare: a Resnet50 End-to-End full fine-tuned (E-E) replicating a Diffusion Policy setup [1], three frozen VFM backbones paired with a standard Spatial Softmax-based adapter (R3M+SS, DINOv2+SS, VC-1+SS), and the best performing backbone paired with Attention Feature Aggregation (AFA) [6] as SOTA attention pooler. DinoV2 is used with their ViT-S backbone, and VC-1 is used with their ViT-B to study different backbone sizes/pretraining impact on downstream policy learning. All baselines share the same downstream architecture (U-Net trained with diffusion objective); leaving as only difference between them the design of the pooling layer and how it is trained. We train all models for 1,000 epochs with 512 as batch size, and report the results with their best performing learning rates (between 1e-3 and 1e-5) and their best configurations (backbone) in the case of AFA and GAP. For all experiments we fix the number of keypoints to 16 per camera. For all tasks and settings, we use the same pre-training proxy task, i.e., `LiftCube` from Robomimic, scripted automatically on a white-background table, requiring no expert action labels (Figure 3). Pre-training runs for 10,000 steps (≈40 min on one A40 GPU) and relies solely

Method	Robomimic: Can					Robomimic: Square				
	15	20	30	50	Avg	15	20	30	50	Avg
E-E (Full FT)	0.55 (0.04)	0.76 (0.02)	0.88 (0.03)	0.95 (0.01)	0.785	0.15 (0.03)	0.19 (0.02)	0.29 (0.03)	0.38 (0.02)	0.253
R3M + SS	0.50 (0.02)	0.75 (0.05)	0.78 (0.04)	0.86 (0.02)	0.723	0.12 (0.02)	0.13 (0.04)	0.17 (0.04)	0.22 (0.02)	0.158
DINOv2 + SS	0.51 (0.08)	0.68 (0.03)	0.73 (0.03)	0.86 (0.00)	0.695	0.10 (0.00)	0.22 (0.02)	0.23 (0.04)	0.26 (0.03)	0.203
VC-1 + SS	0.49 (0.06)	0.64 (0.10)	0.82 (0.05)	0.89 (0.02)	0.710	0.07 (0.06)	0.24 (0.06)	0.23 (0.03)	0.32 (0.05)	0.215
AFA	0.46 (0.01)	0.74 (0.02)	0.78 (0.02)	0.93 (0.05)	0.728	0.15 (0.02)	0.19 (0.03)	0.32 (0.04)	0.43 (0.04)	0.273
GAP (Ours)	**0.62** (0.06)	**0.80** (0.02)	**0.94** (0.04)	**0.96** (0.02)	**0.830**	**0.20** (0.03)	**0.33** (0.03)	**0.37** (0.01)	**0.53** (0.03)	**0.358**
	Robomimic: Tool Hang					**ManiSkill: StackCube**				
Method	**15**	**20**	**30**	**50**	**Avg**	**15**	**20**	**30**	**50**	**Avg**
E-E (Full FT)	0.06 (0.05)	0.2 (0.1)	0.13 (0.06)	0.33 (0.06)	0.18	**0.22** (0.10)	0.23 (0.05)	0.50 (0.08)	0.66 (0.08)	0.403
R3M + SS	0.16 (0.06)	0.17 (0.06)	0.27 (0.06)	0.50 (0.10)	0.275	0.06 (0.01)	0.09 (0.02)	0.15 (0.05)	0.38 (0.09)	0.171
DINOv2 + SS	0.23 (0.06)	0.13 (0.06)	0.20 (0.17)	0.23 (0.23)	0.198	0.07 (0.01)	0.10 (0.03)	0.15 (0.00)	0.60 (0.05)	0.230
VC-1 + SS	0.20 (0.05)	0.17 (0.11)	0.20 (0.10)	0.43 (0.05)	0.250	0.04 (0.01)	0.08 (0.05)	0.28 (0.02)	0.63 (0.03)	0.258
AFA	0.20 (0.10)	0.23 (0.10)	0.30 (0.06)	0.45 (0.10)	0.295	0.09 (0.02)	0.14 (0.03)	0.25 (0.02)	0.44 (0.08)	0.230
GAP (Ours)	**0.27** (0.06)	**0.33** (0.06)	**0.37** (0.06)	**0.63** (0.05)	**0.400**	0.20 (0.03)	**0.24** (0.04)	**0.61** (0.10)	**0.80** (0.02)	**0.463**

TABLE I: **Multi-Task Evaluation Results.** For all tasks, we pre-train on *LiftCube* from Robomimic. Results on the ManiSkill simulator environment are shaded in gray to denote the domain shift. GAP achieves state-of-the-art average performance. For GAP the best performing VFM is VC1 with ViT-B while for AFA is VC1 for *Can* and R3M for the other tasks.

Pre-training Strategy	15 Demos	20 Demos	30 Demos	50 Demos
Resnet50 E-E ([1])	0.22 (0.10)	0.23 (0.05)	0.50 (0.08)	0.66 (0.08)
VC-1 + SS (No Pre-training)	0.04 (0.01)	0.08 (0.05)	0.28 (0.02)	0.63 (0.03)
AFA	0.09 (0.02)	0.14 (0.03)	0.25 (0.02)	0.44 (0.08)
GAP (Same Table)	**0.36** (0.07)	**0.35** (0.06)	0.56 (0.10)	**0.80** (0.01)
GAP (Cross Table)	0.28 (0.15)	0.31 (0.06)	0.54 (0.10)	0.74 (0.02)
GAP (Cross-Sim)	0.19 (0.06)	0.24 (0.04)	**0.61** (0.1)	**0.80** (0.02)

TABLE II: **Proxy Task Ablation. (StackCube, Maniskill)** The performance of StackCube on Maniskill are analyzed under three pretrained conditions: Same Table (pretraining on *PlaceSphere* on Maniskill), Cross Table (pretraining on modified *PlaceSphere* with white table background on Maniskill), and Cross-Sim (pretraining on *LiftCube* on Robomimic). Performance remains consistent across all settings, demonstrating that GAP learns domain-agnostic geometric priors.

on simulator-provided segmentation masks, which are used only in this pre-training phase.

B. Impact of Pretraining on Downstream Task Learning

Table I reports success rates across all four tasks and settings. We note that semantic attention (AFA) [6] overfits heavily at low demo counts: e.g. on StackCube using 15 demos, AFA achieves only 0.09 while VC-1+SS reaches 0.04 and GAP reaches $\mathbf{0.20}$. With 30 demos, AFA (0.25) is actually outperformed by a simpler VC-1+SS (0.28), confirming that larger pooling capacity may be harmful when data is scarce; GAP performs two times better achieving $\mathbf{0.61}$. We also observe that unregularized SS yields inconsistent results, performing reasonably well on simple tasks but failing to anchor to object geometry on harder ones. In addition, it is worth noting that E-E fine-tuning underperforms on `Tool Hang` and lags behind GAP on every task, despite updating more parameters. GAP consistently achieves the highest success rate across all tasks and settings. On `Can`, GAP reaches 0.62 with only 15 demos versus AFA's 0.46 and E-E's 0.55, and reaches 0.96 at 50 demos. On the challenging `Square` task, GAP scores 0.53 with 50 demos versus 0.43 for AFA and 0.38 for E-E. On `Tool Hang` E-E fails in learning the task with 15 demos, while GAP achieves a reasonable 0.27, and 0.63 using 50 demos, consistently outperforming all baselines. Finally, we evaluate the `Square` task using 100 demonstrations to observe if abundant expert data mitigates spatial bottleneck collapse. At this scale, VC-1 + SS, VC-1 + AFA, and our GAP framework achieve success rates of 0.64 ± 0.05, 0.65 ± 0.03, and 0.68 ± 0.02, respectively. As expected, the massive influx of expert trajectories allows the unregularized baselines to largely close the performance difference, confirming that GAP's geometric prior is most critical—and provides the highest relative gains—in data-scarce regimes, but still provides advantages in higher ones. A natural concern is that GAP's advantage may stem from extra data in the training domain rather than a genuinely general geometric prior. Table II reports an ablation study to verify it. We compare three pre-training conditions on StackCube: *Same Table* (PlaceSphere in ManiSkill as the proxy task, using the same wooden table as the downstream task), *Cross Table* (modified PlaceSphere with white table background on Maniskill as proxy tasks, visual shift), and *Cross-Sim* (*LiftCube* in Robomimic as proxy task, simulator shift). At 30 demos, the three conditions yield on average 0.56, 0.54, and 0.59, respectively—with no statistically significant difference between them. All three variations largely outperform the VC-1+SS baseline (0.28). This confirms that GAP learns general priors rather than simulator-specific textures or visual priors. We also note that the success of *Cross-Sim* also proves that GAP is task-agnostic to the choice of proxy task.

C. GAP impact on different backbones

To prove that bottleneck collapse is a problem for various backbones, Figure 4 evaluates all pooling strategies across R3M, frozen DINOv2, and frozen VC-1 on the Square task at 30 demos, requiring learning of precise actions from few demos. Additionally, we also present the results on StackCube (Maniskill). Similar trends can be observed for other tasks and are here omitted for the sake of space. The results demonstrate a flaw in semantic pooling: adding AFA to DINOv2 *degrades* performance w.r.t. standard SS (0.19 vs. 0.23), confirming that semantic mechanisms are

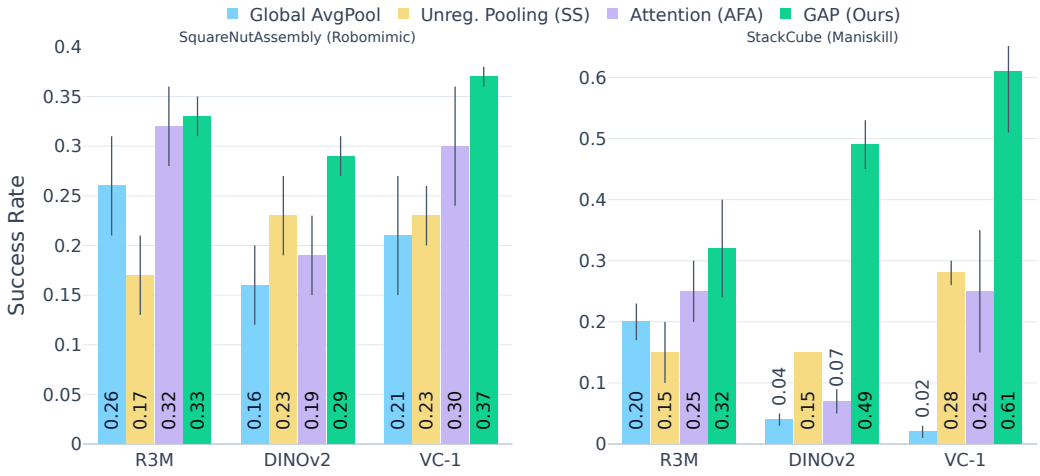


Fig. 4: **GAP impact on different backbones (Square, 30 Demos).** We evaluate various pretrained backbones (R3M [10], VC1 [2] and DinoV2 [3]) with different poolers: a Global Avg. Pooling (blue), unregularized geometric adapter with Spatial Softmax (yellow), AFA [6] (purple) and a GAP-pretrained spatial adapter (green). GAP consistently outperforms other methods by a large margin, demonstrating that all backbones benefit from our pretraining. Results averaged over three seeds.

not enough when data is scarce. GAP instead improves DINOv2, bringing it from 0.23 to **0.29**, and pushes VC-1 to the state-of-the-art **0.37** on this task—higher than any other method including E-E fine-tuning (0.29). GAP acts as a spatial regularizer, compatible with any backbone.

$\mathcal{L}_{cen}$	$\mathcal{L}_{div}$	$\mathcal{L}_{spr}$	**Success Rate**	Δ
✓	✓	✓	**0.37**	
✓	✓		0.24	↓**0.13**
✓		✓	0.32	↓**0.05**
	✓	✓	0.26	↓**0.11**
✓			0.30	↓**0.07**
		✓	0.26	↓**0.11**
	✓		0.22	↓**0.15**

TABLE III: **Loss Ablation (Square, 30 Demos).** Success rates when removing one or two loss components. Removing any individual term or combination degrades performance, proving that all the three terms synergistically contribute to the GAP objective.

Pretraining Objective	**Base: VC-1 + SS**	**Base: VC-1 + AFA**
Baseline (No Pretraining)	0.23 (0.03)	0.31 (0.05)
+ Policy Training	0.19 (0.04) ↓**0.04**	0.27 (0.05) ↓**0.04**
+ GAP (Ours)	**0.37** (0.06) ↑**0.14**	—

TABLE IV: **Pretraining Impact (Square, 30 Demos).** Comparison of pretraining objectives added to standard baselines. Pretraining the encoder on the proxy task for policy learning actively degrades performance, whereas our GAP objective yields substantial improvements.

D. Ablation on GAP objective

We ablate the three components of the GAP loss on the Square task with 30 demos. Numerical results are reported in Table III. Removing $\mathcal{L}_{div}$ causes all keypoints to collapse into the centroid of masks, which causes the loss of orientation and boundary information. Removing $\mathcal{L}_{spread}$ prevents keypoints from reaching the object's geometric boundaries, resulting in a tightly clustered, brittle representation. Both degradations reduce downstream policy success, confirming that all three loss terms—centroid alignment, spread, and diversity—are strictly necessary to fully regularize the bottleneck.

Finally, we also test whether GAP's effectiveness stems from the geometric structure of the proposed loss, or merely from exposure to additional proxy data. While the cross-simulator experiment already rules out simulator-specific data leakage as the primary driver of performance, it does not control for the total amount of proxy data seen. To isolate the contribution of the pre-training objective itself, we compare GAP against baselines that are pre-trained on the same proxy demonstrations but in an end-to-end fashion with a diffusion head.

Specifically, we pre-train both the unregularized bottleneck (Spatial Softmax) and AFA directly on the proxy task using action-supervised imitation, giving them access to the same 100 proxy demonstrations used by GAP — plus the expert action labels and full simulator state, which GAP never requires. This setup deliberately provides a favorable advantage to the baselines, ensuring that any performance gap can be attributed to the geometric pre-training objective rather than data quantity or domain exposure.

Importantly, in this experiment, we only transfer the pre-trained vision encoder f_ϕ and the adapter f_A, deliberately withholding the pretrained diffusion policy, in order to isolate the contribution of the pooling layer pre-training.

Table IV shows an interesting result about the failure mode of standard pretraining. Although all the baselines reach 100% success rate in the trivial *Lift* task, when transferred to the downstream task, the pooling layer fails to adapt. GAP,

in contrast, fully exploits its pretraining, obtaining a 12% improvement over the baseline.

While real-world tasks are not analyzed in this work for the sake of space, and are left for extension, we present a qualitative snapshot depicting zero-shot application of GAP pretraining to a real-world video (no policy learning) in Figure 5.

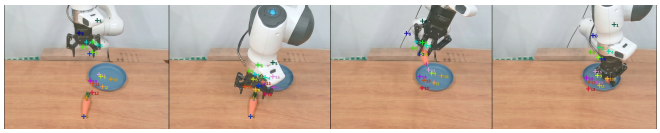


Fig. 5: **VC-1 backbone with GAP pretrained spatial pooler in the wild.** We use one video from [23] of a real-world robot performing a pick and place task and apply the GAP pretrained model (Lift task of Robomimic) to the video. This qualitatively shows a very good initialization of the keypoints even in the sim-to-real scenario, which vouches for good transfer of results in the real world.

V. Conclusion

This paper identifies *bottleneck collapse* as a key failure mode of frozen-VFM pipelines for few-shot visuomotor imitation learning: when only few demonstrations are available, the spatial pooling layer tends to overfit, losing object-centric geometric grounding and producing brittle representations. Empirically, we observe that highly parameterized semantic poolers (e.g., AFA) can drift toward diffuse, unstable attention, while unregularized Spatial Softmax can lock onto arbitrary visual shortcuts. To prevent this, we introduce **Geometric Anchor Pre-training (GAP)**, which regularizes the spatial bottleneck *before* downstream policy learning using a geometric objective composed of centroid alignment ($\mathcal{L}_{center}$), geometric spread ($\mathcal{L}_{spread}$), and keypoint diversity ($\mathcal{L}_{div}$). Pre-training the adapter once on a simple, cheap proxy task (RoboMimic `LiftCube`) produces transferable geometric anchors that can be reused across tasks and simulators. Across four tasks, three backbone architectures, and three proxy visual domains, GAP consistently outperforms all baselines, improving both data efficiency and robustness under domain shift.

Limitations. GAP relies on ground-truth object segmentation masks during proxy pre-training, which typically requires simulator access. In fully real-world settings without a simulator counterpart, masks would need to be obtained from an off-the-shelf segmentation model (e.g., Segment Anything), which may introduce noise and bias into the geometric supervision [24]. In addition, while our proxy-domain ablations indicate robustness to simulator and table changes, we have not yet validated GAP on deformable objects or highly irregular geometries, where centroid-based supervision may be less informative.

Future Work. Our results suggest a clear complementarity between frozen VFMs and GAP: VFMs provide strong semantic representations, while GAP provides precise geometric grounding tied to object structure. A natural extension is to explicitly fuse these signals by combining VFM semantic embeddings with GAP's coordinate-based spatial representation to obtain policies that are simultaneously semantically robust and geometrically precise. A second direction would be study the effectiveness of GAP on text-conditioned models.

References

[1] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.

[2] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, V. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil, *et al.*, "Where are we in the search for an effective robot motor control foundation model?" in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[3] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "DINOv2: Learning robust visual features without supervision," *Transactions on Machine Learning Research*, 2024.

[4] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 512–519.

[5] M. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova, "Tokenlearner: Adaptive space-time tokenization for videos," *Advances in neural information processing systems*, vol. 34, pp. 12 786–12 797, 2021.

[6] N. Tsagkas, A. Sochopoulos, D. Danier, S. Vijayakumar, A. Kouris, O. Mac Aodha, and C. X. Lu, "Attentive feature aggregation or: How policies learn to stop worrying about robustness and attend to task-relevant visual cues," *arXiv preprint arXiv:2511.10762*, 2025.

[7] N. Tsagkas, A. Sochopoulos, D. Danier, C. X. Lu, and O. M. Aodha, "The temporal trap: Entanglement in pre-trained visual representations for visuomotor policy learning," 2025. [Online]. Available: https://arxiv.org/abs/2502.03270

[8] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, F.-F. Li, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 1678–1690.

[9] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T.-k. Chan, *et al.*, "Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai," *arXiv preprint arXiv:2410.00425*, 2024.

[10] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3M: A universal visual representation for robot manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 892–909.

[11] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," *arXiv preprint arXiv:2203.06173*, 2022.

[12] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, *et al.*, "Ego4D: Around the world in 3,000 hours of egocentric video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 995–19 012.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[14] Z. Zhuang, R. Wang, N. Ingelhag, V. Kyrki, and D. Kragic, "Enhancing visual domain robustness in behaviour cloning via saliency-guided augmentation," in *Conference on Robot Learning*. PMLR, 2025, pp. 4314–4331.

[15] 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*. PMLR, 2021, pp. 726–747.

[16] S. Suwajanakorn, N. Snavely, J. J. Tompson, and M. Norouzi, "Discovery of latent 3d keypoints via end-to-end geometric reasoning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[17] L. Tang, M. Jia, Q. Wang, C. P. Phoo, and B. Hariharan, "Emergent correspondence from image diffusion," *Advances in Neural Information Processing Systems*, vol. 36, pp. 1363–1389, 2023.

[18] S. Wang, J. You, Y. Hu, J. Li, and Y. Gao, "SKIL: Semantic keypoint imitation learning for generalizable data-efficient manipulation," *arXiv preprint arXiv:2501.14400*, 2025.

[19] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kPAM: Keypoint affordances for category-level robotic manipulation," in *The International Symposium of Robotics Research*. Springer, 2019, pp. 132–157.

[20] W. Gao and R. Tedrake, "kPAM-SC: Generalizable manipulation planning using keypoint affordance and shape completion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6527–6533.

[21] X. Jia, Q. Wang, A. Wang, H. A. Wang, B. Gyenes, E. Gospodinov, X. Jiang, G. Li, H. Zhou, W. Liao, *et al.*, "PointMapPolicy: Structured point cloud processing for multi-modal imitation learning," in *Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025.

[22] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, *et al.*, "Perceiver io: A general architecture for structured inputs & outputs," in *International Conference on Learning Representations*, 2022.

[23] Y. Fang, Y. Yang, X. Zhu, K. Zheng, G. Bertasius, D. Szafir, and M. Ding, "Rebot: Scaling robot learning with real-to-sim-to-real robotic video synthesis," *arXiv preprint arXiv:2503.14526*, 2025.

[24] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.