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

Davide Buoso[1], Andrea Protopapa, Stefano Di Carlo, Francesca Pistilli, 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).**

I. Introduction

Contact-rich robot manipulation, where agents must skillfully interact with the physical world, requires sub-centimeter geometric precision. While Imitation Learning (IL) has demonstrated very effective for acquiring these behaviors [1], learning reliable visuomotor policies from scarce demonstrations remains an open challenge. On the other hand, recent Vision Foundation Models (VFMs) provide excellent high-dimensional sensing, however, prioritizing global semantic invariance, they inherently suppress details required to locate stable affordances and extremities for physical manipulation.

A common recipe to solve this problem is to freeze a VFM and train a lightweight spatial bottleneck to compress dense feature maps. Recently, the literature is leaning toward highly parameterized, unstructured attention mechanisms (e.g., TokenLearner [2], Attentive Feature Aggregation [3]) to filter this visual data. However, our experiments highlight a critical issue: in data-scarce, this high capacity is actually a liability. When forced to learn precise contact locations from a handful of demonstrations, unstructured poolers suffer from what we define as "bottleneck" collapse. Rather than learning stable, object-centric geometry, they lock onto easy-to-fit visual shortcuts (like background textures), rendering the downstream policy brittle to minor test-time changes. To address this, we propose Geometric Anchor Pre-training (GAP), a structure-preserving learned model designed to explicitly inject rigid, physics-relevant spatial priors into the visual pipeline before downstream policy learning. The key intuition behind GAP is that the geometric structures necessary for contact—object extent, salient extremities, and stable spatial support—are transferable and can be learned independently from the task. Thanks to an action-free warm-up stage on a cheap simulated proxy task, GAP forces the spatial adapter to produces stable geometric anchors that serve as a reliable coordinate interface when learning the downstream policy. In summary, our contributions are:

- Identifying bottleneck collapse as a fundamental failure of unstructured attention poolers in few-shot, contact-rich manipulation.
- Introducing GAP, a mask-supervised proxy pretraining strategy that enforces explicit geometric structure before any policy learning begins.
- Demonstrating that GAP significantly outperforms baseline methods on precision-heavy tasks and exhibits robust sim-to-real and cross-environment transfer.

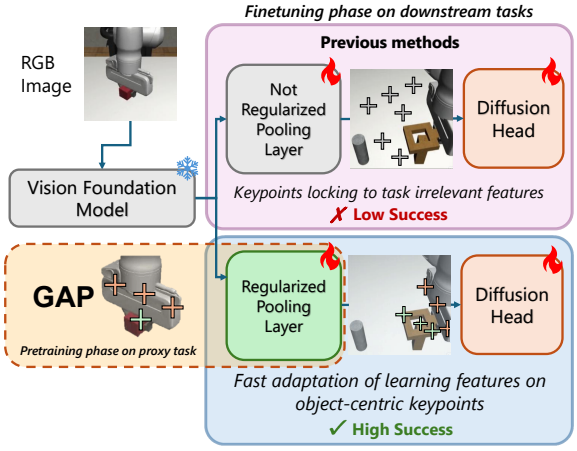

Fig. 1: **Geometric Anchor Pretraining.** GAP is a pretraining strategy applied to the spatial pooling layer on a cheap proxy task, which consistently outperforms other pooling techniques when data is scarce.

[1] Corresponding author: Davide Buoso (davide.buoso@polito.it)

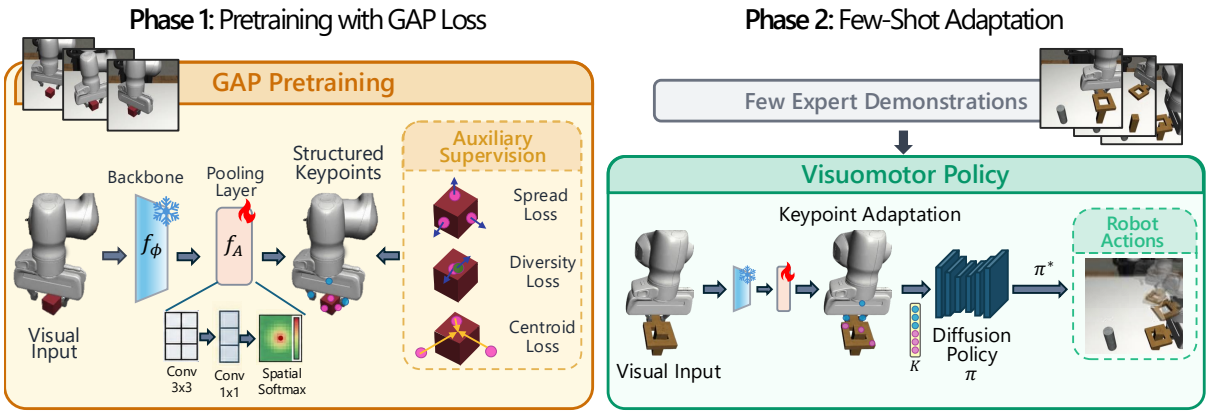

Fig. 2: **Method Overview.** 1. The spatial pooling layer extracts keypoints from the semantic pretrained backbone (frozen). GAP supervises this layer with the proposed loss, providing geometric grounding for policy learning. 2. Backbone and warmed-up pooling layer are then used to generate the input 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.

II. Methodology

Geometric Anchor Pretraining (GAP) is a pretraining strategy designed to extract object-centric geometric priors for visuomotor IL. 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. In this work, we fix the downstream policy architecture as a **Diffusion Policy** [1]. 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.

A. The Spatial Adapter

To extract robust semantic features, we use frozen pretrained backbones f_ϕ (e.g. [4]). 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 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) [5] module to convert these maps into K 2D spatial coordinates, which we define as our candidate *keypoints* $P_t = \{p_{k,t}\}_{k=1}^{K}$.

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. 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.

B. 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 [6]— the simplest task available in the benchmark—in which a robot is tasked to reach, grasp, and lift a randomly positioned cube.

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:

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

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, we enforce the spatial variance of the keypoints σ_p to match the normalized object scale σ_{target}:

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

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

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.

	Robomimic: Can					Robomimic: Square Nut				
Method	**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**

δ_{min} between any pair of keypoints:

$$\mathcal{L}_{div} = \frac{1}{K}\sum_{k=1}^{K}\left[\max\left(0, \delta_{min} - \min_{j\neq k}\|p_{k,t} - p_{j,t}\|_2\right)\right]^2 \tag{3}$$

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 ($\mathcal{L}_{div}$).

$$\mathcal{L}_{GAP} = \lambda_c\mathcal{L}_{center} + \lambda_s\mathcal{L}_{spread} + \lambda_d\mathcal{L}_{div} \tag{4}$$

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. When transitioning from a proxy task to novel, multi-object environments (e.g., `StackCube`, `SquareNut`), these pre-trained subsets deploy as independent semantic trackers. We observe that keypoints rapidly re-anchor to novel geometries.

C. 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.

III. Experiments

To validate the necessity of explicit geometric structure, we evaluate GAP on tasks requiring sub-centimeter physical precision: `Square Nut Assembly` and the long-horizon `Tool Hang` [6], along a more general pick-and-place `Can`. We compare our pretrained adapter (GAP) against a fully fine-tuned End-to-End (E-E) ResNet50 baseline, an unregularized Spatial Softmax-based pooler, and the state-of-the-art attention pooler, AFA [3]. Our quantitative results (Table I) confirm that unregularized poolers suffer from bottleneck collapse when data is scarce. On the `Square Nut` task with 15 demonstrations, AFA achieves only 15% success, matching the end-to-end baseline, while GAP achieves 20%. The gap widens significantly as tasks become more complex: on `Tool Hang` with 50 demos, E-E fine-tuning achieves 33%, AFA achieves 45%, and GAP reaches 63%. Unstructured attention (AFA) frequently overfits to spurious visual cues, whereas GAP's rigid spatial priors allow the downstream policy to quickly anchor to physical object boundaries. Extensive loss ablations confirmed that all three spatial objectives—centroid alignment ($\mathcal{L}_{center}$), spread ($\mathcal{L}_{spread}$), and diversity ($\mathcal{L}_{div}$)—are strictly necessary to prevent representation collapse. Furthermore, cross-simulator experiments (pre-training on `LiftCube` in Robomimic, testing on `StackCube` in ManiSkill [7]) showed great improvements over the baselines, We also explored different pre-training environments for `StackCube`, which resulted in no statistically significant performance gain compared to cross-domain pre-training, proving that GAP learns a domain-agnostic physical prior rather than simulator-specific textures. Most importantly for open-world deployment, this simulated geometric prior exhibits strong zero-shot sim-to-real transfer.

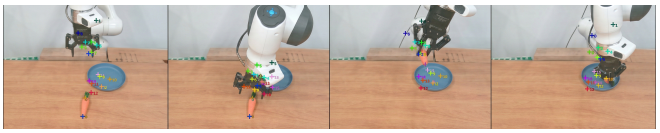

Fig. 3: **Zero-Shot Sim-to-Real Transfer.** A VC-1 backbone equipped with a GAP-pretrained spatial pooler applied to a real-world pick-and-place video [8]. The model successfully initializes and tracks the object and manipulator extremities without any real-world fine-tuning.

Figure 3 demonstrates the GAP-pretrained spatial pooler applied to a real-world, high-dimensional video of a pick-and-place task. Without any real-world actions or spatial fine-tuning, the keypoints successfully initialize and partially track the objects and manipulator extremities, providing an optimal starting point to learn a new policy.

IV. Conclusions

This paper identifies *bottleneck collapse* as a key failure mode of unstructured attention in low-data IL. By enforcing geometric priors, GAP acts as a bridge between high-dimensional visual semantics and precise geometric control. By grounding visual features into stable, physical coordinates, GAP directly advances the goal of rigorous robotic perception, ensuring that policies rely on robust physical structures rather than brittle visual shortcuts. However, achieving generalizable policies for open-world manipulation requires robustness to perception noise, latency, and temporal inconsistencies. Furthermore, GAP's losses are well suited to rigid-body manipulation but may be less reliable for deformable objects. This raises the challenge of designing geometric priors that remain stable when object structure changes dynamically. From a practical standpoint, the current reliance on a separate proxy pre-training step can add pipeline complexity. A highly promising direction is jointly learning these geometric priors during the main task. To achieve the necessary explicit supervision without costly manual annotation, future work could utilize an auxiliary head supervised by zero-shot foundation segmentation models (e.g., SAM) directly on the downstream data. Ultimately, our results highlight a complementary division of roles: VFMs provide semantic recognition, while GAP enables precise geometric grounding. Evolving this into unified architectures that fuse these signals will enable policies to jointly reason about open-world semantics and sub-centimeter geometric precision.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.