2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Evaluating the Out-of-Distribution Generalization of Robot Diffusion Policies under the DINOv2 Visual Encoder

Ángel Montejo[1] and Iñigo Iturrate[1]

***Abstract*— The generalizability of visuomotor policy models is crucial for their real-world usefulness in settings such as industrial environments. This is heavily impacted by the choice of visual encoder. In this paper, we integrate the DINOv2 foundation visual encoder with Diffusion Policy by designing a spatially-aware projection head, that allows the policy to shape its visual representation while benefiting from DINOv2's robust embeddings. We evaluate this in drastic out-of-distribution conditions. As success rate can be uninformative in these conditions, where failure rates are high, we present three evaluation criteria for goal-driven policies that remain informative despite task failure. Our result shows that our approach outperforms the baseline under color alterations and camera displacements. We observe promising emergent task-relevant feature tracking using the DINOv2 visual encoder for policy learning.**

I. INTRODUCTION

Diffusion Policy models (DPM) [1] have shown state-of-the-art results in visuomotor policy learning across real-world robotic manipulation tasks, as they can model multi-modal action distributions and generate temporally coherent actions from visual observations. However, they are sensitive to changes in the environment [2] that perturb the latent visual encoding and shift the policy away from the learned distribution. In industrial environments with tight tolerances, these small policy shifts can lead to task failure.

Numerous works have studied strategies to enhance the robustness of DPMs. Spatial invariance can be achieved by integrating 3D point-cloud information as input [3] or by leveraging geometry-aware representations to enable category-level generalization [4]. Other strategies focus on architectural robustness to visual or geometric transformations using, e.g., SIM(3)-equivariant networks [5] or latent-space disentanglement combined with associative memory [6]. While effective, these techniques require additional sensing hardware or task-specific engineering.

Alternatively, vision Foundation Models promise invariant and semantically rich embeddings, which could improve the generalization of visuomotor Behavior Cloning (BC). Among these, the DINOv2 foundation visual encoder [7] has been widely applied: DINOBot exploits DINOv2's global and local image features for task retrieval, action alignment, and replay, generalizing to novel objects and backgrounds in simple tasks [8]. BC-ViT uses a frozen version of DINO to extract semantically meaningful keypoints as input for visuomotor policies [9]. CAGE fine-tunes DINOv2 using LoRA, and improves data-efficiency and generalization to unseen objects and viewpoints by using causal attention and action diffusion [10]. Theia [11] fuses several Foundation Vision Models, including DINOv2, for a more robust task-agnostic visual representation. By combining DINOv2 with large language models, policies can jointly reason over visual inputs and action sequences [12], [13], [14].

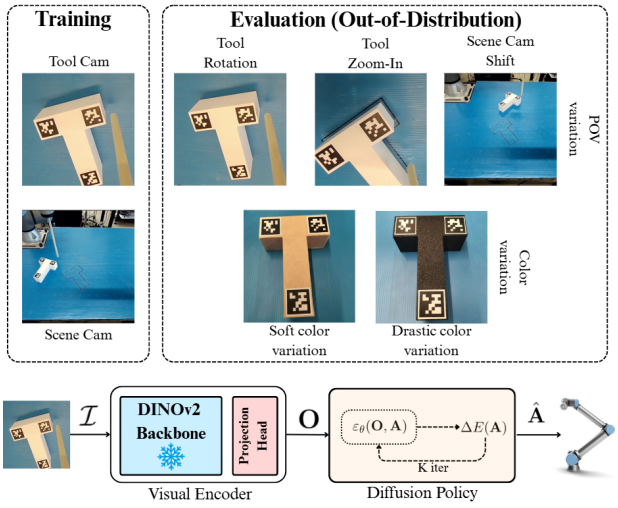

Fig. 1. We study the generalization of visuomotor Diffusion Policy Models (DPM) under drastic out-of-distribution conditions, including point-of-view and workpiece color variation. We propose to adopt the DINOv2 foundation model as the visual encoder for DPM; thus, we design a spatial projection head mapping from DINOv2 latent space to DPM observation space.

However, most studies do not rigorously evaluate policy performance under drastic zero-shot (OOD) generalization or only report success rate metrics, which can be uninformative in cases of task failure and do not provide insight into the policy's actions and the effect of the visual encoder.

In this paper, we adopt DINOv2 as the visual encoder for a DPM-based policy, aiming to improve zero-shot (OOD) generalization (see Fig. 1). Our main contributions are:

1) We design a spatial projection head that maps from DINOv2's embedding space to a representation suitable for a downstream DPM. DINOv2 serves as a frozen backbone while the head is trained along with the DPM motivated by findings that DPMs benefit from end-to-end training of the latent visual representation [1].
2) Inspired by Signal Temporal Logic (STL) [15], we propose three criteria to evaluate goal-driven policies, which are informative even in cases of policy failure.
3) We extensively compare DPM OOD generalization under the proposed DINOv2 encoder and under a

*This work was supported by Fabrikant Vilhelm Pedersen og Hustrus Legat.

[1]Angel Montejo and Iñigo Iturrate are with SDU Robotics, The Maersk Mc-Kinney Moller Insitute, Faculty of Engineering, University of Southern Denmark, 5230 Odense, Denmark; {que,inju}@mmmi.sdu.dk

ResNet18 baseline. We also evaluate the impact of image-only or hybrid image/pose observations. We do this under OOD generalization conditions including workpiece color and camera point-of-view variations.

II. METHODOLOGY

A. Diffusion Policy

A Diffusion Policy model (DPM) is a behavioral cloning approach grounded in Denoising Diffusion Probabilistic Models [16]. During training, samples $\mathbf{A}^0$ from a demonstration dataset of observation-action sequences $\{\mathbf{O}_t, \mathbf{A}_t\}$ are corrupted using *forward diffusion process* by adding K steps of noise, turning them into random Gaussian noise $\mathbf{A}^K \sim \mathcal{N}(0, I)$. The training objective is for a noise prediction network ε_θ to predict the added noise by minimizing the Mean Squared Error (MSE) between the network prediction and the noise added at each step of the process ε^k [1]:

$$\mathcal{L} = MSE(\varepsilon^k, \varepsilon_\theta(\mathbf{O}_t, \mathbf{A}_t^k, k)). \tag{1}$$

During inference, the network receives the observations $\mathbf{O}_t$, latent action sequence $\mathbf{A}_t^k$ and the current denoising step k. The predictions of $\varepsilon_\theta(\mathbf{O}_t, \mathbf{A}_t^k, k)$ are used to navigate toward more likely actions based on the observation following

$$\mathbf{A}_t^{k-1} = \alpha \cdot (\mathbf{A}_t^k - \gamma \cdot \varepsilon_\theta(\mathbf{O}_t, \mathbf{A}_t^k, k)) + \mathcal{N}(0, \sigma^2\mathbf{I}), \tag{2}$$

where α, γ and σ are noise schedulers, and are selected based on the total number of denoising steps K [1]. The term $\mathcal{N}(0, \sigma^2\mathbf{I})$ implements Stochastic Langevin Gradient Dynamics [17] to free the model from local minima and allow it to sample multi-modal actions. It should not be confused with the noise predicted by the model. Our implementation uses position control, with a sequence of Cartesian XY points as the output [1].

Diffusion Policy models operate over the action distribution conditioned on visual observations, $p(\mathbf{A}_t \mid \mathbf{O}_t)$. When operating with RGB images $\mathcal{I} \in \mathbb{R}^3$, these are embedded into a latent observation space $\mathbf{O}$ through a Visual Encoder, which is usually trained end-to-end with the policy [1]. This makes Diffusion Policy models vulnerable to visual alterations, as a numerical shift in $\mathbf{O}$ can displace the distribution $p(\mathbf{A}_t \mid \mathbf{O}_t)$ away from the learned policy, which can significantly hamper generalizability in real-world settings. To address this limitation, this study proposes the use of DINOv2 as an alternative Visual Encoder, as it has been shown to exhibit robust and invariant visual embeddings.

B. DINOv2

DINOv2 is a Foundation Vision Model that claims to generate task-agnostic invariant visual encodings capable of generalizing across a wide range of applications [7].

Its underlying architecture is a Vision Transformer (ViT) [18] with two projection heads: DINO and iBOT, implemented as Multi-Layer Perceptrons (MLP). DINOv2 is extensively pre-trained on a large and diverse dataset utilizing a teacher-student self-supervised strategy. By combining a *self-distillation* objective to train the DINO head [19] with *masked image modeling* to train the iBOT head[20], DINOv2 captures global image descriptions through DINO's [CLS] token, and local spatial descriptions of image patch tokens through the iBOT head. DINOv2 can subsequently be used zero-shot by only pre-training a simple MLP projection head that adapts to the desired task.

We aim to exploit these properties to improve the generalization of DPM. To learn visuomotor policies, it is crucial that the visual encodings preserve the spatial structure of the scene to maintain the correlation between images and manipulator movements. For this reason, our implementation uses only the patch tokens obtained from the iBOT projection head. However, the learned iBOT representations will benefit from the the fact that DINOv2 is trained to minimize both the DINO (global) and iBOT (local) objectives simultaneously, leading to more general features than using iBOT in isolation.

C. Spatial Projection Head

As is standard practice when using DINOv2 for downstream applications [7], we design a projection head appended to DINOv2. Thus, DINOv2 applies a map $h_\phi : \mathcal{I} \to \mathcal{D}$ from input images $\mathcal{I}$ into its latent space $\mathcal{D}$ and the projection head learns a map $p_\psi : \mathcal{D} \to \mathbf{O}$ into the DPM's observation space $\mathbf{O}$. As has been established, this is beneficial for the policy, as Diffusion Policy models prefer to shape their own visual representation [1].

Figure 2 shows a schematic of the proposed projection head architecture, implemented as a Convolutional Neural Network with two main elements: a residual bottleneck block [21] and a Spatial Softmax layer [22]. After an initial reshaping to restore the spatial structure of the observations, the residual bottleneck block applies a high-level transformation via a 3×3Conv in a compressed feature space with a residual connection [21]. The residual connection retains the learned features from DINOv2, while the compression increases adaptability by only retaining salient features.

The Spatial Softmax operation effectively captures and represents spatial information in CNN-based Visual Encoders for Behavior Cloning [22]. We adopt the implementation by Finn et al. [22] called *spatial soft arg-max*. For an input latent map in the form $\{C, H, W\}$, it applies a softmax operation on the spatial dimension $\{H\text{x}W\}$ of each channel C:

$$s_{cij}(a_{cij}) = \frac{e^{a_{cij}}}{\sum_{i'} \sum_{j'} e^{a_{ci'j'}}} \quad \text{with } (i,j) \in \{H, W\}, c \in C. \tag{3}$$

The most significant features are spatially located in regions with higher probability density.

A list of continuous coordinates, $f_c = (x, y)_c$, representing the average position of spatial activation in each channel weighted by the probability distribution of each element in the spatial dimension can be retrieved as

$$f_c = \left(\sum_{i,j} s_{cij} \cdot i, \quad \sum_{i,j} s_{cij} \cdot j \right). \tag{4}$$

These coordinates are linearly projected to obtain the observation $\mathbf{O}$ inputted to the DPM.

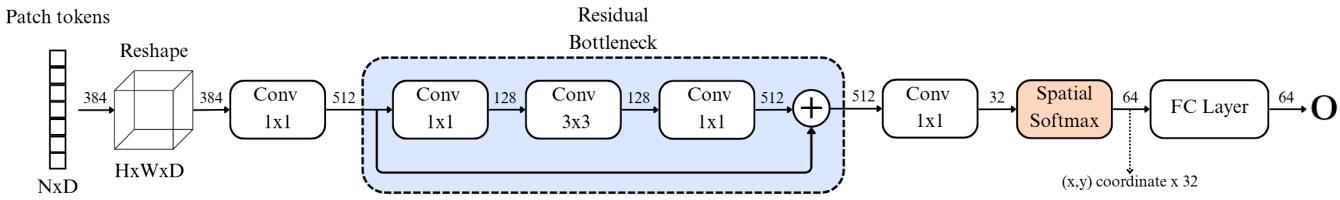

Fig. 2. Projection head appended to DINOv2 for extracting spatial features. The CNN-based head consists of a residual bottleneck block followed by a Spatial Softmax layer, which transforms DINOv2 patch tokens into a set of (x, y) coordinates representing the spatial activation in each channel of the latent feature space. Numbers on the arrows indicate the channel dimensions through the layers.

III. EXPERIMENTAL EVALUATION

We compare the generalization capabilities across color and viewpoint changes between policies trained with the proposed visual encoder and a baseline that utilizes ResNet18 trained end-to-end according to the original DPM implementation [1]. In both cases, the DPM is implemented using a Conditional UNET architecture [1] that receives the observations as conditional inputs. Additionally, we compare the impact of using image-only (I) and hybrid observations (H). Image-only observations are obtained by concatenating the two image encoding produced by independent visual encoders processing the images from each camera. Hybrid observations are formed by concatenating the image encoding with the manipulator's TCP (Tool Center Point) position (x, y) into a single observation vector.

Table I details the models under comparison. ResNet18-based encoders are trained end-to-end jointly with the DPM, while DINOv2-based encoders are frozen and only their projection head is trainable, resulting in a significantly lower number of trained parameters for DINOv2-based encoders. All policies are trained for 300 epochs under the same hyperparameters. The architectural and operational hyperparameters for the DPM are the same ase those reported for the Push-T task in the original paper [1].

TABLE I

PRESENTATION OF THE POLICIES SUBJECT TO THIS EVALUATION. THE POLICY OBSERVATION SPACE CAN BE *images-only* (-I) OR *hybrid* (-H), I.E., IMAGES AND ROBOT POSES. $\mathcal{P}_T$ AND $\mathcal{P}_L$ DENOTE THE TOTAL AND LEARNABLE NUMBER OF PARAMETERS, RESPECTIVELY.

	Observation	**Training**		
Policy	Type	$\mathcal{P}_T$	$\mathcal{P}_L$	epochs
ResNet-I	RGB Images	11.7M	11.7M	300
ResNet-H	RGB + TCP Pose	11.7M	11.7M	300
DINOv2-I	RGB Images	21.36M	0.361M	300
DINOv2-H	RGB + TCP Pose	21.36M	0.361M	300

A. Experimental Setup

We evaluate our approach on a real-world Push-T task. This is a 2 Degrees of Freedom task where a T-shaped piece must be pushed by the robot's Tool Center Point (TCP), moving in the $XY-$plane, into a target region outlined on the worktable to match the contour of the piece, allowing an approximate error margin of 2 centimeters and 10 degrees.

The hardware, displayed in Fig. 3, includes a UR5e collaborative robot (cobot) and two Orbbec Gemini335L cameras: a tool-mounted camera provides a close-up view, while a fixed scene camera provides a view of the entire workspace. The DPM architecture uses separate visual encoders for each of the two cameras. A 30 cm bar mounted to the robot end-effector allows the tool-mounted camera to keep the piece in its view while the robot pushes it. AprilTags [23] are used to track the piece's pose exclusively during evaluation; this information is not fed into the model.

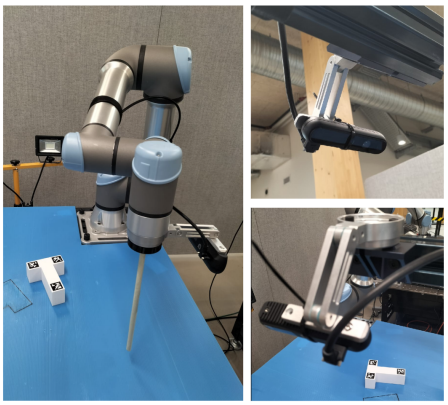

Fig. 3. Components of the physical setup. (Left) Collaborative robot **UR5e** with a 30-centimeter-long bar end-effector and a **tool-mounted camera**. (Right) A fixed **scene camera**, which oversees the entire scene.

B. Signal Temporal Logic Evaluation Metrics

To better assess the generalization capabalities of DPM, we divide the PushT task into three sub-objectives: simply moving the piece ($\bar{\mu_1}$), moving it in the correct direction ($\bar{\mu_2^t}$ and $\bar{\mu_2^r}$), and keeping it in the target region for a period of time ($\bar{\mu_3}$). To quantify this, we define three Signal Temporal Logic (STL) signals based on Kress-Gazit et al. [15]. These metrics should be interpreted hierarchically: $\bar{\mu_3}$ is weighted highest, $\bar{\mu_2^t}$ and $\bar{\mu_2^r}$ are weighted equally to each other, but lower than $\bar{\mu_3}$, and $\bar{\mu_1}$ is weighted lowest. All signals are based on the T-shape piece pose $P(t) = (p(t), w(t))$, the target pose $P^* = (p^*, w^*)$ and the time step t with $\Delta t = 1s$, and a maximun period of $\tau = 120s$ per episode.

1) Criterion 1: Motion Detection: The first criterion determines whether the piece is moved by the cobot's TCP. This reflects the visual encoders' awareness of the piece in the environment and differentiates policies that recognize the piece from those that do not:

$$\mu_1(t) = \mathbf{F}_{[0,1]}\big(|\Delta p(t)| > \delta_t \ \vee \ |\Delta w(t)| > \delta_w\big), \quad (5)$$

where $\mathbf{F}_{[0,1]}$ denotes a temporal logic operator that returns a boolean value if either a translation $\Delta p(t)$ or a rotation $\Delta w(t)$ occurs between consecutive poses of the piece. The *or* operator is denoted by $\vee$, and the thresholds δ_t and δ_w are empirically determined as $\delta_t = 5$ mm for translation and $\delta_w = 0.0873$ rad ($\approx 5°$) for rotation.

We normalize $\mu_1(t)$ by the number of samples N, yielding a motion score over τ bounded in [0,1]:

$$\bar{\mu}_1 = \frac{\sum_{t=1}^{N} \mu_1(t)}{N}. \tag{6}$$

2) Criterion 2: Motion Accuracy: Once the piece is in motion, this metric assesses the precision of the movement by measuring the alignment of the piece displacement with the intended direction. The motion is decomposed into translational accuracy $\mu_2^t(t)$ and rotational accuracy $\mu_2^r(t)$:

$$\mu_2^t(t) = \frac{\langle \vec{d}, \vec{g} \rangle}{|\vec{d}||\vec{g}|} \quad , \quad \mu_2^r(t) = \frac{\langle \vec{r}, \vec{v} \rangle}{|\vec{r}||\vec{v}|}, \tag{7}$$

where $\langle ., . \rangle$ denotes the dot product.

The metrics $\mu_2^t(t)$ and $\mu_2^r(t)$ are computed as normalized scalar projections: $\mu_2^t(t)$ projects the Euclidean translation vector $\vec{d}(t) = p(t) - p(t - \Delta t)$ onto $\vec{g} = p^* - p(t - \Delta t)$, the vector from the previous position to the target, while $\mu_2^r(t)$ projects the rotation vector $\vec{r} = \omega(t) - \omega(t - \Delta t)$ onto $\vec{v} = \omega^* - \omega(t - \Delta t)$, both in $\mathrm{SO}(3)$.

These projections quantify alignment with the target motion, ranging from -1 (opposite direction) to 1 (perfect alignment). To compute a scalar metric, we integrate the STL signals over time and normalize the result. To exclude non-motion samples, we define $\mathcal{U}^t = \{t \mid \mu_2^t(t) \neq 0\}$ and $\mathcal{U}^r = \{t \mid \mu_2^r(t) \neq 0\}$. The directional translational and rotational scores are then computed as:

$$\bar{\mu}_2^t = \frac{\sum_{t \in \mathcal{U}^t} \mu_2^t(t)}{|\mathcal{U}^t|} \quad , \quad \bar{\mu}_2^r = \frac{\sum_{t \in \mathcal{U}^r} \mu_2^r(t)}{|\mathcal{U}^r|}, \tag{8}$$

which are bounded by $[-1, 1]$, and account for the accuracy of the policy motion only if there is any movement.

3) Criterion 3: Time Within Target Area: This criterion aims to identify which policies maintain the piece correctly placed for a longer period within the evaluation time $\tau = 120$. It rewards quicker policies that keep the piece in the target position without compromising the final outcome.

We define a quantitative STL based on the SE(3) distance $d_{SE(3)}(T(t), T^*)$ between the current piece pose $T(t)$ and the target pose T^*, both expressed in homogeneous coordinates. The success margins are set experimentally to $\delta_r = 2$ cm for translation and $\delta_t = 0.1745$ rad (approx. 10 °) for rotation. This metric yields $d_{SE(3)}(T(t), T^*) > 1$ when the piece is outside of the success margin, exactly 1 when its lies on the boundary and < 1 when it is within the defined threshold. Then, $\mu_3(t)$ is defined as:

$$\mu_3(t) = \max\left(0,\, 1 - d_{SE(3)}(T(t), T^*)\right), \tag{9}$$

which approaches 1 as the part aligns with the target position and orientation, and drops to 0 when it exits the defined region. The metric is normalized to yield a in [0,1] using

$$\bar{\mu}3 = \frac{\sum_{t=1}^{N} \mu_3(t)}{N}, \tag{10}$$

which represents the time score within the target region.

Additionally, we use a *Success Rate*, defined as the ratio of successful trials out of 12 tests per case. A trial is successful if the piece is within the target area at the end of the evaluation, i.e., $\mu_3(\tau) > 0$, regardless of the previous performance of the policy.

IV. Results

Table II presents the results for the metrics described in the preceding section. The baseline condition consisted of the same visual conditions used during training, i.e., a white piece and no variation in camera position. For each conducted experiment, the policies operated over four initial conditions (poses for the piece), each one repeated for three trials, obtaining 12 evaluation episodes for each policy in each experiment. The TCP initial pose was always the same.

1) Color-space Generalization: We performed two levels of color variation: *soft color variation* utilized a light brown T-piece close in RGB values to the baseline, while *extreme color variation* used a black piece with RGB values opposite in the spectrum to the baseline.

2) Viewpoint Generalization: Three camera viewpoint variations were tested: First, the scene camera was lowered by 9 cm relative to the robot base frame, resulting in a closer perspective of the working piece. Second, the robot end-effector was shortened by 10 cm, producing a zoomed-in view of the piece for the tool-mounted camera. Third, all cobot TCP axes were rotated by +5 degrees, changing the perspective of the piece for the tool-mounted camera.

V. DISCUSSION

A. Result Analysis

We now analyze the results in Table II. For the *Baseline*, i.e., the in-distribution visual conditions, DINOv2-based policies achieve lower motion accuracy scores than ResNet-based ones, suggesting that DINOv2 policies are less refined, producing suboptimal trajectories to the goal. We hypothesize this is due to the lower number of training parameters in the DINOv2 projection head compared to the ResNet baseline (see Table I), which constrain the DPMs' flexibility to model their visual observations, particularly given the limited training time.

DINOv2 policies adapt better to visual alteration. For *Color Alteration* (see Table II), ResNet-I and DINOv2-H perform comparably under moderate shift, with DINOv2 showing a slightly higher success rate, albeit lower on-target time scores. For drastic color alteration, DINOv2 (both image and hybrid) shows higher levels of interaction with the piece, with DINOv2-H obtaining the highest motion detection and motion accuracy scores. This shows that, not only does the visual encoder identify the piece in the workspace, but the policy generates actions toward the optimal solution. In contrast, ResNet-based policies obtain much lower interaction scores, reflecting that they tend to ignore the altered piece.

TABLE II

PUSHT RESULTS UNDER THE BASLINE CONDITIONS, COLOR ALTERATIONS, AND CAMERA POINT-OF-VIEW ALTERATIONS.

Policy	$\bar{\mu}_1$	$\bar{\mu}_2^t$	$\bar{\mu}_2^r$	$\bar{\mu}_3$	**Success**
Baseline					
ResNet-I	0.2122	**+0.3833**	**+0.2945**	**0.3785**	**12/12**
DINOv2-I	**0.4146**	+0.1860	+0.1775	0.1575	8/12
ResNet-H	0.2954	+0.2920	+0.2329	0.3218	9/12
DINOv2-H	0.3711	+0.2157	+0.2405	0.2610	9/12
Moderated color alteration					
ResNet-I	0.2969	**+0.4057**	**+0.4202**	**0.3005**	9/12
DINOv2-I	**0.4083**	+0.2853	+0.2968	0.2604	9/12
ResNet-H	0.3571	+0.2230	+0.2325	0.1575	5/12
DINOv2-H	0.4020	+0.2729	+0.2824	0.1745	**10/12**
Drastic color alteration					
ResNet-I	0.1548	+0.0572	-0.0180	0	0/12
DINOv2-I	**0.3932**	-0.0238	+0.0441	0	0/12
ResNet-H	0.1169	+0.0476	-0.0254	0	0/12
DINOv2-H	0.2586	**+0.0594**	**+0.1039**	0	0/12
Scene camera translation					
ResNet-Images	0.2451	+0.0533	+0.0378	0	0/12
DINOv2-Images	0.1390	+0.0379	+0.0157	0	0/12
ResNet-Hybrid	**0.4391**	+0.0916	+0.0835	0.0017	0/12
DINOv2-Hybrid	0.3606	**+0.1310**	**+0.0841**	**0.0815**	**3/12**
On-tool camera translation					
ResNet-Images	0.0931	+0.0707	+0.0396	0	0/12
DINOv2-Images	0.2339	+0.0683	+0.0800	0	0/12
ResNet-Hybrid	**0.3824**	**+0.0818**	+0.0463	0	0/12
DINOv2-Hybrid	0.2885	-0.0248	**+0.1522**	**0.0049**	0/12
On-tool camera rotation					
ResNet-Images	0.4363	**+0.1518**	+0.1183	**0.0720**	**3/12**
DINOv2-Images	0.2990	+0.1261	+0.0601	0.0009	0/12
ResNet-Hybrid	0.4321	+0.0923	+0.0926	0.0003	0/12
DINOv2-Hybrid	**0.4874**	+0.1400	**+0.1394**	0.0151	2/12

Similar results are observed for point-of-view invariance. All visual alterations are considered drastic, as reflected in the lower success of all policies. DINOv2-H is the only policy that partially adapts to scene camera translations, with a non-zero success rate, as well as the highest motion accuracy and on-target time scores. For tool-mounted camera translations, DINOv2-H manages to place the piece on the target region, reflected by the non-zero on-target time score, albeit only for a brief period of time. Interestingly, for tool-mounted camera rotations, both ResNet-I and DINOv2-H achieve non-zero success rates, while the best-performer depends on the choice of evaluation criterion.

Note that, in all experiments, the evaluation conditions are extreme. In a more realistic scenario, some alterations and augmentations would be included the training set. However, by evaluating in drastic OOD scenarios, we highlight the effect of the different visual encoders on policy performance.

B. Emergent Behavior of DINOv2

We observe interesting emergent behavior using DINOv2-based visual encoders. As mentioned in section II-C, the proposed projection head contains an arg-max spatial softmax layer. The keypoints in this layer can be interpreted as the points of interest of each of the channels of the latent feature-space of the visual encoders.

By reprojecting these points onto the input images, as shown in Fig. 4, we observe that both DINOv2 policy types track objects relevant to the task, although the encoder was not explicitly trained for this purpose. This may explain the increased generalization capabilities of DPMs using DINOv2-based visual encoders, although the full effects of this on the DPM policies require further study.

C. Hybrid vs image-only observations

A side question of this study is whether the choice of image-only or hybrid observation space affects DPM generalization. We observe that hybrid policies tend to learn near-optimal behavior in shorter training time, as they do not need to infer the robot TCP pose from visual input.

However, combining different types of observations can be counterproductive due to *cross-modal discrepancy*, as this induces a larger input space and imposes a need to weight the different observation channels. We observed situations where hybrid policies failed to solve the task when entering areas of the space not in the training set. Image-based policies had no problem solving this kind of scenarios. This is illustrated by the difference in performance between ResNet-H and ResNet-I models across all tasks.

The distribution of keypoints in Fig. 4 reflects the above behavior. The image-only policy makes richer use of multiple points of interest, assigning several of them to task-relevant objects. In contrast, the hybrid policy shows a high concentration of keypoints near the center of the image – unused channels in the latent feature representation –, suggesting it relies heavily on the provided TCP pose instead.

We can not categorically conclude whether DPMs prefer hybrid or image-only observations. While ResNet policies seem to prefer image-only, the opposite is true for DINOv2-based policies. It remains to be determined whether an image-only policy using DINOv2-based visual encoders, given more trainable parameters in the projection head and/or more training time, could outperform its hybrid counterpart. Nevertheless, the allocation of feature points in Fig. 4 proves promising in terms of the visual generalization abilities of the DINOv2 encoder for visuomotor behavioral cloning tasks.

D. Generality of Our Results

We acknowledge several limitations of our results. The observed reduction in motion accuracy in the baseline may stem from the lower number of training parameters in the visual encoder. It remains unclear whether modifying the architecture, increasing the depth of the spatial projection head, or training with an unfrozen visual backbone would improve performance. Additionally, the simplicity of our use case prevents us from drawing conclusions about more complex tasks. We leave this analysis for future work. However, the obtained results suggest that integrating Foundation Vision Models could improve the robustness of visuomotor policies

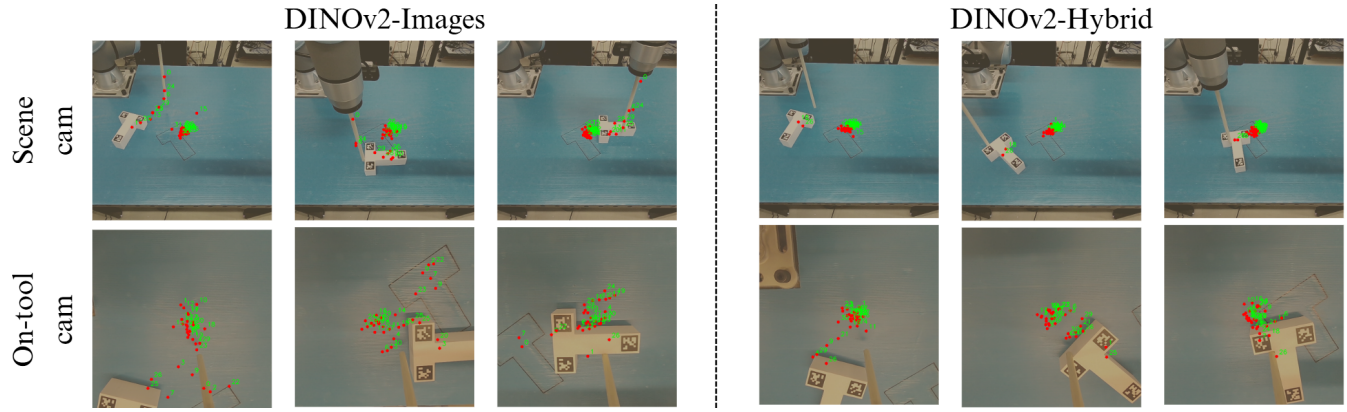

Fig. 4. Reprojection of spatial softmax keypoints using the DINOv2 visual encoder during Push-T task operation. Emergent behavior is observed as keypoints consistently align with the TCP bar, workpiece, and target area across both camera views. In the image-based policy, multiple keypoints actively track all elements, while in the hybrid case, fewer keypoints are used—many remain near the image center, indicating underuse of the visual latent space.

to significant visual variations, making them more reliable under real-world conditions.

VI. CONCLUSION

We studied the out-of-distribution generalization of Diffusion Policies combined with the DINOv2 foundation visual encoder through a spatial projection head, allowing the policy to benefit from DINOv2's invariant and generalizable embeddings. We evaluated our approach on out-of-distribution scenarios, using three Signal Temporal Logic evaluation criteria for goal-driven tasks, which remain informative even in cases of task failure. Our method outperforms the baseline under color alterations and camera displacements. The emergent allocation of visual features to task-relevant areas proves promising in terms of the generalizability of the DINOv2 visual encoder for policy learning.

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*, p. 02783649241273668, 2023.

[2] Y. Chen, D. K. Jha, M. Tomizuka, and D. Romeres, "Fdpp: Fine-tune diffusion policy with human preference," *arXiv preprint arXiv:2501.08259*, 2025.

[3] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," *arXiv preprint arXiv:2403.03954*, 2024.

[4] Y. Wang, G. Yin, B. Huang, T. Kelestemur, J. Wang, and Y. Li, "Gendp: 3d semantic fields for category-level generalizable diffusion policy," in *8th Annual Conference on Robot Learning*, vol. 2, 2024.

[5] J. Yang, Z.-a. Cao, C. Deng, R. Antonova, S. Song, and J. Bohg, "Equibot: Sim (3)-equivariant diffusion policy for generalizable and data efficient learning," *arXiv preprint arXiv:2407.01479*, 2024.

[6] S. Batra and G. Sukhatme, "Zero-shot visual generalization in robot manipulation," *arXiv preprint arXiv:2505.11719*, 2025.

[7] 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," *arXiv preprint arXiv:2304.07193*, 2023.

[8] N. Di Palo and E. Johns, "Dinobot: Robot manipulation via retrieval and alignment with vision foundation models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2798–2805.

[9] W.-D. Chang, F. Hogan, D. Meger, and G. Dudek, "Generalizable imitation learning through pre-trained representations," *arXiv preprint arXiv:2311.09350*, 2023.

[10] S. Xia, H. Fang, C. Lu, and H.-S. Fang, "Cage: Causal attention enables data-efficient generalizable robotic manipulation," *arXiv preprint arXiv:2410.14974*, 2024.

[11] J. Shang, K. Schmeckpeper, B. B. May, M. V. Minniti, T. Kelestemur, D. Watkins, and L. Herlant, "Theia: Distilling diverse vision foundation models for robot learning," *arXiv preprint arXiv:2407.20179*, 2024.

[12] Z. Hou, T. Zhang, Y. Xiong, H. Duan, H. Pu, R. Tong, C. Zhao, X. Zhu, Y. Qiao, J. Dai, *et al.*, "Dita: Scaling diffusion transformer for generalist vision-language-action policy," *arXiv preprint arXiv:2503.19757*, 2025.

[13] J. Yang, W. Tan, C. Jin, K. Yao, B. Liu, J. Fu, R. Song, G. Wu, and L. Wang, "Transferring foundation models for generalizable robotic manipulation," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2025, pp. 1999–2010.

[14] Q. Yang, M. C. Welle, D. Kragic, and O. Andersson, "S 2-diffusion: Generalizing from instance-level to category-level skills in robot manipulation," *arXiv preprint arXiv:2502.09389*, 2025.

[15] H. Kress-Gazit, K. Hashimoto, N. Kuppuswamy, P. Shah, P. Horgan, G. Richardson, S. Feng, and B. Burchfiel, "Robot learning as an empirical science: Best practices for policy evaluation," *arXiv preprint arXiv:2409.09491*, 2024.

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

[17] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28th international conference on machine learning (ICML-11)*. Citeseer, 2011, pp. 681–688.

[18] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[19] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9650–9660.

[20] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, "ibot: Image bert pre-training with online tokenizer," *arXiv preprint arXiv:2111.07832*, 2021.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] 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.

[23] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 3400–3407.