

A scalar per patch from pre-trained ViTs enables fast moving navigation in the real world

Steeven Janny, Leonid Antsfeld, and Christian Wolf

NaverLabs Europe, Grenoble, France
name.surname@naverlabs.com

Abstract. Trained policies for real-world robotics rely on computer vision components, typically in the form of pre-trained visual encoders. These encoders are an essential component and it has been shown that their power does not emerge from training on robotics downstream losses alone. Pre-training with auxiliary losses in the form of computer-vision pre-text tasks is a defining factor and heavily conditions agent performance in robotics tasks. In this unprecedented large-scale study, we ran 966 navigation episodes of static point goal navigation in a real-world building for 24km and asked which components really matter for the computer vision aspects of robotics: we evaluate state-of-the-art visual encoders in realistic conditions. We explore the usefulness of heterogeneous multi-teacher distillation leading to encoders with multiple different and complementary skills. We investigate how much information from these encoders is necessary for robotics by bottlenecking them in a principled and “spatially useful” way and we show that this leads to the emergence of interpretable features linked to affordances. We also argue that training policies on RGB data alone does not lead to an optimal usage of visual features and show this by finetuning policies pre-trained on privileged information. All in all, we paint a more complete picture of what aspects of computer vision are relevant for real-world navigation.

Keywords: Visual encoders, robotics, navigation

1 Introduction

The quest for general-purpose physical intelligence has historically been partitioned into two distinct silos: computer vision, tasked with “knowing what is where” and robotics, tasked with mapping those perceptions to motor commands. Computer vision researchers focused on intermediate representations and tasks, such as 3D reconstruction, semantic segmentation, and object detection or tracking, while the robotics community designed controllers to feed on the corresponding representations and features.

Recent discussions in the field¹ suggest that computer vision, as an independent discipline focused on intermediate mappings, may be reaching its “*Bitter*

¹ See, for instance, https://www.vincentstzmann.com/blog/bitter_lesson_of_cv

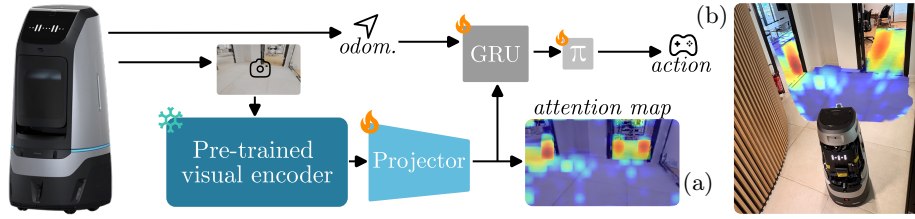


Fig. 1: We provide an analysis of pre-trained vision encoders for robotic perception in a large-scale study of 966 navigation episodes of fast-moving agents in a real environment. (a) we show that interpretable spatial maps emerge through a new proposed bottlenecking layer. These maps seem to encode “affordances” and navigation performance is maintained even if they are the only features available to the agent; (b) Integration of multiple maps into a bird’s-eye-view.

Lesson” moment². Some emerging consensus suggests that the future of perception lies in end-to-end perception-action loops, where the utility of a visual representation is measured solely by its contribution to “intelligent action”.

While we agree with this general but probably very long-term trend, current work on robotics heavily builds on the foundations of computer vision. Even the most generalist agents and VLAs [35, 74, 77] are not trained from scratch on downstream robotics data, but rely on pretrained vision encoders whose features are injected into the final model in several training steps. Given the sparsity of robotics data and the difficulty of obtaining it, it is unclear when this will change. In the meantime robotics strongly relies on computer vision in the form of pre-text tasks, pre-training data, and as we will show in this paper, the combination of multiple vision experts.

In this work we provide a large-scale study consisting of an unprecedented 966 indoor navigation episodes (and many more performed as preparation) with a real robot in realistic conditions. By that we mean fast-moving agents navigating at 0.7 m s^{-1} and which can be deployed to solve real-life tasks. The agents only use RGB input for navigating and do not have access to Lidar or depth cams. Moving at this speed requires robust perception and decision making. The goal of this work is to study the role of visual perception in navigation from several angles and we ask the following questions:

- (1) Can perception be learned from actions alone? We show that direct training from scratch of visual encoders leads to bad performance, ie. difficult transfer to real physical robots, as the learning algorithm overfits the learned representations to visual artifacts and spurious correlations. We argue that the usage of pre-trained vision encoders combined with realistic motion modelling [11, 38] has closed the sim2real gap to negligible size.
- (2) What kind of pre-training is most suitable for navigation downstream tasks? We compare several different pre-training strategies including MAE [30, 52] and the hybrid combination of self-distillation [15] and iBOT [86] used in *DinoV2* [58] and *DinoV3* [75].

² <http://www.incompleteideas.net/IncIdeas/BitterLesson.html>

- (3) Can robotics / navigation tasks benefit from complementary and heterogeneous capabilities stemming from training on different high-level tasks? We explore perception with highly versatile encoders like *AM-Radio* [32] and *Dune* [67] distilled from multiple sources including 3D reconstruction with *MASt3R* [46], human mesh recovery with *MHMR* [5] and general vision features from *DinoV2* [58].
- (4) How much information encoded in these visual features is necessary for the downstream task? We show that the information flow can be restricted to only 1 scalar per image patch. Furthermore, by shaping the information flow such that it corresponds to spatial attention distributions, interpretable features and affordances emerge without loss of navigation quality.
- (5) Is privileged sensor information helpful during training? We show that pre-training agents with RL on Lidar-like input with a full 360° view can provide gains when finetuning and deploying them with input from forward facing RGB only.

In summary, this study offers a comprehensive analysis of how various visual pre-training strategies dictate the navigational performance and reliability of fast-moving robots in complex settings.

2 Related works

Visual encoders in robotics originate from vision transformers trained with masked image modelling [30]. This approach can be adapted to robotic settings by training on large collection of ego-centric images gathered across platforms (*VC1* [52], *MVP* [60]). Self-distilled and iBOT-based models [86], such as the *Dino* series [15, 58, 75] are widely adopted due to their strong transfer performance and availability in multiple scales compatible with on-board computation. In particular, *DinoV2* has become a common backbone for downstream robotic tasks [28, 82, 84], motivating world models that operate in its feature space, both at the input and output levels (e.g. *Dino-WM* [85] and *Dino Foresight* [40]). Complementary to general-purpose encoders, models tailored to 3D geometry (e.g. *DUST3R* [81], *MASt3R* [46], *VGGT* [80]) provide spatially structured features that benefit navigation and manipulation tasks [9, 49]. Finally, multiple specialized encoders can be distilled into a single ViT to produce compact representations aggregating diverse capabilities (e.g., *AM-Radio* [32], *Dune* [67]).

Visual navigation has been classically solved in robotics using explicit modelling [13, 51, 53], which requires solutions for mapping and localization [12, 45, 78], for planning [44, 71] and for low-level control [27, 66]. These methods depend on accurate sensor models, filtering, dynamical models and optimization. End-to-end trained models directly map input to actions and are typically trained with RL [36, 55, 79, 84] or IL [23]. They learn flat representations [11], occupancy maps [17], semantic maps [16, 56], latent metric maps [7, 33, 59], topological maps [6, 18, 72], explicit episodic memory [20, 24, 26, 64], implicit representations [54] or navigability [10]. Our study targets recurrent policies trained end-to-end additional motion model in simulators equipped with realistic motion.

Querying representations in computer vision and embodied AI Feature maps produced by CNN- or ViT-based encoders are commonly extracted through flattening (aggregation of patch/cell embeddings), global average pooling, ROI-pooling for object-centric tasks (introduced in *Faster-R-CNN* [65]), or dedicated class tokens that can be generalized to attention-based read-outs (e.g., *DETR* [14]). Multi-scale features are combined in *Feature Pyramid Networks* [47] and *Hypercolumns* [29], while place recognition relies on *NetVLAD* / *VLAD*-like aggregation [4], where local descriptors are assigned to learned centroids and residuals accumulated. More closely related to our approach, several works query frozen visual representations with cross-attention and learned embeddings, such as *Slot Attention* [50] and *QFormer* [22]. In embodied AI, end-to-end policies typically reuse similar pooling strategies before passing features to recurrent or transformer-based policies. Goal-conditioned tasks (e.g., image-goal navigation) rely on temporal stacking, channel-wise stacking, or cross-attention [9, 57, 73, 76]. Vision-language-action models project visual features into the token space of VLMs or LLMs by representing images as sets of CNN or ViT tokens (e.g., *Open-VLA* [41], *SmolVLA* [74], *Octo* [77], π_0 [8]), which are then integrated by attention layers operating over single or multiple time steps. Prior work [69] evaluates mid-level priors solely in simulation without real-world validation, and [1] study sim-to-real representation transfer but rely on handcrafted features like depth and navigability. In contrast, we use general-purpose pretrained ViTs and compress their features using learned projectors without task-specific engineering.

3 Experimental Setup

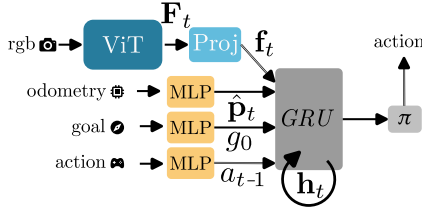


Fig. 2: The navigation agent used in the experimental setup uses recurrent memory. It is based on [38] but uses RGB only input to collect information on the scene structure.

For this study we build on the agent from Janny et al. [38], which has been optimized for real-world scenarios and fast-moving agents trained end-to-end in simulation, with a primary focus on minimizing the sim-to-real gap. To make the paper self-contained, we will describe the agent in this section. We choose static point-goal navigation as a standard task for *visual navigation*: It lets us isolate the impact of vision on policy learning without additional complexity from language ground-

ing or image-goal matching. Although broader tasks require different reasoning mechanisms, static point-goal captures core navigation skills such as obstacle perception, scene understanding, short-horizon memory, and path execution.

At each discrete time step t , the agent receives sensor readings, which include an RGB image $\mathbf{I}_t$. In the original work [11, 38], the agent also received a Lidar-like “scan” vector which represents depth ranges analogous to Lidar data, but which was actually extracted from four directional *RealSense* depth sensors.

In our work, we do not use this sensor information directly related to the 3D structure of the environment and **show that the agent can navigate from RGB being the only input alone**, without Lidar or depth.

The task is *Static PointGoal* in real environments, i.e., the agent’s objective is to reach a target defined by polar coordinates relative to its pose at episode start. In other words, the goal vector is *not* updated when the agent moves and stays constant, for this reason we denote it as $\mathbf{g}_0$. The agent operates without an external map, relying instead on onboard ego-motion estimation. To facilitate goal-tracking, we provide a localization signal relative to the starting position through odometry $\hat{\mathbf{p}}_t$ derived from wheel encoders and IMUs. This signal includes position and velocity estimates.

The agent runs at 3Hz and outputs actions from a discretized action space consisting of 28 velocity pairs (a_v, a_ω) . The linear velocity a_v is sampled from $\{0, 0.2, 0.4, 0.7\} \text{ m s}^{-1}$, while angular velocity a_ω ranges from $\{-3, \dots, 3\} \text{ rad s}^{-1}$ in integer increments. To handle the partial observability of the environment, the agent maintains a recurrent hidden state $\mathbf{h}_t$:

$$\begin{aligned} \mathbf{F}_t &= v(\mathbf{I}_t) && // \text{ Perception / Vision} \\ \mathbf{h}_t &= d(\mathbf{h}_{t-1}, \mathbf{F}_t, \mathbf{g}_0, \hat{\mathbf{p}}_t, \mathbf{a}_{t-1}) && // \text{ State update} \\ \mathbf{a}_t &= \pi(\mathbf{h}_t) && // \text{ Policy} \\ \hat{\mathbf{g}}_t &= l(\mathbf{h}_t) && // \text{ Auxiliary loss f. goal prediction} \end{aligned} \tag{1}$$

In this model, the visual input is processed by a pre-trained ViT v extracting features $\mathbf{F}_t$ from the input RGB image $\mathbf{I}_t$, which will be main object of study in this work. Memory is updated by d , a two-layer (GRU) [21], previous actions are processed with an embedding layer and other state inputs are mapped through MLPs. The final policy π is implemented as a linear head. As in [38], since the goal $\mathbf{g}_0$ is fixed in the initial frame, we assist the agent’s spatial reasoning by adding an auxiliary linear head l that predicts the instantaneous relative goal $\hat{\mathbf{g}}_t = l(\mathbf{h}_t)$, supervised during training via simulation ground truth.

The model is trained with RL, in particular PPO [70], and with a reward inspired by [19, 38], $r_t = R \cdot \mathbb{I}_{\text{success}} - \Delta_t^{\text{Geo}} - \lambda - C \cdot \mathbb{I}_{\text{collision}}$, where $R=2.5$, Δ_t^{Geo} is the gain in geodesic distance to the goal, a slack cost $\lambda=0.01$ encourages efficiency, and a cost $C = 0.1$ penalizes each collision without ending the episode.

Modelling Realistic Dynamics in Real Environments. We corroborate the importance of modelling realistic motion dynamics in simulation introduced in [11, 38]. We replace the standard “teleportation” motion in the Habitat simulator [68] with a second-order dynamical motion model, whose parameters were identified from real-world trajectories with our robot. This ensures that simulation reflects authentic inertial effects: the simulated robot accelerates, brakes and turns as the physical one would in the same conditions. **Realistic motion modelling is crucial:** since the robot moves at 0.7 m s^{-1} and takes decisions at 3 Hz, it needs to be able to correctly anticipate its future trajectory, as any potential corrections for an action can be done after 333 ms at the earliest.

3.1 Experimental Protocol and Metrics

We train the agent with the Habitat simulator [68], modified for realistic motion as described above, and on the *HM3D* [63] training set of 800 scenes 3D-scanned by a Matterport scanner. We evaluate our models across two primary domains to balance statistical significance with real-world validity:

Physical Deployment (“Real”) : Tests are conducted on a “*Rookie*” robot (cf. Fig. 1) within a designated office environment. We report results across 14 episodes per experiment and we systematically report $\text{avg} \pm \text{std-dev}$ over 3 runs, ie. 42 real navigation episodes per evaluated configuration.

Augmented Simulation (“Sim”) : We utilize the simulator integrated with our identified dynamics model, which makes evaluations highly predictive of behavior in the real environment. Evaluations are performed on the HM3D [63] validation set (2,500 episodes).

All tables have color-coded backgrounds indicating the evaluation setting.

Performance metrics. Success is defined by the agent reaching within 0.2m of the goal in simulation (with zero commanded motion) or 1m in real-world trials. We report the usual metrics for navigation [3]: (1) *Success Rate* (SR), ie. the ratio of successfully completed episodes (2) spatial efficiency as *Success weighted by Path Length* (SPL), (3) temporal efficiency as *Success weighted by Completion Time* (SCT) [83]:

$$SR = \frac{1}{N} \sum_{i=1}^N \mathbb{I}_i, \quad SPL = \frac{1}{N} \sum_{i=1}^N \mathbb{I}_i \frac{\ell_i^*}{\max(\ell_i, \ell_i^*)}, \quad SCT = \frac{1}{N} \sum_{i=1}^N \mathbb{I}_i \frac{c_i^*}{\max(c_i, c_i^*)} \quad (2)$$

where $\mathbb{I}_i$ is a success indicator for the i^{th} episode, ℓ_i and ℓ_i^* are the navigated and shortest path, respectively, and c_i and c_i^* are the elapsed time and the theoretically optimal time, respectively. Due to real-world hardware and sensor limitations, we use different success thresholds: simulation uses ground-truth positioning with a 0.2 m threshold and real-world evaluation relies on noisy onboard sensors and AMCL with a 1.0 m threshold.

4 Experimental Results

Lineup of vision models. We tested perception on several widely used vision encoders and state-of-the-art models, if not stated otherwise as ViT/Base:

DinoV2 [58] is trained on large procedurally curated dataset of 142M images with several self-supervised losses, including self-distillation from Dino [15] and the patch-wise iBOT [86]. It provides general features that are highly transferable across diverse downstream tasks without fine-tuning.

DinoV3 [75] scales its predecessor up to larger model architectures and even more massive datasets, incorporating long-sequence training and improved stability during the distillation process.

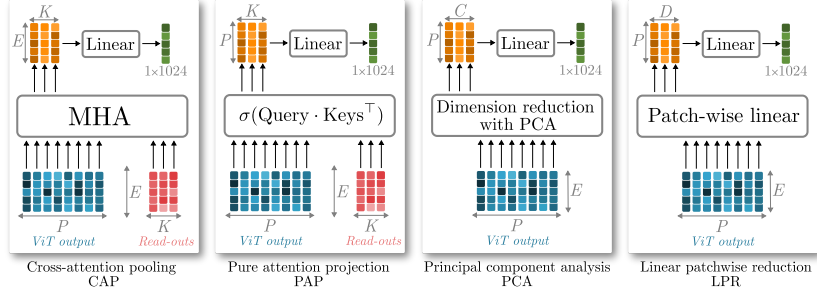


Fig.3: Architecture of projection heads – inserted between the dense, high-dimensional output of the pretrained visual encoders, and the recurrent memory of the agent. The LPR and PAP maintain spatial information by computing patch-wise embeddings. The PCA projects each patch on the C principal components, with no learnable parameter apart from the final linear layer.

VC1 [52] uses MAE [30] for training. It specifically targets robotics in its datasets combining 4.3M images stemming from ImageNet and 4000h of egocentric video from 7 sources.

AM-RADIO v2.5 [32] is distilled from heterogeneous sources such as *CLIP* [61], *DinoV2* [58], and *SAM* [43], aiming to combine the best of semantic, geometric, and segment-level features into a single efficient backbone.

DUNE [67] is also a distilled model but focuses on unifying representations across diverse environments by distilling from multiple specialized teacher models into a compact, robust architecture. It distills from *DinoV2* [58], *MASt3R* [46] and *MHMR* [5] and beats *MASt3R* on its own task, map-free localization, which we argue is particularly relevant for robotics tasks.

ResNet from scratch [31] is a non-pretrained baseline, a ResNet-18 initialized from scratch and entirely trained with the downstream RL loss.

Lineup of projection heads. Apart from the ResNet from scratch, all vision encoders have been frozen during RL training. They produce a feature tensor $\mathbf{F}_t \in \mathbb{R}^{P \times E}$ where P is the number of patches and E the embedding dimension. When useful we will denote individual patch features for patch p by $\mathbf{F}_{t,p}$. We project / pool the features with a trainable layer into a lower dimensional vector $\mathbf{f}_t$ and explore several projection mechanisms (also illustrated in Fig. 3):

Cross-attention pooling (CAP) is a widely used standard features extraction method in computer vision, a.k.a “*Perceiver-Resampler*” [2,37] in the context of VLMs. It employs a set of K learned query embeddings $\mathbf{q}_k$ to cross-attend into the feature tokens with multi-head attention, $\mathbf{f}_t = \{\text{MHA}(\mathbf{q}_k, \mathbf{F}_t)\}_{k=1..K}$.

Pure Attention Projection: (PAP) we introduce a variant of attention pooling which entirely discards the value projections and where the attention distribution has been “*elevated*” from an internal means of computation to the actual representation passed to the downstream agent. For a single head with a query projection matrix $\mathbf{W}_{\text{query}}$ and key projection matrix $\mathbf{W}_{\text{key}}$, this

<i>Trained from scratch (500M steps)</i>						
ViT Backbone	Sim (HM3D/2.5k)			Real (TestBuilding/14)		
	SCT	SPL	SR	SCT	SPL	SR
DinoV2	16.1	49.8	63.4	11.8 $\pm$ 1.3	37.2 $\pm$ 3.9	64.3 $\pm$ 5.8
VC-1	28.9	72.2	86.3	24.3 $\pm$ 1.9	63.4 $\pm$ 4.8	95.2 $\pm$ 6.7
Dune	33.4	76.9	90.7	26.8 $\pm$ 2.0	62.5 $\pm$ 3.4	95.3 $\pm$ 3.3
AM-Radio	27.6	68.2	85.2	29.3 $\pm$ 0.7	63.4 $\pm$ 0.5	100 $\pm$ 0.0
DinoV3	0.0 $\rightarrow$ see Table 4			not evaluated		
*ResNet	15.0	41.0	53.5	not evaluated		

*Half-width ResNet trained from scratch

Table 1: Comparison of vision encoders – kept frozen while the remaining agent is RL-trained from scratch for 500M steps. Projection heads are Cross-attention pooling (CAP). Real experiments: 3 runs $\times$ 14 episodes per model. We report “avg $\pm$ std” (168 real nav episodes in this table).

can be written as $\mathbf{f}_t = \{\sigma(\mathbf{q}_k \mathbf{W}_{\text{query}} \cdot (\mathbf{F}_t \mathbf{W}_{\text{key}})^\top)\}_{k=1\dots K}$, where σ is the softmax function. As we will see in the experiments further below, this will keep the downstream performance of full attention pooling while at the time lead to arguably quite interpretable vision features.

Principal Component Analysis (PCA) projects each patch on C principal components, with no learnable parameter apart from the final linear layer.

Linear Patchwise Reduction (LPR) projects each patch feature vector in $\mathbf{F}_t$ down to a low dimensional vector of size D , $\mathbf{f}_{t,p} = \mathbf{W} \mathbf{F}_{t,p}$ where $\mathbf{W}$ is a weight matrix.

How to train your Encoder. Table 1 shows navigation results in **simulation** with realistic motion (left) and in the **real environment** (right) and for the lineup of different vision encoders. The ViTs were kept frozen, but the rest of the agent (i.e. ViT projection head, modality encoders, recurrent memory and policy) were trained from scratch for 500M steps. We can see that the vision encoder baseline trained from scratch with RL did not perform at all (SR=53.5%), which we link to the combination of the weak learning signal provided by RL and the difficulty of training an agent with realistic motion in the augmented simulator³.

Using pre-trained ViTs with simple Cross-attention pooling (CAP) leads to quite high performance, and we can see that the performance metrics are comparable between the 2,500 episodes evaluated in simulation and the 3×14 episodes evaluated with the real robot in the real environment. This suggests our main finding: we argue that **the usage of pre-trained vision encoders combined with realistic motion modelling [11, 38] has closed the sim2real gap to neglectable size**. Navigation of fast-moving agents can indeed be trained purely in simulation without any real data involved.

The choice of vision encoder does impact performance and generalist ViTs such as DinoV2 and DinoV3 under-perform. We will investigate the failure of DinoV3 at the end of the paper. The two encoders using distillation from heteroge-

³ Training an agent with classical “teleportation” actions, eg. the action space = {move forward 0.25m, turn left 10°, turn right 10°, stop}, leads to relatively high training success rates but also a high sim2real gap and low transferability.

<i>Finetune from an agent receiving a Lidar-like input (+100M steps)</i>						
ViT Backbone	Sim			Real		
	SCT	SPL	SR	SCT	SPL	SR
DinoV2	31.6	74.3	88.0	27.5 $\pm$ 4.2	57.2 $\pm$ 9.4	97.6 $\pm$ 3.3
VC-1	32.4	76.1	90.3	28.3 $\pm$ 2.4	65.0 $\pm$ 4.4	100 $\pm$ 0.0
Dune	34.1	77.5	89.8	30.1 $\pm$ 1.6	64.1 $\pm$ 1.7	100 $\pm$ 0.0
AM-Radio	30.4	72.0	87.4	25.6 $\pm$ 0.7	56.0 $\pm$ 1.5	92.9 $\pm$ 5.8
DinoV3	0.0 $\rightarrow$ see table 4			not evaluated		
*ResNet	14.1	38.0	48.4	not evaluated		
Lidar (n.c.)	38.1	81.9	94.6	36.5 $\pm$ 0.7	76.5 $\pm$ 2.7	100 $\pm$ 0.0

*Half-width ResNet trained from scratch

Table 2: Comparison of vision encoders – where the agent is initialized from a pre-trained model solving the same navigation task, yet using a 2D 360° Lidar-like scan instead of the RGB input. Models are fine-tuned on RGB input for 100M steps (210 real nav episodes in this table). For reference, we also report the metrics for the Lidar-like agent we finetuned from, despite not being comparable.

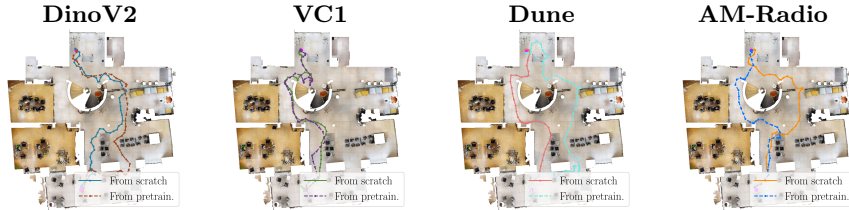


Fig. 4: Comparison of real trajectories – between agents trained from scratch or finetuned from a pretrained baseline. The agent is tasked to navigate from $\bullet$ to $\blacksquare$.

neous teachers perform well, in particular in the real experiments. We conjecture that mixing multiple tasks can overcome shortcomings of pure self-supervised learning. The performance of VC-1 was surprising to us, as the MAE loss is not known to outperform self-distillation. We suspect that the good performance might stem from the data mix of 4.3M images specially targeting robotics.

Pre-train with privileged 3D information. Table 2 shows experiments with agents using the same vision encoders, but this time the rest of the agent is not trained from scratch. Instead, we explore the availability of privileged information available during in simulation. We first train an agent which gets its RGB input replaced by a 2D 360° Lidar-like scan for 850M steps. This input signal is more directly related to the 3D scene structure and allows for easy exploitation by the agent with the RL learning signal. In a second phase, we finetune this agent for 100M steps on vision only input and Cross-attention pooling (CAP).

We observe a boost in metrics by most agents with the exception of the one using the AM-Radio encoder, which translates into better behavior during in the real-world (figure 4). Information from the privileged agent seems to transfer to the vision-only agents although the actual vision encoders were kept frozen. This experiment also substantially decreased the differences between the vision encoders, which provides evidence that the content of their features is useful for

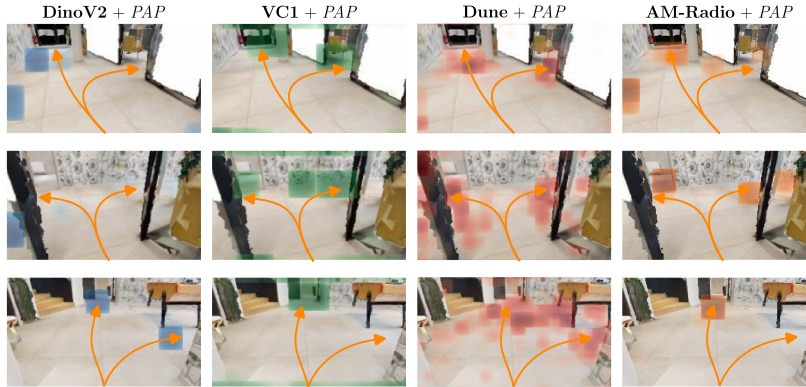


Fig. 5: Affordances emerge with Pure Attention Projection, ie. interpretable maps which are the **only information on the scene provided to the agent**. The shown map corresponds to one of the read-out tokens (2^{nd} column in Fig. 7), which we associate with potential directions accessible to the robot (manually labeled with $\nearrow$.)

robotics in a similar degree. However, there might be differences in the difficulty of decoding the necessary information by an agent trained with weak RL losses.

Interpretable spatial maps emerge from pre-trained encoders. All previously discussed experiments have been performed with Cross-attention pooling (CAP). We now introduce a new pooling, “*Pure Attention Projection (PAP)*” described at the beginning of this section. It is derived from Cross-attention pooling, completely discards the value projections and has the peculiar (and as we argue, very useful) property that the entirety of the features passed to the downstream agent corresponds to interpretable heatmaps. In other words, compared to other methods for interpretability in machine learning, there is no additional hidden information passed to the model, which a human would not see when inspecting interpretable maps. This is achieved while maintaining downstream navigation performance (see Table 3, commented in detail further below).

Similar to CAP, PAP uses a small set of trained query tokens $\mathbf{q}_n$ and produces an attention map per token. Figure 5 shows these maps overlayed over a real input image for 4 different pre-trained ViTs and for a single selected read-out token out of 4 tokens. We have observed that particular semantic meanings emerge in these maps, and found particularly exciting the emergence of what we call “*affordance map*” in this context: attention seems to focus on the areas in the scene, where an agent could potentially go when the current position / area needs to be left. Note, that this is an “unconditional” perception module, it is not conditioned on the agent’s goal. We also found it very interesting that this behavior emerges in projection layers for every pre-trained ViT we evaluated. We would like to stress again, that the heatmaps shown in Fig. 5 (one map for each of the 4 read-out tokens) are **the only information available to the agent**: there are no additional embeddings or features, except odometry input, which does not provide information on the scene.

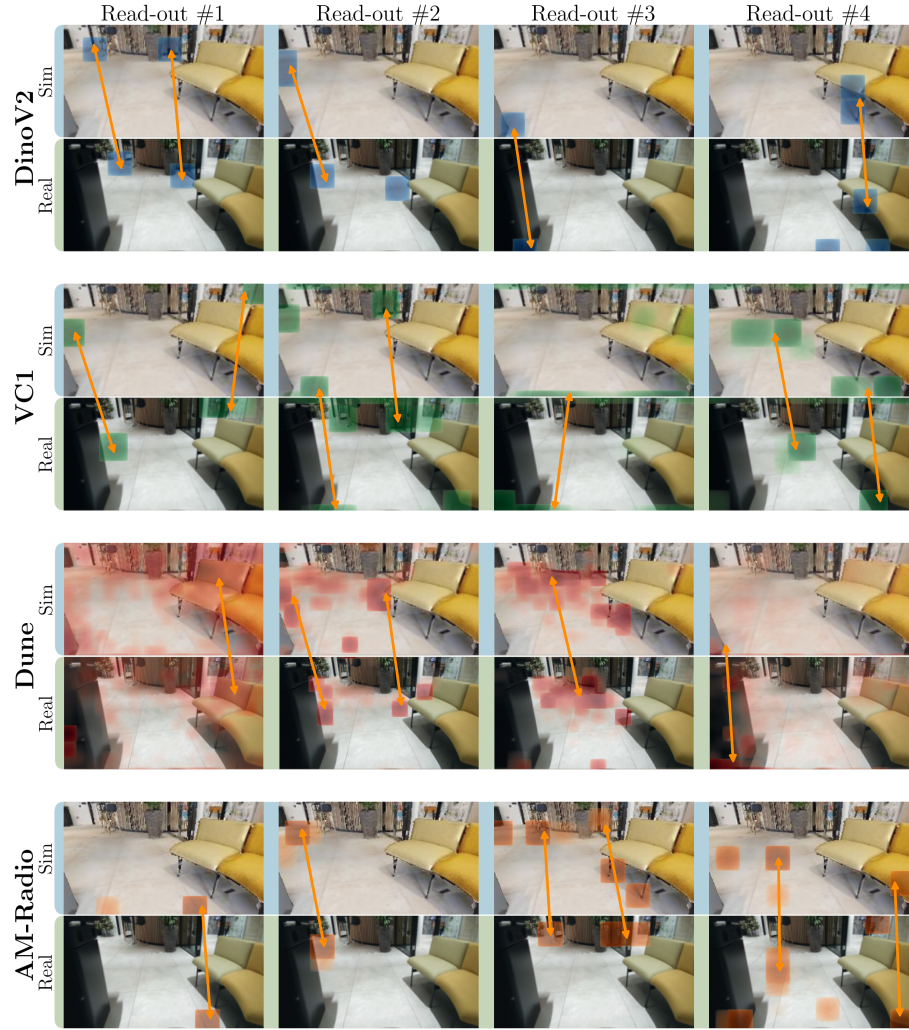


Fig. 6: Low sim2real gap visualized with Pure Attention Projection for different visual encoders. Each column shows the output given by one read-out token. We aligned viewpoints between real images captured by the robot and rendered views from the simulator, and compare the attention distribution. All ViTs seem to attend to similar locations despite difference in lighting condition and texture. Corresponding regions of attention are manually labeled with $\nearrow$.

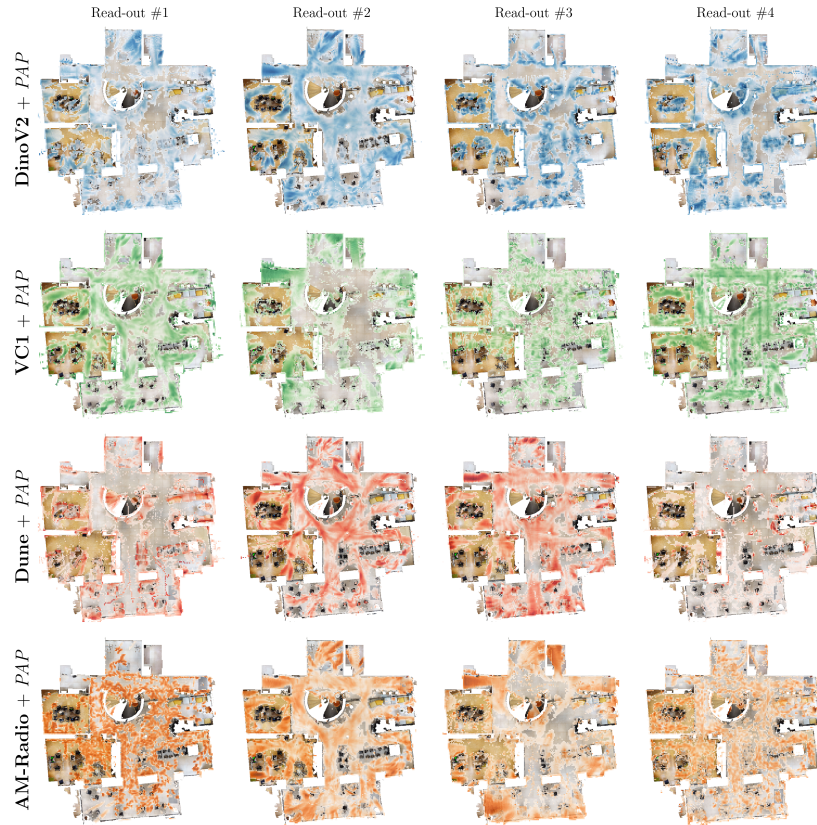


Fig. 7: Attention maps – with pure attention projection (PAP) accumulated and superimposed over a map. (data collected in simulation).

Fig. 6 shows that these maps are robust wrt. to the sim2real gap. The attention on a simulated region in the scene is similar to the attention on a similar input from the real robot’s camera. For convenience we overlaid arrows indicating the same regions in our test building.

In Fig. 7 we investigate how these heatmaps are linked to scene structure and semantics. We un-project and spatially integrate them into a global map, which we overlay over a map in simulation. We observed that they correspond to emerged concept detectors linked to navigable space, obstacles and affordances. **How much visual information is needed?** We have seen that passing simple attention maps to the downstream agent leads to interpretable features, and on the spider plots in Figure 8 we see that this does not sacrifice downstream navigation performance. We evaluate the different pooling methods introduced in the beginning of this section, CAP, PAP, PCA and LPR, using DinoV2 and AM-Radio vision encoders for evaluation in **simulation** and **real** (336 real navigation episodes have been performed in our test building for this figure alone). PCA does surprisingly underperform, we observe similar or better performances in

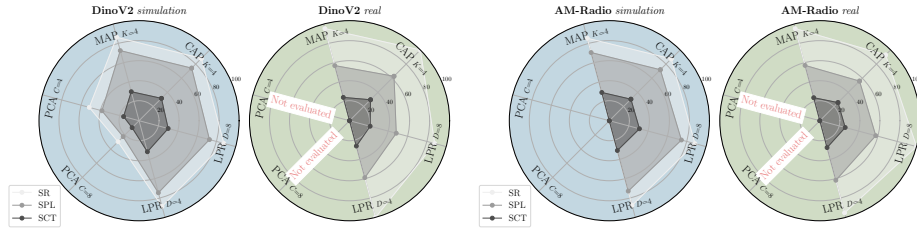


Fig. 8: Comparing different projection mechanisms – applied to DinoV2 and AM-Radio encoders. Models are initialized from privileged agent and fine-tuned for 100M steps. We observe similar or better performances in sim and real with projectors that bottleneck with low complexity (cf. Sec. B.2). See Table 6 in the supp.mat. for numbers and variations, ie. avg $\pm$ std-dev (336 real nav episodes in this Figure).

ViT Backbone	Projector	Sim			Real		
		SCT	SPL	SR	SCT	SPL	SR
DinoV2 + CAP	$K=1$	31.3	74.8	88.0	28.8 $\pm$ 0.6	61.7 $\pm$ 1.5	97.6 $\pm$ 3.3
	$K=2$	32.0	75.2	89.7	23.9 $\pm$ 1.4	54.6 $\pm$ 2.1	95.2 $\pm$ 6.7
	$K=4$	31.6	74.3	88.0	27.5 $\pm$ 4.2	57.2 $\pm$ 9.4	97.6 $\pm$ 3.3
	$K=8$	30.8	73.2	88.6	22.6 $\pm$ 2.0	53.1 $\pm$ 4.5	90.5 $\pm$ 6.7
DinoV2 + PAP	$K=1$	25.3	66.5	81.7	20.4 $\pm$ 0.3	54.9 $\pm$ 1.4	92.9 $\pm$ 0.0
	$K=2$	29.2	72.2	86.8	23.2 $\pm$ 2.4	55.8 $\pm$ 2.2	100 $\pm$ 0.0
	$K=4$	30.0	72.5	86.3	24.1 $\pm$ 1.9	57.0 $\pm$ 3.5	95.3 $\pm$ 3.3
	$K=8$	30.5	75.1	89.3	23.5 $\pm$ 1.3	53.9 $\pm$ 0.7	92.9 $\pm$ 0.0

Table 3: Sensitivity analysis of number of read-out tokens – for CAP and PAP projectors. While performance in simulation tends to increase with the number of read-out tokens, this does not transfer well on real experiments. Agent finetuned from Lidar (comparable to Table 2). (336 real nav episodes in this table).

simulation and real with projectors that conserve spatial information (full data including variations w. avg $\pm$ std-dev is given in Table 6 in the supp.mat.).

This raises the question how much information from each patch embedding delivered by the ViTs is actually useful and *required* for downstream agents. Table 3 provides a sensitivity analysis, where we vary the number K of read-out tokens in the set $\{1, 2, 4, 8\}$ for both CAP and PAP projectors. For CAP, given K read-out tokens, this leads to $K \cdot E$ values, where E is the embedding dimension of the ViT. For PAP, this leads to $K \cdot P$ values, where P is the amount of patches in the input image, each heatmap being composed of P values. **We observe that excellent navigation performance even in real experiments can be achieved with as low as 1 scalar per image patch passed to the downstream agent.** While performance in simulation tends to increase with the number of read-out tokens, this does not transfer well to real experiments, corroborating that bottlenecking has a positive impact on the sim2real gap.

The special case of DinoV3. The extremely bad performance of DinoV3 (cf. Tables 1 and 2) was surprising given the gains it achieved over its predecessor DinoV2 over a set of standard CV tasks. We particularly investigated this to exclude any potential mistakes in the setup. Table 4 shows that the classical CAP

Setup	Sim			Real		
	SCT	SPL	SR	SCT	SPL	SR
CAP $K=4$, Seed #1	00.0	00.0	00.0	<i>not evaluated</i>		
CAP $K=4$, Seed #2	00.0	00.0	00.0			
CAP $K=4$, Seed #3	00.0	00.0	00.0			
CAP $K=2$	00.0	00.0	00.0	<i>not evaluated</i>		
CAP $K=8$	00.0	00.0	00.0			
CAP $K=4$, ViT/s	00.0	00.0	00.0	<i>not evaluated</i>		
PAP $K=2$	12.8	41.6	52.1	<i>not evaluated</i>		
PAP $K=4$	16.1	47.0	59.8	<i>not evaluated</i>		
PAP $K=8$	24.9	63.7	79.7	24.3 ± 0.2	58.1 ± 1.8	95.3 ± 3.3
LPR $D=4$	21.7	59.6	77.7	21.0 ± 2.1	49.6 ± 2.6	88.1 ± 3.4
LPR $D=8$	10.2	23.6	31.6	<i>not evaluated</i>		

Table 4: DinoV3 required extra efforts compared to the other ViTs. We tested different configurations and seeds for CAP, but failed systematically. However, less complex, spatial preserving projectors like PAP and LPR manage to exploit DinoV3 features while transferring well to the real world. (84 real nav episodes in this table).

projector completely failed for this ViT even when exploring different numbers of read-out tokens, changing the size from ViT/B to ViT/S, and of course training with different seeds. However, it was interesting that we could make DinoV3 work with the new interpretable PAP projector. We conjecture that the complexity is at stake, and CAP fails to identify required features from DinoV3 from the weak reward signal. On the other hand, PAP and LRP –while less expressive – are also less complex and preserve spatiality of the representation.

These findings are somewhat corroborated by other experimental studies. As found in [34], DinoV3 relies on globally coherent low-frequency structures, tends to ignore high-frequency components, and “*DinoV3’s decision boundary critically depends on maintaining global spatial organization rather than on localized details*”. Similarly, a study for medical image analysis [48] showed that DinoV3’s natural image features do not transfer to the fine-grained textural analysis.

5 Conclusion

We presented a large-scale study of visual pre-training strategies for real-world navigation and conducted 966 episodes with a real robot in a real building. We show that while learning perception directly from actions alone fails to generalize, leveraging state-of-the-art encoders, and particularly those distilled from heterogeneous teachers like AM-Radio [32] and Dune [67], enables robust sim-to-real transfer when combined with realistic motion modelling. We introduced a new projection layer called “*Pure Attention Projection*” which reveals that navigational information can be extracted from frozen ViTs and compressed into highly efficient, interpretable bottleneck representations that naturally encode affordances without loss of performance. Our results suggest that real-world navigation can be achieved by training agents entirely in simulation if perception is delegated to rich vision encoders combined with proper projections and that robot motion is simulated realistically during training.

References

- [1] Ai, B., Wu, Z., Hsu, D.: Invariance is key to generalization: Examining the role of representation in sim-to-real transfer for visual navigation. In: International Symposium on Experimental Robotics (2023)
- [2] Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. In: Neural Information Processing Systems (2022)
- [3] Anderson, P., Chang, A.X., Chaplot, D.S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., Zamir, A.R.: On evaluation of embodied navigation agents. arXiv preprint (2018)
- [4] Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: Conference on Computer Vision and Pattern Recognition (2016)
- [5] Baradel, F., Armando, M., Galaaoui, S., Brégier, R., Weinzaepfel, P., Rogez, G., Lucas, T.: Multi-hmr: Multi-person whole-body human mesh recovery in a single shot. In: European Conference on Computer Vision (2024)
- [6] Beeching, E., Dibangoye, J., Simonin, O., Wolf, C.: Learning to reason on uncertain topological maps. In: European Conference on Computer Vision (2020)
- [7] Beeching, E., Dibangoye, J., Simonin, O., Wolf, C.: Egomap: Projective mapping and structured egocentric memory for deep RL. In: European Conference on Machine Learning (2020)
- [8] Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Shi, L.X., Tanner, J., Vuong, Q., Walling, A., Wang, H., Zhilinsky, U.: π_0 : A vision-language-action flow model for general robot control. In: arxiv:2410.24164 (2024)
- [9] Bono, G., Antsfeld, L., Chidlovskii, B., Weinzaepfel, P., Wolf, C.: End-to-End (Instance)-Image Goal Navigation through Correspondence as an Emergent Phenomenon,. In: International Conference on Learning Representation (2024)
- [10] Bono, G., Antsfeld, L., Sadek, A., Monaci, G., Wolf, C.: Learning with a Mole: Transferable latent spatial representations for navigation without reconstruction. In: International Conference on Learning Representation (2024)
- [11] Bono, G., Poirier, H., Antsfeld, L., Monaci, G., Chidlovskii, B., Wolf, C.: Learning to navigate efficiently and precisely in real environments. In: Conference on Computer Vision and Pattern Recognition (2024)
- [12] Bresson, G., Alsayed, Z., Yu, L., Glaser, S.: Simultaneous localization and mapping: A survey of current trends in autonomous driving. IEEE Transactions on Intelligent Vehicles (2017)

- [13] Burgard, W., Cremers, A.B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S.: The interactive museum tour-guide robot. In: Association for the Advancement of Artificial Intelligence (1998)
- [14] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision (2020)
- [15] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: International Conference on Computer Vision (2021)
- [16] Chaplot, D.S., Gandhi, D., Gupta, A., Salakhutdinov, R.: Object goal navigation using goal-oriented semantic exploration. In: Neural Information Processing Systems (2020)
- [17] Chaplot, D.S., Gandhi, D., Gupta, S., Gupta, A., Salakhutdinov, R.: Learning to explore using active neural slam. In: International Conference on Learning Representation (2020)
- [18] Chaplot, D.S., Salakhutdinov, R., Gupta, A., Gupta, S.: Neural topological slam for visual navigation. In: Conference on Computer Vision and Pattern Recognition (2020)
- [19] Chattopadhyay, P., Hoffman, J., Mottaghi, R., Kembhavi, A.: Robustnav: Towards benchmarking robustness in embodied navigation. CoRR (2021)
- [20] Chen, S., Guhur, P.L., Tapaswi, M., Schmid, C., Laptev, I.: Think Global, Act Local: Dual-scale Graph Transformer for Vision-and-Language Navigation. In: Conference on Computer Vision and Pattern Recognition (2022)
- [21] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Conference on Empirical Methods in Natural Language Processing (2014)
- [22] Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning. In: Neural Information Processing Systems (2023)
- [23] Ding, Y., Florensa, C., Abbeel, P., Phielipp, M.: Goal-conditioned imitation learning. In: Neural Information Processing Systems (2019)
- [24] Du, H., Yu, X., Zheng, L.: Vtnet: Visual transformer network for object goal navigation. In: International Conference on Learning Representation (2021)
- [25] Fang, A., Jose, A.M., Jain, A., Schmidt, L., Toshev, A.T., Shankar, V.: Data filtering networks. In: International Conference on Learning Representation (2024)
- [26] Fang, K., Toshev, A., Fei-Fei, L., Savarese, S.: Scene memory transformer for embodied agents in long-horizon tasks. In: Conference on Computer Vision and Pattern Recognition (2019)
- [27] Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine (1997)
- [28] Garg, S., Rana, K., Hosseinzadeh, M., Mares, L., Sünderhauf, N., Dayoub, F., Reid, I.: Robohop: Segment-based topological map representation for open-world visual navigation. In: International Conference on Robotics and Automation (2024)

- [29] Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization (2015)
- [30] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Conference on Computer Vision and Pattern Recognition (2022)
- [31] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Conference on Computer Vision and Pattern Recognition (2016)
- [32] Heinrich, G., Ranzinger, M., Hongxu, Yin, Lu, Y., Kautz, J., Tao, A., Catanzaro, B., Molchanov, P.: Radiov2.5: Improved baselines for agglomerative vision foundation models. In: Conference on Computer Vision and Pattern Recognition (2025)
- [33] Henriques, J.F., Vedaldi, A.: Mapnet: An allocentric spatial memory for mapping environments. In: Conference on Computer Vision and Pattern Recognition (2018)
- [34] Huang, Z., Li, J., Wen, H., Li, T., Yang, X., Qi, L., Peng, B., Huang, X., Yang, M.H., Cheng, G.: Rethinking cross-generator image forgery detection through dinov3. In: arxiv:2511.22471 (2025)
- [35] Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Galliker, M.Y., Ghosh, D., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., LeBlanc, D., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Ren, A.Z., Shi, L.X., Smith, L., Springenberg, J.T., Stachowicz, K., Tanner, J., Vuong, Q., Walke, H., Walling, A., Wang, H., Yu, L., Zhilinsky, U.: $\pi_{0.5}$: a vision-language-action model with open-world generalization. In: arxiv:2504.16054 (2025)
- [36] Jaderberg, M., Mnih, V., Czarnecki, W.M., Schaul, T., Leibo, J.Z., Silver, D., Kavukcuoglu, K.: Reinforcement learning with unsupervised auxiliary tasks. In: International Conference on Learning Representation (2017)
- [37] Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., Carreira, J.: Perceiver: General perception with iterative attention. In: International Conference on Machine Learning (2021)
- [38] Janny, S., Poirier, H., Antsfeld, L., Bono, G., Monaci, G., Chidlovskii, B., Giuliani, F., Bue, A.D., Wolf, C.: Reasoning in visual navigation of end-to-end trained agents: a dynamical systems approach. In: Conference on Computer Vision and Pattern Recognition (2025)
- [39] Janny, S., Poirier, H., Antsfeld, L., Bono, G., Monaci, G., Chidlovskii, B., Giuliani, F., Bue, A.D., Wolf, C.: Reasoning in visual navigation of end-to-end trained agents: a dynamical systems approach. In: Conference on Computer Vision and Pattern Recognition (2025)
- [40] Karypidis, E., Kakogeorgiou, I., Gidaris, S., Komodakis, N.: Dino-foresight: Looking into the future with dino. In: Neural Information Processing Systems (2026)
- [41] Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: Openvla: An

- open-source vision-language-action model. In: Conference on Robot Learning (2024)
- [42] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollar, P., Girshick, R.: Segment anything. In: International Conference on Computer Vision (2023)
 - [43] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. In: International Conference on Computer Vision (2023)
 - [44] Konolige, K.: A gradient method for realtime robot control. In: International Conference on Intelligent Robots and Systems (2000)
 - [45] Labbé, M., Michaud, F.: RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics* (2019)
 - [46] Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3D with MAST3R. In: European Conference on Computer Vision (2024)
 - [47] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Conference on Computer Vision and Pattern Recognition (2017)
 - [48] Liu, C., Chen, Y., Shi, H., Lu, J., Jian, B., Pan, J., Cai, L., Wang, J., Yu, J., Gao, Z., Zhang, X., Bai, L., Zhang, Y., Li, J., Bercea, C.I., Ouyang, C., Chen, C., Xiong, Z., Wiestler, B., Wachinger, C., Duncan, J.S., Rueckert, D., Bai, W., Arcucci, R.: Does dinov3 set a new medical vision standard? benchmarking 2d and 3d classification, segmentation, and registration. In: arxiv:2509.06467 (2026)
 - [49] Liu, X., Li, J., Jiang, Y., Sujay, N., Yang, Z., Zhang, J., Abanes, J., Zhang, J., Feng, C.: Citywalker: Learning embodied urban navigation from web-scale videos. In: Conference on Computer Vision and Pattern Recognition (2025)
 - [50] Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., Kipf, T.: Object-centric learning with slot attention. In: Neural Information Processing Systems (2020)
 - [51] Macenski, S., Martín, F., White, R., Clavero, J.G.: The marathon 2: A navigation system. In: International Conference on Intelligent Robots and Systems (2020)
 - [52] Majumdar, A., Yadav, K., Arnaud, S., Ma, Y.J., Chen, C., Silwal, S., Jain, A., Berges, V.P., Abbeel, P., Malik, J., Batra, D., Lin, Y., Maksymets, O., Rajeswaran, A., Meier, F.: Where are we in the search for an artificial visual cortex for embodied intelligence? In: Neural Information Processing Systems (2023)
 - [53] Marder-Eppstein, E., Berger, E., Foote, T., Gerkey, B., Konolige, K.: The office marathon: Robust navigation in an indoor office environment. In: International Conference on Robotics and Automation (2010)
 - [54] Marza, P., Matignon, L., Simonin, O., Wolf, C.: Multi-Object Navigation with dynamically learned neural implicit representations. In: International Conference on Computer Vision (2023)

- [55] Mirowski, P., Pascanu, R., Viola, F., Soyer, H., Ballard, A., Banino, A., Denil, M., Goroshin, R., Sifre, L., Kavukcuoglu, K., Kumaran, D., Hadsell, R.: Learning to navigate in complex environments. In: International Conference on Learning Representation (2017)
- [56] Monaci, G., Antsfeld, L., Chidlovskii, B., Wolf, C.: Zero-bev: Zero-shot projection of any first-person modality to bev maps. In: International Conference on 3D Vision (3DV) (2024)
- [57] Monaci, G., Weinzaepfel, P., Wolf, C.: What does really matter in image goal navigation? In: International Conference on 3D Vision (2026)
- [58] Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Balas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: DINOv2: Learning robust visual features without supervision. *Transaction on Machine Learning Research* (2024)
- [59] Parisotto, E., Salakhutdinov, R.: Neural map: Structured memory for deep reinforcement learning. In: International Conference on Learning Representation (2018)
- [60] Qian, S., Mo, K., Blukis, V., Fouhey, D.F., Fox, D., Goyal, A.: 3d-mvp: 3d multiview pretraining for manipulation. In: Conference on Computer Vision and Pattern Recognition (2025)
- [61] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021)
- [62] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021)
- [63] Ramakrishnan, S.K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J.M., Undersander, E., Galuba, W., Westbury, A., Chang, A.X., Savva, M., Zhao, Y., Batra, D.: Habitat-matterport 3D dataset (HM3D): 1000 large-scale 3d environments for embodied AI. In: Neural Information Processing Systems (2021)
- [64] Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S.G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J.T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., de Freitas, N.: A Generalist Agent. *Transaction on Machine Learning Research* (2022)
- [65] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Neural Information Processing Systems (2016)
- [66] Rösmann, C., Hoffmann, F., Bertram, T.: Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control. In: European Control Conference (ECC) (2015)

- [67] Sariyildiz, M.B., Weinzaepfel, P., Lucas, T., de Jorge, P., Larlus, D., Kalantidis, Y.: Dune: Distilling a universal encoder from heterogeneous 2d and 3d teachers. In: Conference on Computer Vision and Pattern Recognition (2025)
- [68] Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., Parikh, D., Batra, D.: Habitat: A platform for embodied ai research. In: International Conference on Computer Vision (2019)
- [69] Sax, A., Zhang, J.O., Emi, B., Zamir, A., Savarese, S., Guibas, L., Malik, J.: Learning to navigate using mid-level visual priors. In: Conference on Robot Learning. PMLR (2020)
- [70] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint (2017)
- [71] Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences* (1996)
- [72] Shah, D., Levine, S.: ViKiNG: Vision-based kilometer-scale navigation with geographic hints. In: Robotics, Science and System (2022)
- [73] Shah, D., Sridhar, A., Dashora, N., Stachowicz, K., Black, K., Hirose, N., Levine, S.: ViNT: A foundation model for visual navigation. In: Conference on Robot Learning (2023)
- [74] Shukor, M., Aubakirova, D., Capuano, F., Kooijmans, P., Palma, S., Zouitine, A., Aractingi, M., Pascal, C., Russi, M., Marafioti, A., Alibert, S., Cord, M., Wolf, T., Cadene, R.: Smolvla: A vision-language-action model for affordable and efficient robotics. In: arxiv:2506.01844 (2025)
- [75] Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: Dinov3. In: arxiv:2508.10104 (2025)
- [76] Sun, X., Chen, P., Fan, J., Chen, J., Li, T., Tan, M.: Fgprompt: fine-grained goal prompting for image-goal navigation. In: Neural Information Processing Systems (2024)
- [77] Team, O.M., Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., Luo, J., Tan, Y.L., Chen, L.Y., Sanketi, P., Vuong, Q., Xiao, T., Sadigh, D., Finn, C., Levine, S.: Octo: An open-source generalist robot policy. In: Robotics, Science and System (2024)
- [78] Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics (2005)
- [79] Uppal, S., Agarwal, A., Xiong, H., Shaw, K., Pathak, D.: Spin: Simultaneous perception interaction and navigation. In: Conference on Computer Vision and Pattern Recognition (2024)
- [80] Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vgggt: Visual geometry grounded transformer. In: Conference on Computer Vision and Pattern Recognition (2025)
- [81] Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUST3R: Geometric 3D vision made easy. In: Conference on Computer Vision and Pattern Recognition (2024)

- [82] Wang, Y., Zhang, M., Li, Z., Driggs-Campbell, K.R., Wu, J., Fei-Fei, L., Li, Y.: D³fields: Dynamic 3d descriptor fields for zero-shot generalizable robotic manipulation. In: International Conference on Robotics and Automation (2024)
- [83] Yokoyama, N., Ha, S., Batra, D.: Success weighted by completion time: A dynamics-aware evaluation criteria for embodied navigation. In: International Conference on Intelligent Robots and Systems (2021)
- [84] Zeng, K.H., Zhang, Z., Ehsani, K., Hendrix, R., Salvador, J., Herrasti, A., Girshick, R., Kembhavi, A., Weihs, L.: Poliformer: Scaling on-policy rl with transformers results in masterful navigators. Conference on Robot Learning (2024)
- [85] Zhou, G., Pan, H., Lecun, Y., Pinto, L.: DINO-WM: World models on pre-trained visual features enable zero-shot planning. In: International Conference on Machine Learning (2025)
- [86] Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: ibot: Image bert pre-training with online tokenizer. In: International Conference on Learning Representation (2022)

Supplementary Material for Paper #1424

A scalar per patch from pre-trained ViTs enables fast moving navigation in the real world

A Detailed agent architecture and training

A.1 Agent architecture

Table 5 reports additional information about the ViTs used throughout the paper. The on-board camera captures 160×120 images at 3Hz, matching the sampling frequency of the agent. The images are zero-padded and up-scaled to match the resolution expected by the visual encoder. We extract dense features via the frozen ViT in half-precision mode during both training and inference. This guarantees real-time execution on the Jetson Orin running the model on the real platform. Unless explicitly stated in the main paper, we use the ViT-Base version of the pretrained encoders.

The projector is directly applied to the dense features (patch + registers embeddings) given by the visual encoder. Let $\mathbf{F}_p \in \mathbb{R}^E$ be this output, where $p \in \llbracket 0, P \rrbracket$ is the number of patch embeddings.

Cross Attention Pooling (CAP) perform cross multi-head attention (MHA with 4 heads) between the P embeddings from the ViT backbone and K learnable features $\mathbf{q}_k \in \mathbb{R}^E$. The MHA outputs K latent embedding of size E , which are concatenated and projected to a final dimension of 1024 with a linear layer.

Pure Attention Projection (PAP) directly returns the attention weights from the cross attention, and ignore value projection. Concretely, each pair of patch embedding $\mathbf{F}_p$ and learnable read-out token $\mathbf{q}_k$ gives a single scalar value corresponding to the cosine similarity between $\mathbf{F}_p$ and $\mathbf{q}_k$, scaled by *softmax*. We obtain K latent vectors of size P , which are concatenated and project to a final dimension of 1024 with a linear layer.

Principal Component Analysis (PCA) is frequently used to visualize dense embeddings from pre-trained ViT (by computing the top-3 principal component, converted to RGB channels). We built on this idea to see if static projection is sufficient for navigation (a negative result, see table 6). We initialize the top C principal components (PCs) from 1M images collected during navigation in simulation. During training, each patch is projected independently onto the PCs. The projection weights are stacked patch-wise and projected to a final dimension of 1024 with a linear layer.

Linear Patchwise Reduction (LPR) projects each patch embedding independently with a linear layer, reducing their dimension from E to D (typically $D=4$ or $D=8$ scalar values). The resulting vectors are stacked patch-wise and projected to a final dimension of 1024 with a linear layer.

ViT	# Params.	Pre-training	img. size
<i>From scratch</i>			
Dino-v2 [58]	86.23M	Self-distillation [15] + iBOT [86]	224×224
Dino-v3 [75]	85.67M	Self-distillation [15] + iBOT [86]	224×224
VC-1 [52]	85.80M	MAE [30] on egocentric images	224×224
<i>Distilled</i>			
DUNE [67]	85.97M	Dino-V2 [58], Mast3R [46], Multi-HMR [5]	336×336
AM-Radio v2.5 [32]	98.23M	CLIP (DFN [25] and OpenAI [62]), Dino-V2 [58], SAM [42]	144×256

Table 5: Pre-trained ViT models – compared throughout this paper. We used the *base* version of each visual encoders as it offers a good trade-off between performances, training speed and real-time inference time.

The agent maintains a latent memory modelled by a 2-layer gated recurrent unit. The latent memory size is 1024 and is updated from latent observations:

- **RGB** is given by the output of the chosen projector, of size 1024,
- **Odometry** sensor returns the 2D position and orientation of the robot with respect to the episode start, as well as linear and angular velocities. The raw sensor values are encoded by a 1-layer MLP up to a final size of 64,
- **Previous action** is mapped to a learnable embedding of size 32,
- **Static point goal** is encoded with a 1-layer MLP up to a final size of 64.

The policy is a linear layer applied on the latent memory of the agent. The pretrained agent on privileged observations (used in table 2 does not have an RGB encoder, but instead encodes the 2D Lidar observation with a 1D CNN.

A.2 Training and evaluation setup

As mentioned in the main paper, each agent is trained with reinforcement learning, in particular PPO [70], and with a reward inspired by [19, 38], $r_t = R \cdot \mathbb{I}_{\text{success}} - \Delta_t^{\text{Geo}} - \lambda - C \cdot \mathbb{I}_{\text{collision}}$, where $R=2.5$, Δ_t^{Geo} is the gain in geodesic distance to the goal, a slack cost $\lambda=0.01$ encourages efficiency, and a cost $C = 0.1$ penalizes each collision without ending the episode. Models from scratch (in table 1) are trained for 500M steps on a single A100 GPU (~ 3 -4 weeks of training), and agent derived from privileged agent are finetuned for 100M steps. The privileged agent has been trained for 800M steps, following [39].

We evaluate each models in both simulated and real environment. For security reasons and to prevent dangerous situation, we excluded from real world experiment any model that performs below 60% in success rate in simulation. We repeated each sequence of real episodes $\times 3$ to account for uncertainty and changing environmental condition, such as lighting. An episode is considered successful if the agent stops within 0.2m of the goal in simulation (episode is terminated at first stop), and within 1 m in real. Additionally, we found beneficial to perform an agent reset when approaching 2 m of the goal, based on local

ViT Back.	Projector	Sim			Real		
		SCT	SPL	SR	SCT	SPL	SR
DinoV2	CAP	31.6	74.3	88.0	27.5 $\pm$ 4.2	57.2 $\pm$ 9.4	97.6 $\pm$ 3.3
	PAP	30.0	72.5	86.3	24.1 $\pm$ 1.9	57.0 $\pm$ 3.5	95.3 $\pm$ 3.3
	LPR $D=4$	32.1	74.5	88.7	26.2 $\pm$ 0.4	58.9 $\pm$ 0.4	100 $\pm$ 0.0
	LPR $D=8$	30.1	71.8	87.7	21.3 $\pm$ 2.9	48.3 $\pm$ 4.4	88.1 $\pm$ 6.7
	PCA $C=4$	16.0	38.4	51.5	<i>not evaluated</i>		
	PCA $C=8$	9.8	22.5	29.7	<i>not evaluated</i>		
AM-Radio	CAP	30.4	72.0	87.4	25.6 $\pm$ 0.7	56.0 $\pm$ 1.5	92.9 $\pm$ 5.8
	PAP	29.2	70.5	85.3	23.8 $\pm$ 1.3	57.2 $\pm$ 3.3	95.3 $\pm$ 3.3
	LPR $D=4$	30.8	72.8	87.6	27.1 $\pm$ 2.0	60.6 $\pm$ 3.9	95.3 $\pm$ 3.3
	LPR $D=8$	31.1	74.5	88.5	26.2 $\pm$ 2.8	58.0 $\pm$ 6.4	100 $\pm$ 0.0
	PCA $C=4$	0	0	0	<i>not evaluated</i>		
	PCA $C=8$	0.1	0.2	0.4	<i>not evaluated</i>		

Table 6: Ablation on ViT projection – Detailed metrics associated with figure 8. All variants are initialized from the pre-trained agent with privileged information and finetuned for 100M steps. In general, we found beneficial to preserve spatiality in the final representation.

odometry measurements. In that case, the internal state is reset and the point goal is fed again to the agent. Moreover, the robot integrates a safety mechanism that prevent collision with object and people based on obstacle detection from an on-board 2D Lidar. While this sensory input is *not* provided to the agent, the safety mechanism is simulated during training.

B Additional experiments

Table 6 reports metrics displayed in figure 8 in the main paper. Our proposed alternatives for patch embedding projection performs on par or better than classical cross-attention projection. PAP and LPR methods are arguably less expressive than CAP, yet still enable efficient navigation in both simulated and real environments.

Remarkably, we observe very similar explainable feature maps with PAP and LPR, which indicates that the detected semantic classes are fundamental for navigation. We recall that these maps emerge solely from RL training and are not supervised directly. Figure 9 shows the magnitude of the intermediate representations from LPR, and exhibit segmentation-like properties for navigable floor, obstacles and affordances.

B.1 Cross-attention projection(CAP)

We also provided projected attention maps derived from CAP (using models from table 2, i.e. initialized from privileged agent). Conversely to PAP, the attention maps are not clearly interpretable. We argue that most of the information is

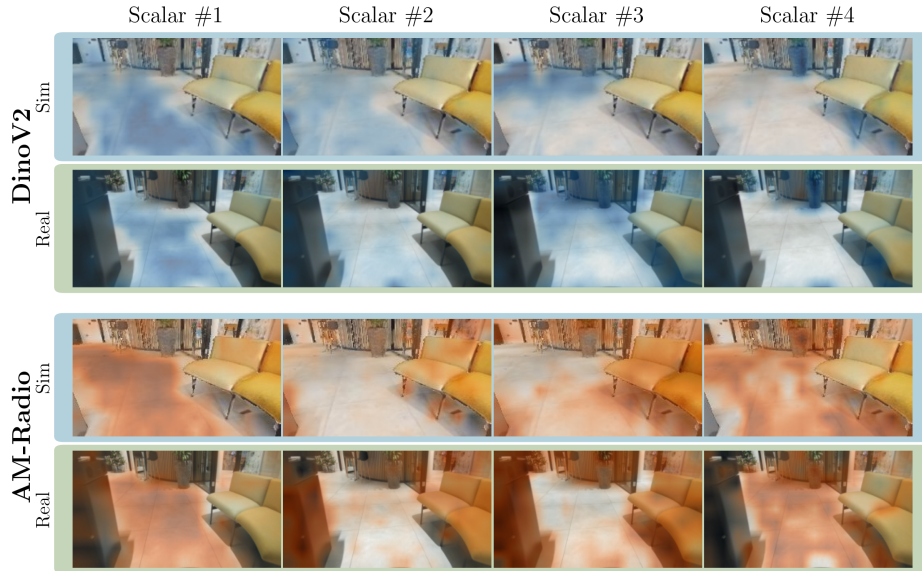


Fig.9: Magnitude from LPR – can be seen akin to attention maps from PAP, and exhibits very similar behavior. Sim2real gap is limited, as corresponding region in simulated and real images trigger similar magnitude level.

stored in the value projection, which can’t be visualized easily. Surprisingly, *DinoV2* with CAP exhibits a grid-like pattern which aligns with the floor tiling of our test environment.

Figure 11 show trajectory samples collected during real episodes with agents trained from scratch. Each figure shows the best realization for a given episode per model among the 3 repetitions. While the robot reaches the goal most of the time, the trajectory can be quite far from the optimal shortest path, which reflects performances measured in simulation. Qualitatively, *DinoV2* trajectories contain more spurious rotations which makes displacement erratic.

B.2 Investigating the impact of spatiality for projectors

The failure of *DinoV3* under several variants of projectors was a surprise, but we manage to obtain encouraging results with LPR and PAP, which led us to two potential explanations. We believe that extracting simple information for navigation from *DinoV3*’s complex features might be too difficult, especially from pure RL training. We hypothesize that the relative success of LPR and PAP might be explained by either (1) their ability to preserve spatial information in the representation, or (2) by the simplicity and weaker expressivity of the projector itself, which could pair better with reinforcement learning.

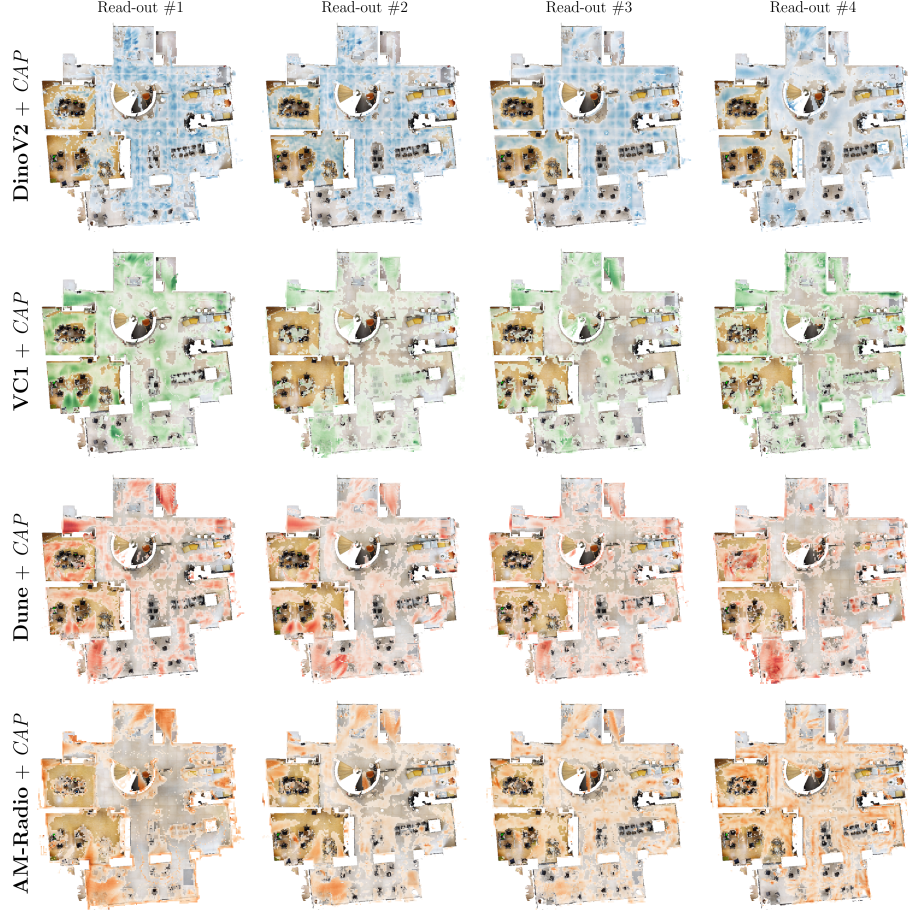


Fig. 10: Attention maps – with cross attention projection (CAP) accumulated and superimposed over a map. (data collected in simulation).

We conducted an additional experiment with a different projector preserving spatiality (similar to PAP and LPR), yet with a expressivity power comparable to CAP, ie. self-attention pooling (SAP) where the value projection matrix $\mathbf{W}_v$ projects the path embedding to a smaller dimension (4 scalars per patch):

$$\mathbf{f}_t = \{\sigma(\mathbf{F}_t \mathbf{W}_q \cdot (\mathbf{F}_t \mathbf{W}_k)^\top) \mathbf{W}_v \mathbf{F}_t$$

where $\mathbf{W}_v \in \mathbb{R}^{E \times 4}$. We report the metrics obtained with this variant in

ViT Backbone	Self-attention pooling		
	SCT	Sim SPL	SR
DinoV2	33.6	76.9	90.2
VC-1	31.8	75.1	89.0
Dune	32.1	73.7	87.6
AM-Radio	31.1	73.9	88.1
DinoV3	0.00	0.00	0.00

Fig. 12: Self-Attention Pooling – as an ablation on *DinoV3*. The failure of this projector indicates that complexity of the attention pooling is (in part) responsible for the failure of *DinoV3*.

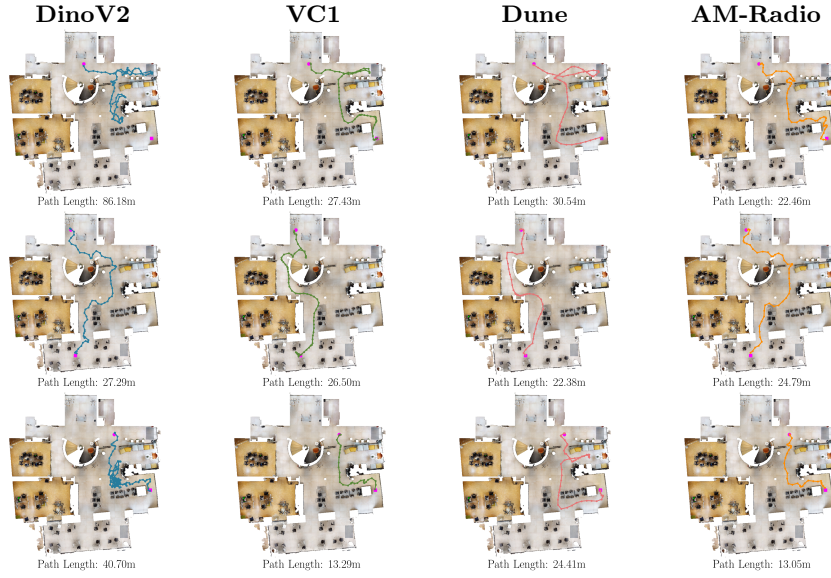


Fig. 11: Sampled trajectories from real experiments – using models trained from scratch (table 1). Overall, distilled models show better trajectories closer from the optimal path, while exhibiting better behavior with less hesitation and backtracking. For each model and episode, we pick the realization with the highest SPL. The agent is tasked to navigate from ● to ■.

table 12, which failed again using *DinoV3* as visual encoder. It indicates that compression and spatiality are not sufficient for enabling *DinoV3*. The complexity (i.e. expressivity) of the projector is also at stake, since more expressive projectors seem to struggle with this vision encoder.