Towards Stable Self-Supervised Object Representations in Unconstrained Egocentric Video

Supplementary Material

This appendix provides supplementary materials for the main paper. The contents are organized as follows:

- Detailed Method Description (Sec. A)
- Depth Decoder Architecture (Sec. B)
- Experimental Details (Sec. C)
- On the Reproducibility of the DoRA Baseline (Sec. D)
- Additional Experimental Results (Sec. E)
- More Visualization (Sec. F)
- Broader Impacts and Ethical Considerations (Sec. G)
- Detailed Comparison and Positioning Analysis with DINO and DoRA (Sec. H)
- Zero-Shot Generalization: Ego4D Case Study (Sec. I)

A. Detailed Method Description

We provide the notation used throughout the paper in Table 6, and a more detailed description of our method components in the following part.

A.1. Details of Class-Agnostic Proto-object Extraction

This section provides the formal vector-matrix formulations and dimensional specifications corresponding to the delineation process in Section 3.1.1. Let S denote the number of spatial tokens and D the feature dimension.

We articulate the process in three formal steps:

1. Prototype Synthesis. For the n-th attention head, we synthesize a prototype vector $\mathbf{o}_n^t \in \mathbb{R}^D$ by aggregating the query tokens $\mathbf{q}_n^t \in \mathbb{R}^{S\times D}$ weighted by the spatial attention map $\mathbf{A}_n^t \in \mathbb{R}^{1\times S}$. To facilitate matrix operations, we formulate this as:

$$\mathbf{o}_n^t = (\mathbf{A}_n^t \cdot \mathbf{q}_n^t)^\top \tag{5}$$

This operation collapses the spatial dimension S, distilling the head's attentional focus into a single global descriptor.

2. Soft Assignment via Normalized Similarity. We compute the alignment between the teacher's patch embeddings $\mathbf{e}^t \in \mathbb{R}^{S\times D}$ and the synthesized prototype. To robustly measure visual correspondence, we project the **normalized** patch embeddings onto the prototype vector:

$$\mathbf{M}_n^t = \frac{\mathbf{e}^t}{\|\mathbf{e}^t\|_2} \cdot \mathbf{o}_n^t \quad \in \mathbb{R}^S \tag{6}$$

This formulation effectively captures the angular alignment between patch features and the proto-object concept, matching the logic of cosine similarity while preserving the magnitude information of the prototype.

3. Mask Generation & Disentangled Feature Extraction. A binary mask is derived via parameter-free mean-thresholding: $\text{Mask}_n^t = \mathbb{1}(\mathbf{M}_n^t > \mathbb{E}[\mathbf{M}_n^t])$. To ensure maximal feature disentanglement, this mask is **upsampled** to the image resolution and applied to the raw input pixels X^t. Let $\text{Up}(\cdot)$ denote the nearest-neighbor interpolation operator mapping spatial tokens to pixel coordinates:

$$\mathbf{f}_n^t = g_\theta(X^t \odot \text{Up}(\text{Mask}_n^t)) \tag{7}$$

Design Rationale: Why Pixel-level Masking? A critical methodological choice in Eq. 7 is to apply the mask to the raw input X^t rather than intermediate feature maps. While feature-level masking is computationally cheaper, we prioritize **signal independence**. In any shared backbone, the features of a specific patch inevitably aggregate information from the background and other objects due to expanding receptive fields or global self-attention mixing. By forcing the student encoder to process the masked image from scratch, we physically block this "information leakage." Although computationally sub-optimal, this design provides cleaner possible testbed for validating our core hypothesis regarding compositional consistency.

Rationale for Soft Assignment It is instructive to contrast our patch assignment logic with methods like DoRA [62]. DoRA utilizes the Sinkhorn-Knopp algorithm to enforce a competitive, "winner-take-all" partition, meaning each patch must belong to exactly one slot. This structural rigidity can be limiting in cluttered egocentric scenes where occlusion and ambiguity are prevalent.

In contrast, our approach employs a similarity-based soft assignment followed by independent thresholding. We do not force prototypes to compete; a patch can be claimed by multiple heads or none at all. This flexibility is better suited for unconstrained "in-the-wild" video data, which necessitates a more permissive proto-object delineation strategy.

A.2. Implementation Details of Proto-object Representation Learning

This appendix provides the details for computing the compositional feature $\mathbf{f}_{\text{agg}}^t$, which is used in the second term $H(\mathbf{f}', \mathbf{f}_{\text{agg}})$ of the $\mathcal{L}_{\text{proto}}$ objective (Eq. 2).

Table 6. **Notation used throughout the paper.** All variables are indexed by frame time t. For shapes, S denotes the number of spatial patch tokens. D_1, D_2, D_3 are layer-specific feature dimensions for the backbone. D, D_q, D_k are the feature dimensions for patch embeddings and attention components.

Symbol	Meaning	Shape / Dim.	Source
X^t	RGB input frame	$H \times W \times 3$	Input
P_n^t	Masked RGB input for proto-object n	$H \times W \times 3$	Input
Student Network Outputs			
$\mathbf{m}^t$	Middle-layer representation	$\mathbb{R}^{D_1}$	Student
$\mathbf{z}^t$	Penultimate-layer feature	$\mathbb{R}^{D_2}$	Student
$\mathbf{f}^t$	Final-layer feature (unmasked input)	$\mathbb{R}^{D_3}$	Student
$\mathbf{f}_n^t$	Individual proto-object feature	$\mathbb{R}^{D_3}$	Student
$\mathbf{f}_{\text{agg}}^t$	Aggregated compositional feature	$\mathbb{R}^{D_3}$	Student
$\hat{D}^t$	Predicted depth map	$H \times W$	Student
Teacher Network Outputs & Pseudo-Labels			
D^t	Pseudo-ground truth depth map	$H \times W$	frozen backbone [71]
$\mathbf{z}'^t$	Penultimate-layer feature	$\mathbb{R}^{D_2}$	Teacher
$\mathbf{f}'^t$	Final-layer feature	$\mathbb{R}^{D_3}$	Teacher
$\mathbf{e}^t$	Patch embedding	$S \times D$	Teacher
$\mathbf{q}_n^t$	Query of the n-th proto-object	$\mathbb{R}^{D_q}$	Teacher
$\mathbf{k}_n^t$	Key of the n-th proto-object	$\mathbb{R}^{D_k}$	Teacher
$\mathbf{A}_n^t$	Attention map of proto-object n	$h \times w$	Teacher
Method-specific Variables			
w_n^t	Importance weight for proto-object n	scalar	Teacher
Mask_n^t	Spatial mask for proto-object n	$H \times W$	Teacher
$M_n^{(t,t')}$	Temporal validity mask	—	Teacher
$\mathcal{T}$	Sampled proto-object subset	—	—
$\mathcal{P}$	Set of valid proto-objects ($\mathcal{L}_{\text{temp}}$)	—	—
T_{clip}	Total number of frames in the clip	integer	—

At each training iteration, for every video sequence $\{X^t\}_{t=1}^T$, we randomly sample a subset of attention heads $\mathcal{T} \subset \{1, \ldots, N\}$ of size K (set to $K = 3$ in our experiments). This sampling strategy acts as a regularization mechanism, by forcing the student to form consistent scene representations from varying partial subsets of proto-objects.

Subset-wise Importance Weighting. For each head n within the sampled subset $\mathcal{T}$, we compute its relative importance using the teacher's attention maps. We first derive a saliency score s_n^t by averaging the attention values over all spatial tokens (excluding the `[CLS]` token):

$$s_n^t = \mathbb{E}_{\text{spatial}}[\mathbf{A}_n^t] = \frac{1}{S} \sum_{i=1}^{S} \mathbf{A}_{n,i}^t, \quad \forall n \in \mathcal{T}, \tag{8}$$

where $\mathbf{A}_n^t \in \mathbb{R}^{1 \times S}$ denotes the teacher's spatial attention map for head n at time t, where S is the number of spatial tokens. Heads whose attention mass collapses to the `[CLS]` token tend to yield low spatial averages s_n^t, making s_n^t a simple proxy for their semantic usefulness. We then compute the normalized importance weights w_n^t by applying a temperature-controlled softmax strictly over the sampled heads in $\mathcal{T}$:

$$w_n^t = \frac{\exp(s_n^t / \tau_w)}{\sum_{k \in \mathcal{T}} \exp(s_k^t / \tau_w)}, \quad \forall n \in \mathcal{T}, \tag{9}$$

where τ_w is a temperature parameter (set to 0.1 in our implementation), ensuring that the weights sum to 1 within the current sampling context.

Local-to-Global Aggregation. Given the student's proto-object features $\{\mathbf{f}_n^t\}_{n \in \mathcal{T}}$, where each $\mathbf{f}_n^t \in \mathbb{R}^D$ is the head-specific proto-object representation defined in Eq. (7), we aggregate them using the derived weights:

$$\mathbf{f}_{\text{agg}}^t = \sum_{n \in \mathcal{T}} w_n^t \cdot \mathbf{f}_n^t. \tag{10}$$

The resulting vector $\mathbf{f}_{\text{agg}}^t \in \mathbb{R}^D$ serves as the student's reconstruction of the scene based on the selected proto-objects.

A.3. Depth Loss Details

To effectively leverage geometric priors, we employ a composite loss, $\mathcal{L}_{\text{depth}}$. This loss is designed to distill structural information into our video encoder while remaining robust to the scale and shift ambiguities inherent in pseudo-labels. It consists of a global alignment term ($\mathcal{L}_{\text{si}}$) and a gradient consistency term ($\mathcal{L}_{\text{grad}}$), both computed in the linear depth space consistent with our normalized output range $[0, 1]$.

The first component is the **Scale-Invariant Loss ($\mathcal{L}_{\text{si}}$)**. We implement this term in linear space to specifically penalize relative structural errors while ignoring global offset discrepancies. Let $d^t = \hat{D}^t - D^t$ be the pixel-wise difference between the predicted depth $\hat{D}^t$ and the teacher's depth prior D^t. The loss is defined as:

$$\mathcal{L}_{\text{si}} = \frac{1}{\Omega}\sum_{i=1}^{\Omega}(d_i^t)^2 - \beta \cdot \left(\frac{1}{\Omega}\sum_{i=1}^{\Omega} d_i^t\right)^2 \tag{11}$$

Here, Ω denotes the total number of valid pixels in the frame X^t. Following our implementation, we set $\beta = 0.5$.

The second component, the **Gradient Consistency Loss ($\mathcal{L}_{\text{grad}}$)**, encourages the student to capture high-frequency geometric details such as object boundaries. We compute the Mean Absolute Error (MAE) between the spatial gradients of the prediction and the prior:

$$\begin{aligned}\mathcal{L}_{\text{grad}} &= \text{MAE}(\nabla_x \hat{D}^t, \nabla_x D^t) + \text{MAE}(\nabla_y \hat{D}^t, \nabla_y D^t) \\ &= \frac{1}{n_x}\sum_{i=1}^{n_x}\left|\nabla_x \hat{D}_i^t - \nabla_x D_i^t\right| \\ &\quad + \frac{1}{n_y}\sum_{i=1}^{n_y}\left|\nabla_y \hat{D}_i^t - \nabla_y D_i^t\right|\end{aligned} \tag{12}$$

where ∇_x and ∇_y are first-order finite difference operators. n_x and n_y represent the number of valid gradient components in the horizontal and vertical directions, respectively.

The total depth loss is the weighted sum $\mathcal{L}_{\text{depth}} = \mathcal{L}_{\text{si}} + \lambda_{\text{grad}}\mathcal{L}_{\text{grad}}$, with $\lambda_{\text{grad}} = 1.0$. This regularization explicitly guides the model to learn geometrically grounded representations without requiring manual annotations.

A.4. Temporal Proto-object Consistency Learning

For clarity, we denote the second frame as t', which corresponds to a temporal step $t + w$ relative to the first frame t, where $|w| \leq W$ is the offset within the temporal window W.

To ensure representations are consistent over time, we introduce a teacher-filtered temporal contrastive loss. Within a temporal window W, for any two frames t and t', we first assess the reliability of correspondence using the teacher network. Specifically, we compute the cosine similarity between penultimate-layer features $\mathbf{z}_n'^t$ and $\mathbf{z}_n'^{t'}$:

$$\text{sim}(\mathbf{z}_n'^t, \mathbf{z}_n'^{t'}) = \frac{\mathbf{z}_n'^t \cdot \mathbf{z}_n'^{t'}}{\|\mathbf{z}_n'^t\|_2 \cdot \|\mathbf{z}_n'^{t'}\|_2}. \tag{13}$$

We then define a validity mask $M_n^{(t,t')}$ by thresholding this similarity with a confidence threshold λ:

$$M_n^{(t,t')} = \mathbb{1}[\text{sim}(\mathbf{z}_n'^t, \mathbf{z}_n'^{t'}) > \lambda], \tag{14}$$

where $\mathbb{1}[\cdot]$ is the indicator function. This mask is 1 only if the teacher deems the correspondence reliable.

For every valid pair $M_n^{(t,t')} = 1$, we apply a contrastive loss. We treat the student feature $\mathbf{z}_n^{t'}$ as the query, the matching teacher feature $\mathbf{z}_n'^t$ as the positive key, and all other teacher proto-object features at time t, $\mathbf{z}_k'^t$ for $k \neq n$, as negative keys. The loss for this valid pair (t, t') is:

$$\mathcal{L}_{\text{temp}}^{(t,t')} = -\frac{1}{|\mathcal{P}|}\sum_{n\in\mathcal{P}}\log\frac{\exp(\text{sim}(\mathbf{z}_n^{t'}, \mathbf{z}_n'^t)/\alpha)}{\sum_{k=1}^{\mathcal{K}}\exp(\text{sim}(\mathbf{z}_n^{t'}, \mathbf{z}_k'^t)/\alpha)} \tag{15}$$

Here, $\mathcal{P} = \{n \mid M_n^{(t,t')} = 1\}$ is the set of valid proto-objects, $|\mathcal{P}|$ is the count of valid proto-objects, $\mathcal{K}$ is the total number of proto-objects, and α is a temperature parameter.

Total Temporal Loss Aggregation The total temporal loss $\mathcal{L}_{\text{temp}}$ is the average of the single-pair losses $\mathcal{L}_{\text{temp}}^{(t,t')}$ over all possible pairs within the clip. Assuming the clip length is T_{clip}, the final loss is defined as:

$$\mathcal{L}_{\text{temp}} = \frac{1}{N_{\text{pairs}}}\sum_{t=1}^{T_{clip}}\sum_{w=1}^{W}\mathcal{L}_{\text{temp}}^{(t,t+w)} \tag{16}$$

where N_{pairs} is the total number of valid sampled temporal pairs $(t, t + w)$ in the clip, constrained by $t + w \leq T_{clip}$. Note that the summation uses the explicit time offset w for clarity in defining the aggregation window.

A.5. Complete Algorithm

We present the complete algorithm of EgoViT in Algorithm 2.

B. Depth Decoder Architecture

In this section, we present a detailed description of the **Depth Decoder** architecture employed in our *Depth Regularization* module. The decoder is specifically designed to process intermediate feature representations $\mathbf{m}$ from the student encoder while maintaining computational efficiency. The input feature $\mathbf{m}$ has a shape of $C_{\text{in}} \times H' \times W'$ (where $C_{\text{in}} = D_1$ from Table 6). Our decoder employs a progressive upsampling strategy, gradually recovering spatial details through multiple stages while reducing channel dimensionality.

Algorithm 2 EgoViT – Full Pseudocode of One Training Iteration

Input: Input video clip $\{X^t\}_{t=1}^{T_{clip}}$, student parameters θ, teacher parameters θ'.
Output: Updated parameters θ and θ'.
1. Forward pass on original images for global features & mask generation
  for each frame t in the clip $\{X^t\}_{t=1}^{T_{clip}}$ **do**
    $(\mathbf{f}^t, \mathbf{m}^t) \leftarrow \text{Student}_\theta(X^t)$ ▷ Get final $\mathbf{f}^t$, and middle $\mathbf{m}^t$ features
    $(\mathbf{f}'^t, \mathbf{A}^t) \leftarrow \text{Teacher}_{\theta'}(X^t)$ ▷ Get features and attention maps
  end for
2. Generate a batch of masked images for proto-objects
  Generate masks $\{\text{Mask}_n^t\}$ from teacher attention maps $\{\mathbf{A}^t\}$.
  Create a set of masked images $\{P_n^t \leftarrow X^t \odot \text{Mask}_n^t\}$.
3. Forward pass on masked images to get object-specific features
  $\{\mathbf{f}_n^t\}, \{\mathbf{z}_n^t\} \leftarrow \text{Student}_\theta(\{P_n^t\})$ ▷ Individual proto-object final features $\mathbf{f}_n^t$ & bottleneck features $\mathbf{z}_n^t$
  $\{\mathbf{z}_n'^t\} \leftarrow \text{Teacher}_{\theta'}(\{P_n^t\})$ ▷ Teacher bottleneck features for temporal filter $\mathbf{z}_n'^t$
4. Compute all loss terms
  Compute depth loss $\mathcal{L}_{\text{depth}}$ from intermediate features $\{\mathbf{m}^t\}$.
  Compute proto-object loss $\mathcal{L}_{\text{proto}}$:
    a. Global alignment $H(\mathbf{f}'^t, \mathbf{f}^t)$.
    b. Compositional alignment $H(\mathbf{f}'^t, \mathbf{f}_{\text{agg}}^t)$ using $\{\mathbf{f}_n^t\}$.
    Set $\mathcal{L}_{\text{proto}} = H(\mathbf{f}'^t, \mathbf{f}^t) + H(\mathbf{f}'^t, \mathbf{f}_{\text{agg}}^t)$.
  Compute temporal loss $\mathcal{L}_{\text{temp}}$ using filtered pairs from $\{\mathbf{z}_n^t\}$ and $\{\mathbf{z}_n'^t\}$.
5. Joint Optimization and Model Update
  Aggregate the total loss: $\mathcal{L}_{\text{total}} = \gamma_P \mathcal{L}_{\text{proto}} + \gamma_D \mathcal{L}_{\text{depth}} + \gamma_T \mathcal{L}_{\text{temp}}$.
  Update student parameters θ via backpropagation on $\mathcal{L}_{\text{total}}$.
  Update teacher parameters θ' using an Exponential Moving Average (EMA) of θ.

Architecture Overview. The decoder begins with the input feature tensor $\mathbf{m}$ and processes it through an initial stage of refinement. The architecture can be formally described as:

$$F_1 = \text{GELU}(\text{GN}_8(\text{Conv}_{3\times3}(\mathbf{m}, C_{\text{in}} \to 256))) \tag{17}$$

where $\text{Conv}_{3\times3}(\text{C}_{\text{in}} \to 256)$ denotes a 2D convolution with kernel size 3 and padding 1, GN_8 represents Group Normalization with 8 groups, and GELU is the activation function.

Progressive Upsampling. The decoder comprises four sequential upsampling stages, each following the structure:

$$F_{i+1} = \text{GELU}(\text{GN}_{k_i}(\text{ConvTranspose}_{3\times3}(F_i))) \tag{18}$$

for $i \in \{1, 2, 3, 4\}$, where $k_i = \min(8, C_{\text{out}})$ is the number of groups in Group Normalization, adaptively set based on the output channel dimension C_{out}. The $\text{ConvTranspose}_{3\times3}$ operation uses a stride of 2, padding of 1, and an output padding of 1 to ensure the spatial resolution is exactly doubled at each stage. The channel dimensions progressively decrease through the stages as follows: $256 \to 128 \to 64 \to 32 \to 16$.

Final Depth Prediction. The final depth map $\hat{D}^t$ is produced through a convolution with output channel 1, followed by a sigmoid activation to normalize the output to the range $[0, 1]$:

$$\hat{D}^t = \sigma(\text{Conv}_{3\times3}(F_5, 1)) \tag{19}$$

where σ denotes the sigmoid activation function, and the final $\text{Conv}_{3\times3}$ outputs 1 channel.

Initialization Strategy. We employ He initialization for all convolutional and transposed convolutional layers to ensure stable training:

$$w \sim \mathcal{N}\left(0, \frac{2}{n_{\text{in}}}\right) \tag{20}$$

where n_{in} is the number of input units in the weight tensor. The Kaiming (He) initialization uses the assumption of a ReLU non-linearity (nonlinearity='relu') as standard practice in PyTorch for deep convolutional networks.

Normalization and Activation. The use of Group Normalization instead of Batch Normalization makes the model more robust to varying batch sizes and provides consistent performance across different training configurations. The GELU activation was chosen for its smooth characteristics and compatibility with transformer-based architectures.

C. Experimental Details

This appendix provides supplementary details for our experimental setup, including pre-training data, baselines, im-

plementation hyperparameters, and downstream task protocols.

C.1. Pre-training Datasets

We use videos from the Walking Tours (WT) dataset [62], which consists of 10 long-form, first-person videos captured in various cities worldwide. All videos were recorded at 4K resolution and 60 FPS, providing high-quality and temporally dense data for self-supervised pre-training. For our experiments, we selected a subset of these videos, as detailed in Table 7. This selection was made to expose our model to diverse visual environments, approximating open-world learning conditions.

Table 7. Details of egocentric videos used for pre-training. All videos are 4K at 60 FPS. The primary video is marked in bold; all others are used for augmentation.

Video Name	Duration (min)
WT-Zurich	**∼ 65**
WT-Istanbul	∼73
WT-Stockholm	∼68
WT-Chiang Mai	∼70
WT-Kuala Lumpur	∼66
WT-Venice	∼110
WT-Amsterdam	∼82
WT-Bangkok	∼175
WT-Singapore	∼97
WT-Wildlife	∼60

C.2. Architecture and Training Configuration

We adopt ViT-S/16 [20] as our backbone architecture, consisting of a 12-layer transformer with embedding dimension 384 and 6 attention heads per layer. We randomly sample video clips of $T = 8$ frames with temporal separation of 1 second (i.e., one frame every 60 frames at 60 FPS) for each mini-batch.

To fully leverage high-resolution 4K video content, we first crop a 640×640 region from each frame at a scale from 0.4 to 1, and then apply a multi-crop strategy at a relatively small scale ranging from 0.15 to 0.3. This scale range maintains the balance between object diversity and visual clarity. We use only two global crops per frame without additional local crops combined with cross-entropy loss, as videos inherently contain numerous irrelevant objects that could introduce excessive noise during training. Additionally, we apply masking to the global crops fed into the student model to better facilitate learning of robust proto-object representations.

For optimization, we employ AdamW with base learning rate $\eta = 5 \times 10^{-4}$, initial weight decay $\lambda_{\text{wd}} = 0.04$, and linear warm-up over the first 10 epochs. Given the temporal nature of video data, we define one epoch as a complete traversal of the video dataset and train for 320 epochs by default. All experiments use a global batch size of 256.

C.3. Proto-object Learning Hyperparameters

Table 8. Hyperparameters for proto-object learning framework.

Parameter	Symbol	Value
Student Temperature	α_{student}	0.1
Teacher Temperature (Final)	α_{teacher}	0.04
Teacher Similarity Threshold	λ	0.8
Sampled Proto-objects (K)	K	3
Temporal Window Size	W	4
Teacher momentum coefficient	m	0.996

Table 8 summarizes the key hyperparameters for our proto-object learning framework. Note that the Teacher Momentum m follows a cosine schedule, increasing towards 1.0 during training.

C.4. Downstream Task Evaluation Protocols

Classification We evaluate the classification capabilities of EgoViT and several baseline detection models pretrained on the Zurich dataset using ImageNet-1k as the downstream benchmark. Specifically, EgoViT is pretrained on Zurich following the approach described in Sec. C. For fair comparisons, the baseline models are pretrained similarly, with an initial crop of 640×640 pixels taken from each frame, while subsequent data augmentations and training procedures follow the original baseline methodologies. In the pretraining stage, we employ AdamW optimization with a global batch size of 256, a base learning rate of 5×10^{-4}, and a minimum learning rate of 1×10^{-6}.

For downstream classification evaluation, we perform two standard tasks: linear probing and k-nearest neighbor (k-NN) classification. In the linear probing setting, we follow the evaluation protocol of Caron et al. [11]. Specifically, we freeze the pretrained backbone features and train a linear classifier under supervised conditions on the ImageNet-1K training set, using a batch size of 1024. Performance is reported as top-1 accuracy (%) on the ImageNet-1K validation set. For k-NN classification, we again freeze the pretrained backbone to extract features from the ImageNet-1k training set and apply a k-nearest neighbor classifier with k=20. We report top-5 accuracy (%) as the primary evaluation metric for comparison.

Object Discovery Following LOST [57], we extract and average the self-attention maps from the final layer of our pretrained ViT-S/16, retaining 80% of the total attention mass. We evaluate object localization performance on the Pascal VOC 2012 [22] dataset, consisting of 11,540 images, using the CorLoc metric. CorLoc measures the localization accuracy as the percentage of correctly predicted bounding boxes, where a prediction is considered correct if its intersection-over-union (IoU) with the ground truth bounding box is greater than or equal to 0.5.

Object Detection and Instance Segmentation Due to computational constraints, we evaluate EgoViT for object detection and instance segmentation on the Mini COCO dataset [55], a category-balanced subset of MS COCO [41] that effectively reflects model performance on the complete dataset. Specifically, we use ViT-S/16 as our backbone network, following the approach described in iBOT [75], and apply a multi-scale training strategy. During training, input images are randomly resized, with their shorter sides ranging between 480 and 800 pixels while ensuring the longer side does not exceed 1333 pixels. The entire network is fine-tuned using a standard $1\times$ schedule (12 epochs in total), with an initial learning rate of 1×10^{-4}, weight decay of 0.05, and learning rate decay by a factor of 10 at epochs 9 and 11. Moreover, we explore different layer-wise learning rate decay values, specifically $\{0.65, 0.75, 0.8, 0.9\}$, where a decay value of 1.0 indicates no layer-wise decay.

To construct hierarchical feature representations, we adapt the standard ViT-FPN conversion used in DINO. We extract features from layers 4, 6, 8, and 12 of the backbone, mapping them to standard FPN levels (P_2, P_3, P_4, P_5 strides). Concretely, we perform two successive deconvolutions on the features from layer 4 to reach the highest resolution, a single deconvolution on layer 6 features, identity mapping on features from layer 8, and max-pooling to downsample features from layer 12. This process converts the single-scale ViT output into a multi-scale FPN suitable for detection and segmentation tasks.

Semantic Segmentation For the Semantic Segmentation task, we fine-tune the model on ADE20K [74] using a UperNet segmentation head for 160K iterations. Our experimental settings closely follow the procedure introduced in BEiT [7]. Specifically, we employ the AdamW optimizer with an initial learning rate of 6×10^{-5} and a weight decay of 1×10^{-2}. A linear warm-up schedule is applied during the first 1,500 iterations. The model is fine-tuned with a batch size of 4.

Video Object Segmentation For evaluating the performance of EgoViT on the video object segmentation task, we utilize the DAVIS 2017 dataset [51]. Following the evaluation protocol described in DINO [11], segmentation is performed on video frames at 480p resolution, each containing between two and four distinct objects. We report performance using mean region-based similarity (J_m) and mean contour-based accuracy (F_m) metrics.

D. On the Reproducibility of the DoRA Baseline

To establish a fair and rigorous comparison, we made a significant effort to reproduce the results of our primary video-based baseline, DoRA. This section details our reproduction process and findings.

Table 9. Feature extraction depth evaluation on k-NN and CORLOC metrics. Shallower layers (depth 3–4) better preserve spatial information while maintaining comparable semantic features.

METHOD	Depth	k-NN	CORLOC
DINO	×	33.7	28.6
EgoViT-D	3	34.5	**40.6**
EgoViT-D	4	**35.0**	39.8
EgoViT-D	5	34.3	35.2
EgoViT-D	6	34.9	37.2
EgoViT-D	8	34.8	39.0
EgoViT-D	12	35.2	26.7

Our process was based on the official source code (commit hash: `DoRA_ICLR24`) and we meticulously followed the experimental settings described in their paper, as detailed in our implementation setup in Sec. C.

Despite these efforts, we observed a notable discrepancy between our reproduced results and those reported in the original paper. This gap suggests a high sensitivity to specific, unstated details of the training environment or data preprocessing pipeline. For instance, on ADE20k semantic segmentation, our implementation achieved 21.6 mIoU, compared to the 35.4 mIoU reported in the original work. Similarly, for ImageNet-1k linear probing, we obtained 29.6% accuracy, whereas the original work reported 44.5%.

We note that this reproducibility challenge is not unique to our experience. Similar difficulties have been reported by other researchers in public forums, such as the issues section of the official DoRA GitHub repository (e.g., Issue #1, #3, #4, and #5).

Therefore, to maintain a controlled and scientifically valid comparison, all baseline results presented in our main paper are generated from our own implementation within a unified execution environment. This ensures that the performance gains of our proposed method, EgoViT, are evaluated against a consistently implemented and directly comparable baseline, providing a true measure of its advancements.

E. Additional Experimental Results

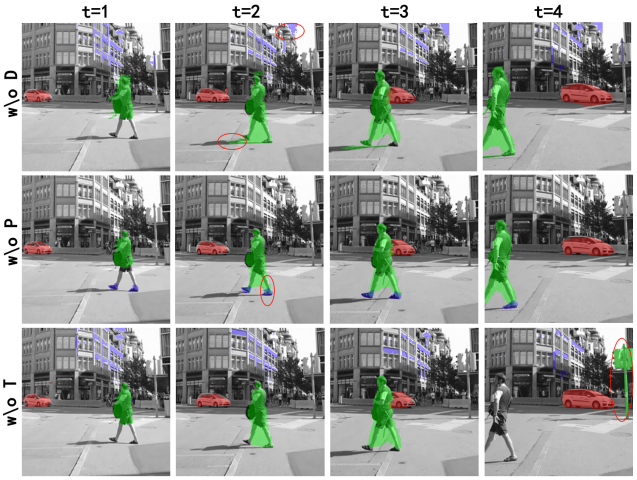
E.1. Ablation Study Details

Similarity threshold and window size analysis Table 10 reveals that higher similarity thresholds (λ: 0.8-0.9) generally yield better performance in our temporal proto-object learning module. This suggests that stricter matching criteria lead to more reliable temporal associations between proto-objects, supporting our design choice to focus on high-confidence object correspondences across frames. We also observe that classification performance (k-NN) is less sensitive to the specific threshold, achieving its optimum at $W = 4$ and $\lambda = 0.8$ (22.9%).

Table 10. Ablation on temporal modeling with different thresholds (λ) and temporal window sizes (W). CORLOC benefits from stricter correspondence filtering.

Threshold and Window Size Effect			
W	λ	k-NN	CORLOC
3	0.5	20.8	32.2
3	0.7	22.6	36.2
3	0.8	22.7	35.9
3	0.9	22.1	37.5
4	0.7	22.2	37.0
4	0.8	**22.9**	37.9
4	0.9	22.2	**38.0**
5	0.7	22.0	35.2
5	0.9	22.0	35.4

Figure 7. Ablation visualization of EgoViT.



Feature extraction depth As shown in Table 9, the choice of feature extraction layer significantly impacts model performance. Shallower layers (depth 3-4) yield optimal results for both metrics, substantially outperforming the DINO baseline. Interestingly, the deepest layer (12) maintains strong classification performance but performs poorly on localization (26.7% CORLOC). This suggests a clear trade-off: shallower layers preserve spatial information critical for object localization, while deeper layers capture abstract semantic features beneficial for classification.

Qualitative Analysis of Learned Components Fig. 7 shows: w/o D proto-objects attach to more background, w/o P heads attend to the same object, and w/o T assignments become unstable across frames.

E.2. Other Experimental Results

We present the comprehensive experimental results of EgoViT compared to state-of-the-art self-supervised methods. Our analysis is structured around quantifying the performance benefits derived from our **Proto-Consistency** paradigm, particularly focusing on object localization and temporal generalization.

E.2.1. Performance on Downstream Tasks

In this section, we provide additional comparisons that were excluded from the main text to ensure strictly controlled experimental conditions. We focus on two aspects: (1) comparison with specialized architectures, and (2) the scalability of EgoViT across different data regimes.

Comparison with Specialized Architectures (SAVi++, PooDLe). In the main paper, we restricted our comparison to methods utilizing standard ViT-S backbones. Here, we extend the evaluation to include SAVi++ and PooDLe in the discussion. The former is a representative work that relies on slots for self-supervised learning, and the latter is a self-supervised paradigm aimed at object segmentation on ResNet-50. As shown in Table 11, although SAVi++ uses slots for object segmentation in its architecture, EgoViT$_{\text{Zurich}}$, based on the standard ViT, unexpectedly outperforms it in the segmentation task. This result suggests that our Proto-Consistency objective effectively induces object-centric features within standard transformer architectures, without requiring complex slot-based modules. While PooDLe shows strong performance in Semantic Segmentation, it significantly underperforms in temporal tasks (VOS), whereas EgoViT maintains balanced performance across all metrics. Notably, the official PooDLe paper reports in its appendix that the ViT-S variant tends to collapse during video-based self-supervised training, which is why their semantic segmentation evaluation relies on a ResNet-50 backbone instead. Since ResNet-50 typically exhibits stronger performance than ViT-S under similar settings, this backbone discrepancy may partially account for PooDLe outperforming our EgoViT on semantic segmentation tasks.

Scalability Analysis (Sub-5 and Full Data). To demonstrate the data efficiency and scalability of our approach, we evaluate EgoViT on three progressively larger data scales:

1. **Single Video:** Models trained on individual scenes (*Zurich*, *Venice*).
2. **WT-Sub5 (5 Videos):** An intermediate scale trained on a curated subset of five videos to test generalization across diverse environments. This subset includes: *Zurich, Venice, Istanbul, Stockholm, and Chiang Mai*.
3. **WT-All (Full Dataset):** The model trained on the entire available dataset.

Trend Analysis: The bottom section of Table 11 reveals a consistent upward trend. Moving from single-video training to the WT-Sub5 set yields immediate gains (e.g., +3.0% mIoU over Zurich), confirming that the model benefits from increased visual diversity. Furthermore, EgoViT$_{\text{WT-All}}$

Table 11. Performance comparison of EgoViT against state-of-the-art methods across a range of downstream tasks. Our main model, **EgoViT$_{\text{Zurich}}$**, is trained on a single 65-minute video, while EgoViT$_{\text{WT-Sub5}}$ is an additional model trained on five videos to demonstrate scalability. Models without an explicit subscript are trained on the default Zurich video unless otherwise noted. Except for PooDLe, which uses a ResNet-50 backbone (weights obtained from the official release), all other baselines and our models use a ViT-S architecture for fair comparison.

METHOD	Semantic SEG.		Object DET.	Instance SEG.	Video Object Segmentation			Object DIS.	Classification	
	mIoU	Acc$_m$	mAP	mAP	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$	CORLOC	LP	k-NN
SimCLR	22.5	32.8	22.2	20.1	52.3	51.5	53.1	35.5	28.2	33.1
AttMask	25.1	35.4	25.9	23.9	52.2	52.5	51.8	37.8	25.4	35.9
MoCo-v3	17.9	26.0	19.0	17.3	52.5	50.9	54.0	43.1	22.5	31.2
MAE	23.0	33.2	24.6	22.1	51.3	50.3	52.2	35.9	13.0	17.8
iBOT	23.9	34.5	22.1	19.6	53.9	53.5	54.3	36.3	27.5	33.6
SAVi++	24.4	35.7	24.7	23.4	52.0	50.8	53.2	34.2	29.3	31.2
DORA	21.6	31.1	22.6	20.4	53.8	51.9	55.6	24.1	29.6	33.7
DORA$_{\text{Venice}}$	22.4	32.3	23.2	21.2	53.0	51.7	54.2	23.9	30.2	34.8
PooDLe$_{\text{Venice}}$	33.5	41.6	26.2	23.7	19.7	21.4	17.8	32.7	28.9	30.8
DINO	21.2	30.3	22.0	20.6	53.8	52.5	55.1	37.2	30.9	35.5
EgoViT$_{\text{Venice}}$	27.1	37.5	26.2	24.6	54.5	52.7	56.4	45.4	35.8	39.2
EgoViT	26.0(+4.8)	36.6 (+6.3)	26.7 (+4.7)	24.3(+3.7)	54.3 (+0.5)	52.7 (+0.2)	55.9 (+0.8)	45.2 (+8.0)	34.0 (+3.1)	38.9 (+3.4)
EgoViT$_{\text{WT-Sub5}}$	**29.0**(+7.8)	**38.7** (+8.4)	**28.2** (+6.2)	**26.2**(+5.6)	**56.1** (+2.3)	**54.2** (+1.7)	**57.8** (+2.7)	**48.9** (+11.7)	**37.2** (+6.3)	**42.7** (+7.2)
EgoViT$_{\text{WT-all}}$	**30.6**(+9.4)	**39.3** (+9.0)	**29.6** (+7.6)	**26.8**(+6.2)	**57.0** (+3.2)	**55.0** (+2.5)	**58.9** (+3.8)	**50.2** (+13.0)	**39.1** (+8.2)	**45.3** (+9.8)

Table 12. DINO, DoRA, and EgoViT (Zurich-pretrained) backbones are evaluated under the OSTrack framework on TrackingNet and GOT-10k.

TrackingNet			
Metric	DINO	DoRA	EgoViT
AUC	76.2	77.7	**78.9**
P_Norm	81.2	82.5	**83.5**
P	73.7	74.7	**77.6**
GOT-10k (zero-shot)			
Metric	DINO	DoRA	EgoViT
AO	61.6	63.8	**67.0**
SR$_{0.5}$	71.9	73.8	**77.0**
SR$_{0.75}$	59.5	62.4	**65.5**

achieves the best overall performance, indicating that our framework scales effectively with data volume and has not yet reached saturation.

E.2.2. Temporal Generalization on Object Tracking

Validation through Object Tracking. Table 12 evaluates the robustness of the learned features on two standard tracking benchmarks, TrackingNet and GOT-10k, using the OSTrack framework. On TrackingNet, EgoViT achieves the best performance across all three metrics, with an AUC of 78.9% and consistently higher P and P_Norm scores than both DINO and DoRA. On GOT-10k in the challenging zero-shot setting, the gap is even clearer: EgoViT attains an AO of 67.0% together with the highest SR$_{0.5}$ and SR$_{0.75}$, indicating stronger generalization to unseen targets and motion patterns.

These consistent improvements across two datasets with different object categories and motion patterns provide strong evidence that our design, which combines *explicit temporal consistency supervision* with a depth-based geometric regularizer, leads to more robust and persistent object representations than purely spatial or view-based consistency. The resulting geometry-regularized and temporally filtered prototypes learned by EgoViT transfer effectively to downstream trackers, enabling more reliable cross-frame identity association and long-term tracking.

Table 13. EPIC-KITCHENS VISOR VOS results. All methods use a ViT-S/16 backbone.

Method	Pretrain	VOS Fine-Tune	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$	$(\mathcal{J}\&\mathcal{F})_{unseen}$
Baseline	–	✓	55.5	53.9	57.1	49.5
DNIO	WT$_{\text{Zurich}}$	✓	62.4	61.5	63.3	56.2
DORA	WT$_{\text{Zurich}}$	✓	63.3	62.2	64.3	57.4
EgoViT	WT$_{\text{Zurich}}$	✓	**69.4**	**67.8**	**70.9**	**64.3**

E.2.3. Performance on Egocentric Video Benchmark: Epic-Kitchens VISOR

We follow the Epic-Kitchens VISOR VOS protocol [17], with only the backbone replaced. Table 13 shows EgoViT significantly outperforms baselines (e.g., +6.1% in $\mathcal{J}\&\mathcal{F}$; +6.9% on unseen subset). These gains indicate that EgoViT learns stable representations for egocentric perception.

F. More Visualization

Qualitative analysis of temporally consistent proto-objects. Figure 8–10 visualize EgoViT's predictions across 8 consecutive egocentric frames for diverse scenarios with dynamic objects and viewpoint shifts.

In Figure 8, EgoViT robustly maintains object identity for both a moving tram (red) and an approaching car (green), despite abrupt camera panning and strong background clutter (e.g., crosswalk stripes and shadows). The persistence of masks demonstrates the model's ability to filter motion-independent structure by leveraging depth and teacher-guided consistency.

In Figure 9, EgoViT successfully segments a person (green) and a luggage trolley (blue) in a crowded station scene with significant occlusion and illumination shifts.

Importantly, the assigned masks remain identity-consistent even as both objects deform or partially disappear, showing that EgoViT encodes proto-objects beyond mere appearance.

Figure 10 further highlights EgoViT's capacity to disambiguate multiple overlapping proto-objects (signboard, bag, suitcase), despite their similar texture and partial occlusions across frames. This illustrates the effectiveness of our depth-anchored proposal and temporal filtering modules in learning object-centric representations under self-supervision.

Overall, these results show that EgoViT goes beyond spatial saliency or appearance clustering: it learns **temporally persistent, semantically coherent object-level concepts** from egocentric video without requiring class labels.

G. Broader Impacts and Ethical Considerations

Our work on EgoViT introduces a self-supervised framework for learning temporally consistent object representations from egocentric video streams. This capability has implications across embodied AI, cognitive modeling, and potential real-world deployments. **We emphasize that EgoViT is presented as a research prototype designed to advance fundamental understanding in self-supervised learning.**

G.1. Potential Positive Impacts

EgoViT offers a foundation for more perceptually grounded embodied agents. By learning to track spatially coherent entities over time without supervision, EgoViT enables downstream models to develop object permanence and identity persistence—capabilities critical for long-horizon interaction, manipulation, and navigation in real-world environments.

Additionally, our approach reduces reliance on large-scale manual annotations, making it suitable for deployment in novel, open-ended scenarios where semantic labels are scarce or costly to obtain (e.g., in-situ robotic learning or home-scale exploration). The model's biologically inspired structure—linking depth cues with temporal attention—also provides a computational tool that may inform studies in human perception.

G.2. Risks and Mitigation Strategies

Egocentric visual data inherently contains sensitive information about individuals and personal environments. Deploying systems like EgoViT without safeguards may lead to privacy breaches, especially through bystander re-identification or context inference. We recommend future applications of EgoViT incorporate: (1) on-device processing, (2) anonymization pipelines (e.g., face blurring), and (3) user-controlled data access policies.

Further, the ability to stably track objects and infer scene structure could be misused in surveillance contexts. While EgoViT is intended for research and interaction-based learning, we advise usage restrictions and transparent model cards to guide ethical downstream applications.

G.3. Responsible Development

To promote responsible use, we release our code and models with clear licensing terms that discourage surveillance use. We are committed to continuing research in privacy-preserving self-supervised learning and encourage community engagement to identify and mitigate emergent risks.

H. Detailed Comparison and Positioning Analysis with DINO and DoRA

This appendix aims to precisely articulate the technical inheritance and paradigmatic distinction between our work and two pivotal prior works: DINO and DoRA. Our objective is to eliminate ambiguity regarding the contribution of this work.

H.1. Foundational Framework: Build upon DINO

The foundational training framework of EgoViT is built upon the self-distillation mechanism proposed in DINO [11]. We explicitly inherit its core Teacher-Student architecture, including the Exponential Moving Average (EMA) for parameter updates, the knowledge distillation loss, and the centering and sharpening strategies for the teacher's outputs. We adopt this mature framework to ensure training stability and efficiency. Our core innovation lies in the novel supervisory signals we provide to this framework.

H.2. Core differences between EgoViT and DoRA

The most fundamental distinction between EgoViT and DoRA lies in the self-supervised learning paradigms they follow. DoRA adopts a *Multi-view Spatial Consistency* paradigm as its core learning principle, whereas EgoViT introduces a new paradigm we term *Proto-Consistency*.

The DoRA Paradigm: Multi-view Spatial Consistency DoRA extends DINO's concepts from static images to video. Its core paradigm can be summarized as **Multi-view Spatial Consistency**. The supervisory signal primarily originates from the alignment of different spatial views generated from the same point in time:

- **Local-to-Global Alignment:** Requires features extracted from random local crops to align with features from a global view.

Figure 8. EgoViT maintains stable and distinct masks for the moving tram (red) and the pedestrian (green), effectively distinguishing them despite rapid camera panning and a cluttered background. When the pedestrian leaves the frame, the model seamlessly transitions to tracking the approaching car (also green). Additionally, EgoViT potentially exhibits partial capability in attending to small, distant objects, such as the signage (blue).

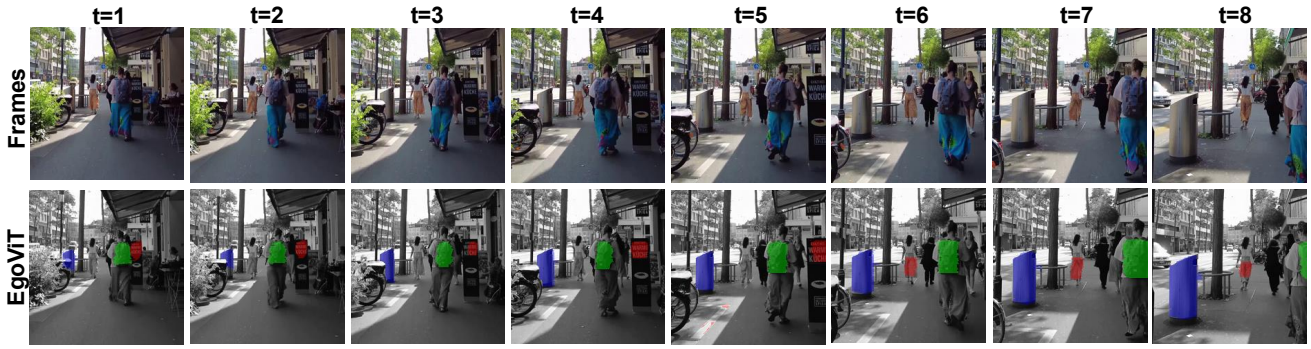


Figure 9. EgoViT maintains identity-consistent masks for the walking person (green) and the trash bin (blue), even as illumination, viewpoint, and crowd density vary. The text on the signage (red) is also consistently segmented across frames.

- **Masked-to-Global Alignment:** Utilizes self-attention maps to track salient image patches, generates a masked view, and requires features from this view to align with those of the global view.

Essentially, DoRA constructs its self-supervisory signal by creating and aligning different "views", making inter-view consistency the core of its learning process.

The EgoViT Paradigm: Proto-Consistency We posit that in the complex scenarios of egocentric video, view-based alignment faces significant challenges. Therefore, EgoViT introduces a new paradigm of **Proto-Consistency**. Our core objective is not to align different views, but to learn a robust set of prototypes and enforce their consistency across multiple dimensions, particularly over time. This principle forms the cornerstone of our multi-task learning framework, manifesting as:

- **Spatial Consistency:** Our 'proto_loss' requires the prototype features, obtained via soft aggregation, to be consistent with the global scene representation perceived by the teacher network.
- **Temporal Consistency:** Our 'temporal_loss' leverages cross-frame contrastive learning to ensure that the prototype representation of an object remains stable and consistent as time progresses.

In essence, the core of EgoViT's learning is the consistency of the prototype itself, which is jointly supervised across spatial and temporal dimensions through our multi-task objective.

Technical Implementation: Code-Level Evidence for the Two Paradigms This paradigmatic difference is substantiated by the causal relationship between **masked img** and **proto** in the respective implementations:

- In DoRA, the **proto** is the **cause** (a template for mask generation), and the **masked img** is the **effect** (the final product for data augmentation). The entire process serves to create a new "view".
- In EgoViT, the **masked img** is the **cause** (a proposal to delineate foreground regions), and the **proto** is the **effect**

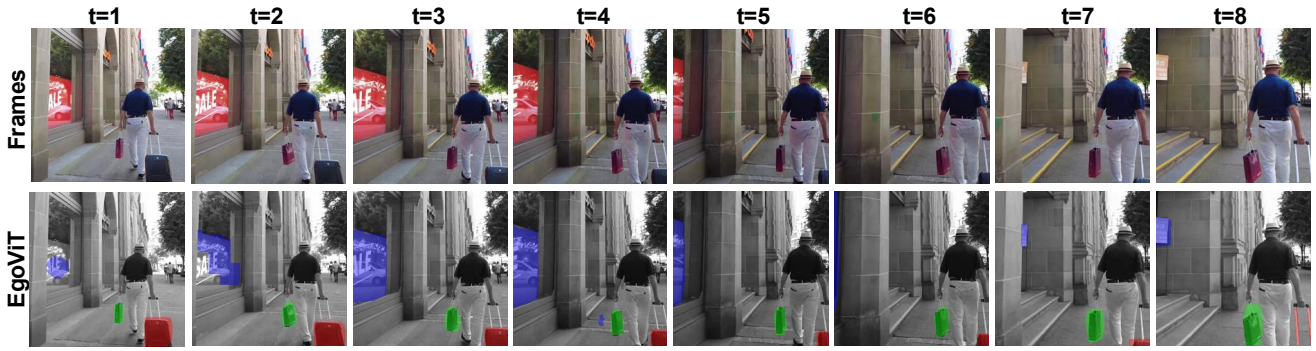


Figure 10. Even when the camera turns and the target becomes partially occluded, EgoViT tends to preserve relatively distinct proto-objects, such as the store sign (red), the shoulder bag (blue), and the suitcase (green). This may indicate that the model is robust to moderate viewpoint changes and occlusions.

(a direct learning objective that is optimized in the feature space). The entire process serves to learn the prototype itself.

Conclusion In summary, the two methods differ fundamentally in their core paradigms (Multi-view Consistency vs. Proto-Consistency), learning objectives (single-task vs. multi-task), and underlying design philosophies, as summarized in Table 14.

I. Zero-Shot Generalization: Ego4D Case Study

I.1. Motivation

Our main pre-training setup uses a single long Zurich city-walk video. This deliberately minimalist training domain naturally raises the question of how well the learned representations generalize to more diverse egocentric environments. In particular, we are interested in (i) larger egocentric corpora such as Ego4D, and (ii) scenarios with substantially different dynamics and visual conditions, such as cluttered indoor scenes and in-car low-light conditions. To investigate this, we conduct an additional zero-shot case study on Ego4D that primarily focuses on qualitative behavior: we keep the pre-training protocol fixed (single Zurich city-walk video) and only change the evaluation domain.

I.2. Ego4D-mini Benchmark and Setup

We construct a small but deliberately diverse Ego4D-mini benchmark by selecting three long clips from Ego4D(example frames are shown in Fig. 11):

- **E4D-Kitchen**: indoor scenes with strong hand–object interaction and heavy background clutter(ID: 2422d726-0286-48bc-96a6-fe29c45cc409);
- **E4D-Driving**: in-car sequences with low-light, motion blur, and dashboard/road objects(ID: 28170c86-29ba-43e8-8699-e76161f16b98);
- **E4D-HomeWalk**: domestic walking scenes with repeated occlusions by the wearer's body or carried objects(ID: 1635447d-f96f-4f1b-8e02-faaebcd8a6d2).

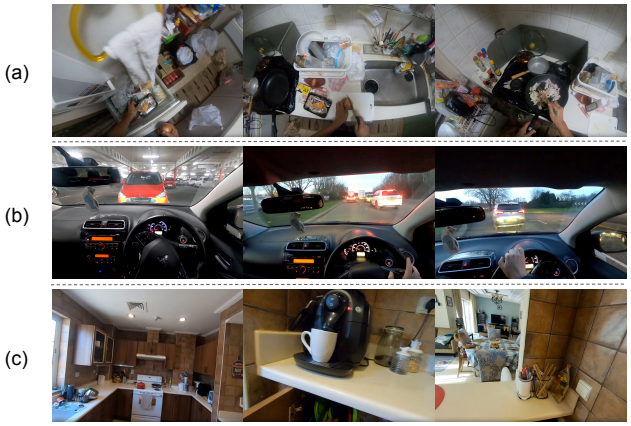


Figure 11. Example frames from the proposed Ego4D-mini benchmark. Row (a) shows cluttered kitchen scenes with strong hand–object interactions. Row (b) shows in-car driving sequences under low light and motion blur, with diverse dashboard and road objects. Row (c) shows domestic walking scenes. These clips are used only for evaluation; both DINO and EgoViT are pre-trained on a separate Zurich city-walk video.

We decode the video at 1 fps for testing. We reiterate that no Ego4D frame is used during pre-training: both DINO and EgoViT are trained only on our single Zurich city-walk video.

I.3. Pseudo Ground-Truth from SAM2

Ego4D does not provide generic bounding boxes for the main object of interest per frame. To obtain a rough notion of object locations without manual annotation, we use SAM2[53] to generate a dense set of instance masks on each

Table 14. Conceptual comparison between DoRA and EgoViT.

Dimension	DoRA	EgoViT (ours)
Core paradigm	Multi-view spatial consistency on global image embeddings.	Proto-consistency on object-centric prototypes distilled from attention heads.
Learning unit	Whole-scene representation; no explicit object-level carrier.	Proto-object prototype as the basic unit for representation and supervision.
Temporal modeling	Similar patches across time used only to create augmented views; the loss is purely spatial, without explicit temporal consistency.	Dedicated temporal loss that penalizes prototype drift and enforces identity persistence across frames.
Structure awareness	RGB-only, without geometric priors.	Auxiliary depth regularization encourages geometry-aware, structure-sensitive proto-object representations.

frame and convert them into bounding boxes.

Concretely, for each split $s \in \{\text{kitchen}, \text{driving}, \text{homewalk}\}$ and frame image I, we run the official Mask Generator and collect all masks whose area exceeds a small threshold to filter out tiny or noisy components. Each mask is converted to an axis-aligned bounding box, yielding a set:

$$\mathcal{B}_{\text{SAM2}}(I) = \{b_1, \ldots, b_K\},$$

where K is the number of boxes in frame I. These boxes are stored in JSON files per split. We found that SAM2 can under-segment cluttered regions or merge small objects, thus we treat its outputs as noisy pseudo ground-truth, primarily relying on them for visual inspection, not as a definitive quantitative benchmark.

I.4. Zero-Shot Box Prediction and Inference Protocol

We apply exactly the same LOST-style object discovery pipeline used in our VOC CorLoc experiments to the Ego4D-mini frames. The tested models are:

- **DINO**: ViT-S/16 backbone with official DINO pretraining;
- **EgoViT**: our proposed model (ViT-S/16). Both pretrained only on the single Zurich city-walk video.

The protocol involves extracting patch-level features, running the LOST algorithm to obtain predictions ($\mathcal{B}_{\text{DINO}}(I)$ or $\mathcal{B}_{\text{EgoViT}}(I)$), and storing the results per split.

I.5. Qualitative Observations

Each selected frame is visualized as a triplet figure: (1) SAM2 pseudo boxes (red), (2) DINO prediction (green), and (3) EgoViT prediction (blue). We consistently observe the following patterns, which match the design goal of EgoViT:

More Object-Centric Localization in Cluttered Indoor Scenes. In the Kitchen clip (Fig. 12), DINO often locks onto large, high-contrast background structures. By contrast, EgoViT tends to place tighter boxes around manipulated tools and objects near the hands, aligning more closely with salient object regions.

Figure 12. Qualitative comparison on the **Kitchen** clip. Each column corresponds to a different time step. From top to bottom: SAM2 pseudo boxes (red), DINO predictions (green), and EgoViT predictions (blue). EgoViT typically suppresses background clutter and localizes manipulated tools and food items near the hands, whereas DINO often locks onto larger static structures such as countertops or the sink area.

Improved robustness in low-light, dynamic in-car scenes. In the Driving clip (Fig. 13), low-light conditions and fast camera motion make the scene challenging. We observe that DINO's predictions sometimes drift to the dashboard, windshield borders, or large textureless areas, especially when motion blur is strong. EgoViT predictions more often remain on salient foreground entities such as the leading car or steering wheel, and exhibit better temporal stability when viewed as a sequence of frames.

Figure 13. Qualitative comparison on the **Driving** clip. The layout is the same as in Fig. 12. Under low light and strong camera motion, DINO frequently drifts to dashboard edges or windshield borders, while EgoViT more reliably tracks salient traffic participants and in-cabin control elements over time.

Figure 14. Qualitative comparison on the **HomeWalk** clip. The layout is the same as in Fig. 12. Due to frequent egocentric self-occlusions by the wearer's body and carried items, DINO often falls back to static background regions (walls, floors, furniture), whereas EgoViT maintains focus on object across occlusion and reappearance.

Consistent tracking of salient objects under egocentric occlusions. In the HomeWalk sequence (Fig. 14), the wearer's body and carried items repeatedly occlude parts of the scene. DINO occasionally switches to static background regions (walls, floors), while EgoViT more frequently maintains focus on the object being carried or manipulated across time, even when partial occlusions occur.

These trends match the design goal of EgoViT: by injecting depth cues and temporal consistency losses, the model learns to prefer depth-consistent, temporally stable proto-object regions that correspond to manipulable objects, rather than arbitrary textured patterns.

Table 15. CorLoc (%) on Ego4D-mini benchmark with SAM2 pseudo GT.

Split	EgoViT (%)	DINO (%)
Kitchen	10.51	11.63
Driving	16.61	12.11
HomeWalk	25.93	14.89

I.6. On Quantitative CorLoc with SAM2 Pseudo Boxes

We also experimented with a CorLoc-style quantitative evaluation where SAM2 pseudo boxes are treated as ground-truth and DINO/EgoViT boxes are counted as correct if they satisfy $\mathrm{IoU} \geq 0.5$. However, we found that the resulting scores are highly sensitive to SAM2's segmentation granularity.

As a result, the CorLoc numbers on these pseudo labels understate the qualitative differences visible in the visualizations and can even favor overly coarse boxes. For this reason, we choose to present the Ego4D-mini experiment as a qualitative case study in the appendix, as shown in Table 15, and reserve quantitative comparisons for datasets with human-annotated boxes and masks (PASCAL VOC, DAVIS-2017, ADE20K) in the main paper.

Conclusion. Without any additional pre-training data, EgoViT exhibits more object-centric and temporally stable behavior than the DINO baseline when evaluated zero-shot on unseen egocentric environments. This provides strong qualitative evidence that EgoViT's geometry- and time-aware design substantially improves generalization beyond the original training video.