# DynaProto: Dynamic Prototypical Contrast for Temporally Consistent Object-Centric Learning

**Tianran Ouyang[1]* Peiqin Xu[1]* Xingping Dong[1]† Kaihao Zhang[2] Bo Du[1]**
[1]School of Computer Science,
National Engineering Research Center for Multimedia Software,
Institute of Artificial Intelligence,
Hubei Key Laboratory of Multimedia and Network Communication Engineering,
Wuhan University, Wuhan, China
[2]Australian National University, Canberra, Australia
{ouyangtianran,peiqin,xingpingdong,dubo}@whu.edu.cn
khzhang@gmail.com

## Abstract

Unsupervised video object-centric learning requires temporally consistent representations for downstream tasks. Existing methods mainly build temporal consistency on the assumption that adjacent frames share the same object composition. However, this local assumption is insufficient for real videos. Some objects remain semantically consistent over much longer temporal spans, while object composition may also change as objects appear or disappear. Fixed-slot recurrent Slot Attention architectures struggle to accommodate such visibility changes, limiting their ability to learn stronger temporal consistency beyond adjacent frames. We propose **Dynamic Prototypical Contrast (DynaProto)** to address these challenges. DynaProto introduces a dynamic recurrent Slot Attention architecture that determines whether each slot should be activated at each frame, thereby adapting to changing object visibility over longer temporal scopes. In addition, DynaProto introduces a prototypical contrastive objective that provides stronger clip-level temporal consistency supervision. Together, these two components enable DynaProto to learn more stable slot binding over extended temporal ranges. Extensive experiments show that DynaProto learns more temporally consistent slot representations and substantially outperforms prior methods on object discovery and object property prediction on real-world datasets, including an 11.0% mBO improvement on YouTube-VIS. The code is available at https://github.com/ouyangtianran/DynaProto.

## 1 Introduction

Object-centric learning (OCL) [1–3] seeks to represent visual scenes in terms of a set of object-level components. A common formulation uses a fixed number of latent vectors, often called slots [1], where each slot is encouraged to capture one entity or a coherent scene part. By explicitly decomposing scenes into entities, object-centric representations provide a convenient interface for downstream tasks that require structured reasoning [4, 5], prediction [6, 7], and control [8, 9]. Video-based OCL [10–13] is particularly appealing because it can exploit natural object dynamics. Objects move, deform, interact, disappear, and reappear, and these temporal cues provide supervision without annotations [12]. When slots remain temporally coherent, they can act as stable carriers of object information across frames, which is essential for challenging applications such as unsupervised

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

online tracking and structured world modeling [13]. Despite recent progress, maintaining temporal consistency remains difficult on real-world videos, where clutter, viewpoint changes, occlusions, and multi-object interactions frequently break object continuity and induce identity drift.

Recent work [13] assumes that adjacent frames largely share the same object composition, and introduces a slot contrastive loss to encourage neighboring slots to remain consistent. This highlights the value of temporal consistency regularization for video OCL. More importantly, temporal consistency in real videos often extends beyond adjacent frames. Some objects remain semantically stable over much longer temporal spans, motivating consistency modeling beyond local frame pairs. At the same time, object composition may change as objects appear or disappear, which calls for a dynamic architecture that can handle changing visibility. However, extending temporal consistency to longer ranges remains challenging. Existing recurrent Slot Attention architectures represent each frame with a fixed set of slots, making it difficult to accommodate dynamic object visibility. As the temporal scope increases, the mismatch between fixed slots and changing object composition becomes more severe and can undermine stable slot binding. As a result, existing methods struggle to exploit longer-range temporal consistency while remaining compatible with real video dynamics.

In this paper, we propose **Dynamic Prototypical Contrast (DynaProto)** to address these challenges and extend temporally consistent unsupervised video OCL from adjacent-frame supervision to the clip level. DynaProto introduces a dynamic recurrent Slot Attention architecture that activates only the slots matching the objects visible in the current frame, and applies decoding and consistency regularization only to these activated slots, thereby accommodating changing object visibility over time. To provide stronger temporal supervision beyond local frame pairs, DynaProto further introduces a prototypical contrastive objective. For each slot track within a clip, we maintain a prototype and optimize two complementary terms, proto-slot and proto-proto, which jointly encourage stable slot binding over longer temporal ranges. We evaluate DynaProto on object discovery and object property prediction across both synthetic and real-world datasets. The results show that DynaProto substantially outperforms prior methods, achieves state-of-the-art performance on real-world scenes, and yields significant gains in the zero-shot supervised setting. Our key contributions are as follows:

- We introduce a **dynamic recurrent Slot Attention architecture** that dynamically activates slots at each frame and, to the best of our knowledge, is the first video-based OCL framework to explicitly model changing object visibility through dynamic slot activation.
- We propose a **clip-level prototypical contrastive loss** that provides stronger temporal consistency supervision beyond adjacent frames.
- DynaProto substantially outperforms prior methods on object discovery and object property prediction, achieving **state-of-the-art** performance on real-world scenes, including an 11.0% mBO improvement on YouTube-VIS.
- We further study the **zero-shot supervised setting** through integration with SAM2, where DynaProto delivers significant gains over prior baselines.

## 2 Related Works

**Unsupervised Video Object-Centric Learning.** Object-centric learning (OCL) aims to decompose visual scenes into a set of object-level representations. A dominant paradigm is slot-based decomposition as exemplified by Slot Attention [14] where a set of latent slots iteratively attends to image features and a decoder reconstructs the input yielding object representations without human annotations. This paradigm has become a widely adopted backbone underpinning numerous recent studies [1–3, 15–19]. Building on this, video-based OCL [10–13, 20, 21] extends slot mechanisms across the temporal dimension typically by recurrently propagating slots from previous frames to initialize the current step. Recently, research has expanded from synthetic environments to unconstrained real-world videos by leveraging explicit temporal correspondence. For instance, SOLV [21] proposes reconstructing middle-frame representations and merging similar slots. VideoSAUR [12] utilizes temporal feature similarity to introduce a motion bias for object discovery. Furthermore, SlotContrast [13] incorporates slot contrastive learning between adjacent frames into recurrent slot attention to enhance temporal consistency. Despite this progress, object representation learning in unconstrained real-world scenes remains challenging due to the changing set of visible objects. Our method addresses this by introducing a dynamic architecture which enables slots to learn more discriminative features thereby achieving superior scene decomposition.

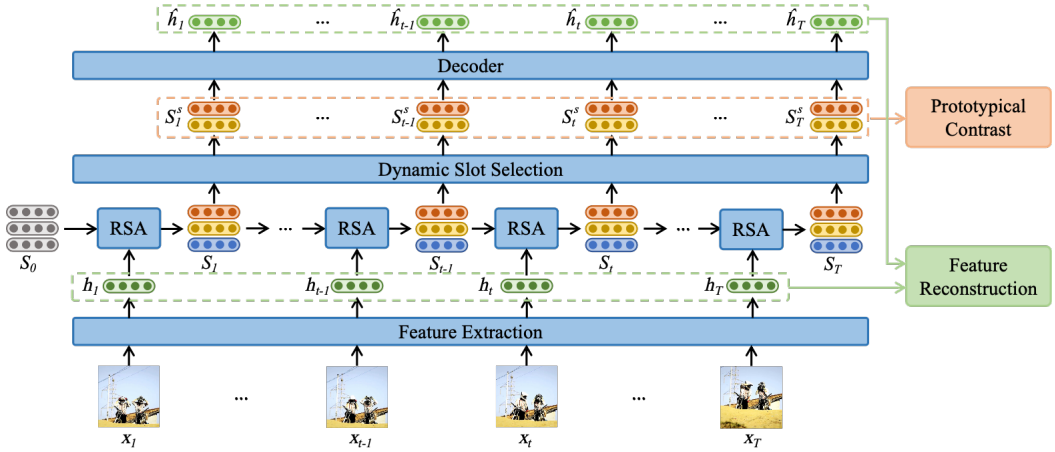



Figure 1: **Dynamic recurrent Slot Attention framework.** Video frames are initially processed by the Feature Extraction module to extract scene-level features. These features are transformed into object-level representations via the Recurrent Slot Attention (RSA) module. Next, the Dynamic Slot Selection module dynamically identifies the primary slots for each frame. Finally, the selected slots are decoded to yield reconstructed features. The framework is trained using a combined objective consisting of Feature Reconstruction and Prototypical Contrastive losses.

**Temporal Consistency in Object-Centric Learning.** Temporal consistency constitutes a fundamental objective in various video tasks. In the realm of video-based object-centric learning, several methods have been proposed to enhance this consistency. Certain approaches incorporate persistence priors to improve stability via latent state recurrence as seen in VONet [22] and GateL0RD [23]. Other strategies leverage explicit external memory to maintain historical identities employing retrieval and update mechanisms to handle occlusions and mitigate drift, e.g., OC-MOT [24] and STATM [25]. Additionally, specific works impose auxiliary consistency constraints as demonstrated by SMTC [26]. More recently, SlotContrast [13] introduces adjacent-frame slot contrast into recurrent Slot Attention to improve temporal consistency. However, real videos require consistency modeling beyond local frame pairs. Our work addresses this with clip-level prototypical contrast.

## 3 Method

Our method builds on prior reconstruction-based video object-centric approaches [10, 11, 13, 20]. First, we introduce a dynamic recurrent Slot Attention architecture that activates slots adaptively at each frame. Second, we improve the temporal consistency of slot binding through a clip-level prototypical contrastive loss. Together, these two components substantially improve temporally consistent object-centric learning. Figure 1 presents an overview of the DynaProto architecture.

### 3.1 Dynamic Recurrent Slot Attention

The proposed architecture consists of four components, namely Feature Extraction, Recurrent Slot Attention, Dynamic Slot Selection, and Dynamic Decoder. Next, we describe these in detail.

**Feature Extraction.** Given a video frame $x_t$, where $t \in \{1, 2, \ldots, T\}$, and a frozen backbone $f$, we extract $N$ patch features and adapt them for object-centric grouping with an MLP projector $g_\psi$

$$g_t = f(x_t) \in \mathbb{R}^{N \times D}, \qquad h_t = g_\psi(g_t). \tag{1}$$

**Recurrent Slot Attention.** Given the projected patch features $h_t$ for frame $t$, the recurrent state before observing this frame is the predicted slot set from the previous time step, $S^p_{t-1} = \{s^{k,p}_{t-1}\}^K_{k=1}$. The grouping module $C_\theta$ takes $h_t$ and $S^p_{t-1}$ as input and applies Slot Attention to obtain the corrected slot representations $S^c_t = \{s^{k,c}_t\}^K_{k=1}$. The predictor module $P_\omega$ then maps the corrected slots to the

predicted slot set $S_t^p = \{s_t^{k,p}\}_{k=1}^K$ used as the recurrent state for the next frame:

$$S_t^c = C_\theta(h_t, S_{t-1}^p), \qquad S_t^p = P_\omega(S_t^c). \tag{2}$$

Thus, $S_t^p$ is fed back into the grouping module at time $t+1$. We initialize the recurrence with dataset-level learned slots, i.e., $S_0^p = S_0$. Following SlotContrast [13], we use the corrected slots $S_t^c$ as the slot representations for decoding and contrastive learning, and denote them by $S_t$ hereafter.

**Dynamic Slot Selection.** For each frame $x_t, t \in \{1, 2, \ldots, T\}$, we dynamically select slots such that only the activated slots $S_t^s$ participate in feature reconstruction and prototypical contrast. Motivated by previous work [16], we leverage both slot quality and coverage to jointly select slots. Specifically, we compute slot quality based on the attention maps $A_t$, and we iteratively add the highest-quality slots into $S_t^s$ until a coverage-rate threshold is reached.

Concretely, let $A_t \in [0,1]^{N\times K}$ denote the attention map for all tokens (patches) and slots, normalized over slots (i.e., $\sum_{k=1}^K A_t^{n,k} = 1$ for each token $n$). For each token $n$, define its winner slot as

$$winner_t^n \;=\; \underset{k\in\{1,\ldots,K\}}{\arg\max}\; A_t^{n,k}. \tag{3}$$

For $s_t^k \in S_t$, we define its winner mass, total mass, and slot-quality score as

$$Winner_t^k = \sum_{n:\, winner_t^n = k} A_t^{n,k}, \qquad W_t^k = \sum_{n=1}^N A_t^{n,k}, \qquad Q_t^k = \frac{Winner_t^k}{W_t^k + \epsilon} \;\in\; [0,1]. \tag{4}$$

Here, $Q_t^k$ measures the purity of $s_t^k$'s mass within the region where it wins.

Then, we set a *coverage-quality threshold* $\sigma \in (0,1]$ and a *coverage-rate threshold* $\rho \in (0,1]$. A token $n$ is marked as *covered* if the cumulative attention mass from the selected slots $S_t^s \subseteq \{1,\ldots,K\}$ exceeds $\sigma$, and the coverage rate is computed as

$$\mathrm{Covered}_t(n) \;=\; \mathbb{1}\left[\sum_{k\in S_t^s} A_t^{n,k} \geq \sigma\right], \qquad \mathrm{Coverage}_t = \frac{1}{N}\sum_{n=1}^N \mathrm{Covered}_t(n). \tag{5}$$

We add slots into $S_t^s$ in descending order of the quality score $Q_t^k$ in Eq. 4 until $\mathrm{Coverage}_t \geq \rho$.

**Dynamic Decoder.** Following prior work [1, 2, 13], we use a spatial broadcast MLP decoder [27] applied only to the dynamically activated slots $S_t^s$, thereby eliminating the influence of inactive slots.

**Inference.** Slot Attention is typically evaluated using attention masks $A$ that indicate the association of each image patch with a specific slot [1]. These masks can be derived either from the slot attention module or from the decoder. In our method, we utilize masks from the slot attention module for evaluation. At each time step $t$, we reshape the attention map $A_t$ to the spatial token grid and upsample to the resolution of ground truth masks, obtaining $\tilde{A}_t$. Each pixel $(i,j)$ is assigned to the slot with the largest upsampled attention value, i.e., $\mathrm{winner}_t^{(i,j)} = \arg\max_k \tilde{A}_t^k(i,j)$. The mask of slot $k$ is then defined as $\mathrm{Part}_t^k = \{(i,j) \mid \mathrm{winner}_t^{(i,j)} = k\}$.

### 3.2 Clip-level Prototypical Contrast

Contrastive learning is a dominant paradigm in self-supervised representation learning, shaping the embedding space through positive and negative pairs [28–31]. SlotContrast [13] brings this idea to video OCL by enforcing slot consistency between adjacent frames. However, temporal consistency in real videos often extends beyond local frame pairs. To capture this stronger supervision, we introduce a clip-level prototypical contrastive loss with a proto-slot term and a proto-proto term, as shown in Fig. 2. Our formulation is also efficient, as analyzed in Appendix B.

**Notation.** Consider a training clip $C_v$ from video $v$, with frames indexed by $t \in \{1,\ldots,T\}$. Let $s_{v,t}^k \in \mathbb{R}^D$ denote the raw slot embedding of slot index $k \in \{1,\ldots,K\}$ at time $t$. Dynamic Slot Selection provides a set of selected slot indices $S_{v,t}^s \subseteq \{1,\ldots,K\}$ for each frame. We define the binary selection mask and normalized slot embedding as

$$m_{v,t}^k = \mathbb{1}\big[k \in S_{v,t}^s\big] \in \{0,1\}, \qquad z_{v,t}^k = \mathrm{normalize}(s_{v,t}^k). \tag{6}$$

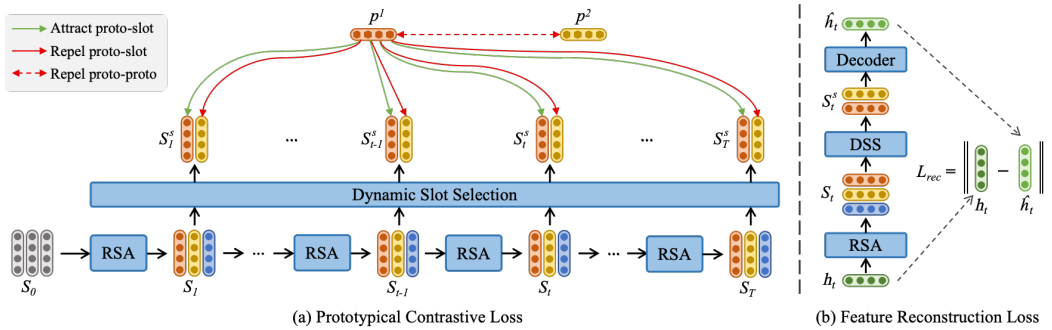



Figure 2: **Losses utilized in DynaProto.** (a) The Prototypical Contrastive Loss comprises the proto-slot and proto-proto terms. The proto-slot term attracts prototypes toward positive slots (green solid arrows) while repelling them from negative slots (red solid arrows). Simultaneously, the proto-proto term enforces separation between distinct prototypes (red bidirectional dashed arrows). (b) The Feature Reconstruction Loss minimizes the distance between original and reconstructed features.

Only slots with $m_{v,t}^k = 1$ participate in the contrastive objective.

**Proto-Slot Term.** For each slot track $(v, k)$, we collect all selected slot embeddings across the clip and compute a prototype for every active track:

$$\begin{aligned} \mathcal{Q}_v^k &= \left\{ z_{v,t}^k \,\middle|\, t \in C_v,\ m_{v,t}^k = 1 \right\}, \qquad N_v^k = |\mathcal{Q}_v^k| = \sum_{t \in C_v} m_{v,t}^k, \\ p_v^k &= \text{normalize}\left( \frac{1}{N_v^k + \epsilon} \sum_{t \in C_v} m_{v,t}^k z_{v,t}^k \right), \qquad N_v^k > 0. \end{aligned} \tag{7}$$

Here, $\mathcal{Q}_v^k$ denotes the set of selected normalized slot embeddings belonging to slot track $(v, k)$ within clip $C_v$, and $N_v^k$ is the number of selected slots in this track. A track is considered active when $N_v^k > 0$. The prototype $p_v^k$ is obtained by averaging the selected embeddings in the track, with $\epsilon$ used for numerical stability. Thus, $p_v^k$ aggregates clip-level evidence for slot track $(v, k)$.

For track $(v, k)$, positives are selected slots from the same track, while negatives are selected slots from all other tracks in the same clip:

$$\mathcal{N}_v^k = \left\{ z_{v,t}^j \,\middle|\, j \in \{1, \ldots, K\} \setminus \{k\},\ t \in C_v,\ m_{v,t}^j = 1 \right\}, \qquad \mathcal{A}_v^k = \mathcal{Q}_v^k \cup \mathcal{N}_v^k. \tag{8}$$

Here, $\mathcal{N}_v^k$ denotes the negative set for track $(v, k)$. The candidate set $\mathcal{A}_v^k$ contains both the positive set $\mathcal{Q}_v^k$ and the negative set $\mathcal{N}_v^k$. We then use a multi-positive InfoNCE objective:

$$\begin{aligned} \ell_{ps}(v, k) &= -\frac{1}{N_v^k} \sum_{x \in \mathcal{Q}_v^k} \log \frac{\exp\big(sim(p_v^k, x)/\tau_s\big)}{\sum_{a \in \mathcal{A}_v^k} \exp(sim(p_v^k, a)/\tau_s)}, \\ \mathcal{L}_{ps} &= \frac{1}{\sum_v \sum_{k=1}^K N_v^k} \sum_v \sum_{k=1}^K N_v^k \, \ell_{ps}(v, k). \end{aligned} \tag{9}$$

For each prototype, all selected slots from the same track are treated as positives and share the same denominator. This encourages the prototype to assign high probability mass to all selected observations of its own track, while suppressing selected slots from other tracks in the same clip.

**Proto-Proto Term.** We further encourage prototypes from different slot tracks within the same clip to be distinct. Let $\mathcal{P}_v$ be the set of valid prototypes and $\mathcal{P}_{v,\text{neg}}^k$ be the negative prototype set for $p_v^k$:

$$\mathcal{P}_v = \left\{ p_v^j \,\middle|\, j \in \{1, \ldots, K\},\ N_v^j > 0 \right\}, \qquad \mathcal{P}_{v,\text{neg}}^k = \left\{ p_v^j \,\middle|\, j \in \{1, \ldots, K\} \setminus \{k\},\ N_v^j > 0 \right\}. \tag{10}$$

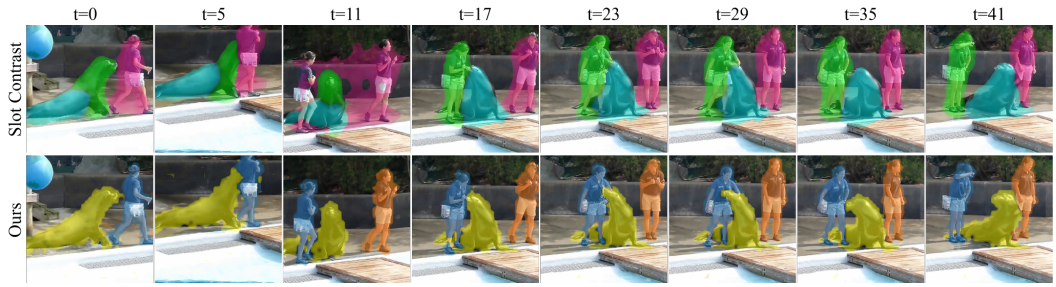



Figure 3: **Qualitative comparison with SlotContrast on the YTVIS21.** Our method produces finer object boundaries, maintains stable slot binding to the same target instance across frames, and remains robust when objects disappear or reappear. Colors are adjusted for better visual distinction.

The proto-proto loss and its average over all valid prototypes are

$$\ell_{pp}(v,k) = -\log \frac{\exp\big(sim(p_v^k, p_v^k)/\tau_p\big)}{\exp(sim(p_v^k, p_v^k)/\tau_p) + \sum\limits_{p \in \mathcal{P}_{v,\text{neg}}^k} \exp(sim(p_v^k, p)/\tau_p)},$$
$$\mathcal{L}_{pp} = \frac{1}{\sum_v \sum_{k=1}^K \mathbb{1}[N_v^k > 0]} \sum_v \sum_{k=1}^K \mathbb{1}\big[N_v^k > 0\big]\ \ell_{pp}(v,k). \tag{11}$$

**Final Loss.** Following prior work [1, 12, 13], we adopt a feature reconstruction loss for scene decomposition and combine it with the two contrastive terms:

$$\mathcal{L}_{con} = \mathcal{L}_{ps} + \alpha \mathcal{L}_{pp}, \qquad \mathcal{L} = \mathcal{L}_{rec}(h_{1:T}, \hat{h}_{1:T}) + \beta \mathcal{L}_{con}. \tag{12}$$

# 4 Experiments

First, we assess our model on both temporally consistent object discovery and per-frame object discovery. We then follow [13] to evaluate the zero-shot supervised setting and examine its integration potential. Furthermore, we evaluate object property prediction to assess the quality of the learned slot representations. Finally, we conduct ablation studies on the main components and hyperparameters. Additional visualizations and analysis are provided in Appendix F.

**Datasets.** We evaluate our method on both synthetic and real-world datasets. For synthetic benchmarks, we use MOVi-C and MOVi-E [32]. For real-world evaluation, we use the YouTube-VIS 2021 (YTVIS21) [33] following SlotContrast [13]. For object property prediction, we utilize the COCO [34] following prior studies [1].

**Metrics.** Consistent with prior methods [1, 10, 13, 20], we report the *video foreground adjusted rand index (FG-ARI)* and the *video intersection over union with mean best overlap matching (mBO)*. Both metrics are computed over the full video. Additionally, we evaluate per-frame performance denoted as *Image FG-ARI* and *Image mBO* to demonstrate single-image object discovery capabilities. For property prediction [1], we evaluate macro-averaged Top-1/Top-5 accuracy for classification and the coefficient of determination ($R^2$) for center coordinates ($R^2$-X, $R^2$-Y).

**Implementation Details.** We adopt DINOv2 [35] as the feature encoder, using ViT-S/14 for MOVi-C and ViT-B/14 for both MOVi-E and YTVIS21. The slot dimension is fixed at 64 across all datasets. Regarding input processing, we use a resolution of $336 \times 336$ for MOVi datasets and $518 \times 518$ for YTVIS21 resulting in sequence lengths of 576 and 1369 tokens respectively. We set the batch size to 64 across all datasets and train for 100,000 steps. Details are provided in Appendix A.

## 4.1 Object Discovery

**Temporally Consistent Object Discovery (Table 1 & Figure 3).** On synthetic benchmarks, DynaProto outperforms recent methods such as SlotContrast, SAVi, and STEVE. Compared with

Table 1: **Comparison of temporally consistent object discovery.** Results are from VideoSAUR [12] and SlotContrast [13]. The best values are in bold and the second-best values are underlined.

| | MOVi-C | | MOVi-E | | YTVIS21 | |
|---|---|---|---|---|---|---|
| Method | FG-ARI | mBO | FG-ARI | mBO | FG-ARI | mBO |
| SAVi [10] | 22.2 ± 2.1 | 13.6 ± 1.6 | 42.8 ± 0.9 | 16.0 ± 0.3 | – | – |
| STEVE [11] | 36.1 ± 2.3 | 26.5 ± 1.1 | 50.6 ± 1.7 | 26.6 ± 0.9 | 15.0 ± 0.7 | 19.1 ± 0.4 |
| VideoSAUR [12] | 64.8 ± 1.2 | **38.9 ± 0.6** | 73.9 ± 1.1 | **35.6 ± 0.5** | 28.9 ± 0.4 | 26.3 ± 0.2 |
| VideoSAURv2 | – | – | 77.1 | <u>34.4</u> | 31.2 | 29.7 |
| SlotContrast [13] | <u>69.3</u> | 32.7 | <u>82.9</u> | 29.2 | <u>38.0</u> | <u>33.7</u> |
| DynaProto | **73.3 ± 4.9** | <u>37.4 ± 1.0</u> | **86.4 ± 0.2** | 31.8 ± 0.1 | **45.7 ± 0.2** | **44.7 ± 0.5** |

Table 2: **Comparison of per-frame object discovery.** All baseline results are taken from their original papers and SlotContrast [13]. The best values are in bold.

| | Method | Objective | MOVi-C Image FG-ARI | MOVi-E Image FG-ARI |
|---|---|---|---|---|
| Image | DINOSAUR [1] | Image Rec. | 68.6 | 65.1 |
| | AdaSlot [15] | Image Rec. | 75.6 | 76.7 |
| | MetaSlot [17] | Image Rec. | 63.9 | – |
| Video +Motion | MoToK [36] | +Mot. Seg. | 69.3 | 66.7 |
| | Safadoust et al. [37] | +GT Flow | 73.8 | 78.3 |
| | DIOD [38] | +Mot. Seg. | – | 82.2 |
| Video | STEVE [11] | Video Rec. | 51.9 | 59.5 |
| | VideoSAUR [12] | Temp. Sim. | 75.5 ± 0.9 | 78.4 ± 0.7 |
| | SOLV [21] | Mid. Fr. Pred. | – | 80.8 |
| | SlotContrast [13] | Slot Contrast | 75.1 | 84.8 |
| | DynaProto | Proto. Contrast | **87.5 ± 0.1** | **90.6 ± 0.1** |

VideoSAUR, it achieves substantially higher FG-ARI, indicating stronger temporal identity consistency. Although VideoSAUR obtains higher mBO on MOVi, this is closely related to the characteristics of these synthetic benchmarks, where objects mostly undergo rigid motion and provide limited semantic cues due to repetitive geometry and weak textures. Its motion-based objective is therefore well matched to this simplified setting. Results on the real-world YTVIS21 dataset support this explanation. In realistic scenes, rigid-motion assumptions are less reliable, and motion cues are often confounded by non-rigid deformation, occlusion, and camera motion. Under these conditions, VideoSAUR degrades substantially, whereas DynaProto maintains strong performance by leveraging more discriminative visual evidence for consistent instance separation. Consequently, DynaProto outperforms all baselines on YTVIS21, improving FG-ARI and mBO by 7.7% and 11.0%, respectively. Overall, these results show that DynaProto achieves strong temporally consistent object discovery, with particularly clear advantages on real-world videos. They further suggest that DynaProto encourages each slot to learn stable and discriminative object representations. Qualitative results indicate that such representations can support stable rebinding after temporary disappearance caused by occlusion, with additional examples provided in Appendix F.

**Per-Frame Scene Decomposition (Table 2).** Previous studies have shown that unsupervised object discovery benefits from suitable inductive biases and training objectives, including reconstruction, motion cues, and adjacent-frame consistency. Here we show that DynaProto further improves scene decomposition. Following SlotContrast [13], we compare three categories of methods, namely image-based methods, video-based methods, and motion-augmented video methods. Image-based methods are trained on images only. Video-based methods are trained on videos only, where STEVE [11] reconstructs current-frame features, SOLV [21] predicts middle-frame features, VideoSAUR [12] predicts temporal feature similarity, and SlotContrast enforces slot consistency between adjacent frames. We further compare with weakly supervised methods that use motion masks [36, 38] or ground-truth optical flow [37]. Video-based methods generally outperform image-based methods, while additional motion supervision or temporal consistency learning further improves scene decomposition. By

Table 3: **Comparison in zero-shot supervised setting.** Baseline results are sourced from [13].

| | zero-shot | | | one-shot |
|---|---|---|---|---|
| | SlotContrast+SAM2 | Ours | Ours+SAM2 | GT+SAM2 |
| FG-ARI | 46.3 | 45.7 $\pm$ 0.2 | 50.9 $\pm$ 0.6 | 60.4 |
| mBO | 43.7 | 44.7 $\pm$ 0.5 | 56.6 $\pm$ 0.6 | 72.0 |

Table 4: **Object property prediction on COCO.** DINOSAUR results are from original paper. We evaluate official checkpoints for SPOT following DINOSAUR and for SlotContrast under our setting.

| | Class-top1 ↑ | Class-top5 ↑ | $R^2$-X ↑ | $R^2$-Y ↑ |
|---|---|---|---|---|
| DINOSAUR-MLP [1] | 60.5 | 83.2 | 59.1 | 58.7 |
| DINOSAUR-Trans [1] | 60.8 | 83.3 | 49.7 | 52.5 |
| SPOT [2] | 48.4 | 75.6 | 56.4 | 60.7 |
| SlotContrast [13] | 59.5 | 83.9 | 64.7 | 65.1 |
| DynaProto | **65.8** $\pm$ **0.6** | **88.9** $\pm$ **0.3** | **74.2** $\pm$ **0.5** | **75.1** $\pm$ **0.3** |

combining dynamic slot activation with prototypical contrast, our method achieves state-of-the-art performance and yields substantial gains in per-frame FG-ARI.

**Comparison with Supervised Methods (Table 3).** Following SlotContrast [13], we evaluate temporal consistency in a zero-shot supervised setting. Specifically, we use first-frame predictions from unsupervised OCL methods to initialize SAM2 [39]. The results show that our standalone model performs comparably to the SlotContrast-initialized SAM2 baseline. Unlike baselines constrained by first-frame prompts, our method can naturally handle newly appearing objects through dynamic slot binding. Qualitative visualizations are provided in Appendix F. Furthermore, when SAM2 initialized with ground truth is used as a supervised upper bound, the strong performance of *Ours+SAM2* shows that our method effectively narrows the gap between unsupervised and strong supervised approaches.

### 4.2 Object Property Prediction

Beyond mask-based evaluation, we further assess the quality of the learned slot representations through object property prediction to demonstrate their downstream potential [40]. Specifically, we use the model trained on YTVIS21 to extract slots from COCO without fine-tuning. For static-image evaluation, each COCO image is treated as a single-frame video and processed with only the first-frame recurrent inference procedure. Using these fixed slots, we train a lightweight MLP to predict object properties, including class and position. This setup follows prior work [1], with details provided in Appendix C. To evaluate representation quality, we compare against strong image-based methods, namely DINOSAUR [1] and SPOT [2], as well as the video-based SlotContrast [13]. Table 4 shows that our method performs best on all metrics. Surpassing both video-based and in-domain image baselines indicates that our model generalizes effectively, learns richer representations than image models, and captures temporal dynamics better than prior video methods. We attribute this to the synergy between clip-level temporal consistency and the dynamic architecture, which enables finer semantic modeling beyond simple object partitioning.

### 4.3 Analysis

**Ablation of Main Components (Table 5).** We incrementally add dynamic recurrent Slot Attention and Prototypical Contrast to a baseline. Using dynamic recurrent Slot Attention alone introduces adaptive slot number and improves image-level metrics on the relatively simple MOVi-C and MOVi-E datasets, but degrades video-level performance. On the more challenging real-world YTVIS21 dataset, training collapses entirely. These results suggest that the dynamic framework can improve scene decomposition, but is difficult to train stably over time without additional temporal regularization, especially in complex real-world scenes. Using Prototypical Contrast alone yields comparable results. While temporal consistency regularization benefits slot learning, enforcing clip-level consistency without the dynamic framework inevitably leads to slot misbinding. In contrast, combining dynamic recurrent Slot Attention with Prototypical Contrast yields substantial gains. This synergy provides

Table 5: **Ablation study on feature reconstruction (Rec.), dynamic framework (Dyna.), and prototypical contrast (Proto.).** The best values are in bold.

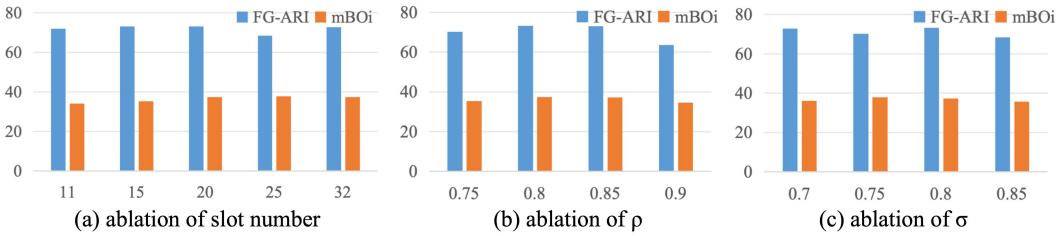
| Rec. | Dyna. | Proto. | MOVi-C FG-ARI | MOVi-C mBO | MOVi-C Image FG-ARI | MOVi-C Image mBO | MOVi-E FG-ARI | MOVi-E mBO | MOVi-E Image FG-ARI | MOVi-E Image mBO | YTVIS21 FG-ARI | YTVIS21 mBO | YTVIS21 Image FG-ARI | YTVIS21 Image mBO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | | | 55.9 ± 1.2 | 30.6 ± 0.5 | 69.8 ± 0.8 | 38.8 ± 0.6 | 74.0 ± 0.1 | 28.0 ± 0.2 | 79.0 ± 0.1 | 33.2 ± 0.4 | 37.8 ± 0.4 | 36.6 ± 0.1 | 48.3 ± 0.4 | 46.0 ± 0.4 |
| ✓ | ✓ | | 58.1 ± 4.2 | 27.1 ± 2.5 | 77.3 ± 1.3 | 42.1 ± 1.5 | 72.5 ± 3.5 | 23.1 ± 0.4 | 83.6 ± 1.4 | 35.9 ± 0.9 | 4.5 ± 4.5 | 17.4 ± 15.3 | 14.7 ± 3.0 | 22.0 ± 19.9 |
| ✓ | | ✓ | 52.9 ± 1.3 | 29.3 ± 0.4 | 69.6 ± 1.1 | 38.6 ± 0.4 | 74.5 ± 0.5 | 29.4 ± 0.4 | 79.1 ± 0.5 | 34.2 ± 0.3 | 41.2 ± 3.0 | 38.6 ± 2.8 | 51.4 ± 2.3 | 47.1 ± 1.3 |
| ✓ | ✓ | ✓ | **73.3** ± 4.9 | **37.4** ± 1.0 | **87.5** ± 0.1 | **44.8** ± 1.4 | **86.4** ± 0.2 | **31.8** ± 0.1 | **90.6** ± 0.1 | **37.3** ± 0.3 | **45.7** ± 0.2 | **44.7** ± 0.5 | **53.9** ± 0.9 | **52.0** ± 0.4 |

Table 6: **Evolution of temporal consistency with increasing contrast ranges.** We report mBSR on YTVIS21, where lower values indicate better temporal consistency.

| | DynaProto | | | | | Dyna.+SlotContrast |
|---|---|---|---|---|---|---|
| Contrast Range | 2 | 4 | 6 | 8 | 12 | 2 |
| mBSR (%) ↓ | 9.10 | 8.68 | 7.34 | 7.22 | 8.50 | 10.75 |

effective temporal regularization while mitigating slot misbinding caused by varying object presence, showing that both components are essential.

**Analysis of Contrast Range (Table 6).** We further study how the contrast range affects temporal consistency. We use the mean Binding Switch Rate (mBSR) under Best-Overlap matching as the metric, where lower values indicate fewer identity switches. Details are provided in Appendix D. Due to hardware constraints, all models in this study are trained with a reduced batch size of 48. As the contrast range increases from 2 to 8, mBSR decreases consistently, although the improvement gradually saturates. When the range is further increased to 12, temporal consistency degrades. This suggests that a larger contrast range can strengthen temporal consistency, but excessively large ranges introduce object variation beyond what the current mechanism can effectively handle. We further compare with SlotContrast at the same range of 2. Our method achieves a substantially lower mBSR than Dyna.+SlotContrast, with 9.10% versus 10.75%, indicating that prototypical contrast imposes a stronger temporal consistency constraint than pairwise slot contrast even under the same range.



(a) ablation of slot number (b) ablation of ρ (c) ablation of σ

Figure 4: **Influence of $K$, $\rho$, and $\sigma$ on MOVi-C.**

**Influence of $K$, $\rho$, and $\sigma$ (Figure 4).** As the number of slots $K$ increases from 11 to 20, mBO improves moderately and remains stable with further increases. This suggests that the slot budget mainly needs to cover the maximum number of visible objects, and a moderate surplus does not degrade performance. For both $\rho$ and $\sigma$, the model maintains stable performance over a relatively wide range, further demonstrating the robustness of our method to hyperparameter choices.

## 5 Conclusion

In this paper, we presented DynaProto for temporally consistent unsupervised video object-centric learning. Our method revisits temporal consistency from a broader temporal perspective. Rather than relying only on adjacent-frame supervision, DynaProto combines dynamic ecurrent Slot Attention with clip-level Prototypical Contrast to accommodate changing object visibility and to impose stronger temporal consistency supervision over longer ranges.

Extensive experiments on synthetic and real-world datasets validate the proposed design. DynaProto achieves strong temporally consistent object discovery and state-of-the-art performance on

real-world videos, showing that it can effectively handle changing object composition and maintain stable slot binding under realistic visibility changes, occlusions, and appearance variations. DynaProto also improves per-frame scene decomposition, which suggests that alleviating temporal slot misbinding allows slots to learn finer and more discriminative object representations. In the zero-shot supervised setting, DynaProto further narrows the gap to strong supervised approaches through integration with SAM2. Finally, its superior object property prediction results indicate that the learned object-centric representations preserve more accurate semantic information and therefore provide a stronger foundation for downstream applications.

Despite the progress brought by our method, several limitations remain. Our current design does not explicitly incorporate motion cues, the gating mechanism still operates primarily at the frame level and does not fully exploit temporal continuity, and the recurrent architecture is less effective on short videos, as discussed in Appendix E. We hope this work offers a useful foundation for future research on dynamic and temporally consistent video object-centric learning.

## Acknowledgments

This work was supported by National Natural Science Foundation of China under Grants 62225113 and 62471342, the Fundamental Research Funds for the Central Universities (No. 2042026kf0044), the Innovative Research Group Project of Hubei Province under Grants 2024AFA017, and the New Cornerstone Science Foundation through the XPLORER PRIZE. We also thank Juhaokan Technology Co., Ltd. for its support.

## References


[1] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. In *ICLR*, 2023.

[2] Ioannis Kakogeorgiou, Spyros Gidaris, Konstantinos Karantzalos, and Nikos Komodakis. Spot: Self-training with patch-order permutation for object-centric learning with autoregressive transformers. In *CVPR*, pages 22776–22786, 2024.

[3] Pinzhuo Tian, Shengjie Yang, Hang Yu, and Alex Kot. Pay attention to the foreground in object-centric learning. In *CVPR*, pages 30281–30290, 2025.

[4] Aniket Didolkar, Andrii Zadaianchuk, Rabiul Awal, Maximilian Seitzer, Efstratios Gavves, and Aishwarya Agrawal. Ctrl-o: language-controllable object-centric visual representation learning. In *CVPR*, pages 29523–29533, 2025.

[5] Jiaqi Xu, Cuiling Lan, Wenxuan Xie, Xuejin Chen, and Yan Lu. Slot-vlm: Object-event slots for video-language modeling. In *NeurIPS*, volume 37, pages 632–659, 2024.

[6] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. In *ICLR*, 2023.

[7] Sha Meng, Dian Shao, Jiacheng Guo, and Shan Gao. Tracking without label: Unsupervised multiple object tracking via contrastive similarity learning. In *ICCV*, pages 16264–16273, 2023.

[8] Dan Haramati, Tal Daniel, and Aviv Tamar. Entity-centric reinforcement learning for object manipulation from pixels. In *ICLR*, 2024.

[9] Jaesik Yoon, Yi-Fu Wu, Heechul Bae, and Sungjin Ahn. An investigation into pre-training object-centric representations for reinforcement learning. In *ICML*, pages 40147–40174, 2023.

[10] Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *ICLR*, 2022.

[11] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. In *NeurIPS*, volume 35, pages 18181–18196, 2022.

[12] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. In *NeurIPS*, volume 36, pages 61514–61545, 2023.

[13] Anna Manasyan, Maximilian Seitzer, Filip Radovic, Georg Martius, and Andrii Zadaianchuk. Temporally consistent object-centric learning by contrasting slots. In *CVPR*, pages 5401–5411, 2025.

[14] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, volume 33, pages 11525–11538, 2020.

[15] Ke Fan, Zechen Bai, Tianjun Xiao, Tong He, Max Horn, Yanwei Fu, Francesco Locatello, and Zheng Zhang. Adaptive slot attention: Object discovery with dynamic slot number. In *CVPR*, pages 23062–23071, 2024.

[16] Tianran Ouyang, Xingping Dong, Jing Zhang, Mang Ye, Kaihao Zhang, and Bo Du. Qasa: Quality-guided k-adaptive slot attention for unsupervised object-centric learning. In *ECCV*, pages 492–508. Springer, 2026.

[17] Hongjia Liu, Rongzhen Zhao, Haohan Chen, and Joni Pajarinen. Metaslot: Break through the fixed number of slots in object-centric learning. In *NeurIPS*, 2025.

[18] Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. Slotdiffusion: Object-centric generative modeling with diffusion models. volume 36, pages 50932–50958, 2023.

[19] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. In *NeurIPS*, volume 36, pages 8563–8601, 2023.

[20] Gamaleldin Elsayed, Aravindh Mahendran, Sjoerd Van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. In *NeurIPS*, volume 35, pages 28940–28954, 2022.

[21] Görkay Aydemir, Weidi Xie, and Fatma Guney. Self-supervised object-centric learning for videos. In *NeurIPS*, volume 36, pages 32879–32899, 2023.

[22] Haonan Yu and Wei Xu. Vonet: Unsupervised video object learning with parallel u-net attention and object-wise sequential vae. *arXiv preprint arXiv:2401.11110*, 2024.

[23] Christian Gumbsch, Martin V Butz, and Georg Martius. Sparsely changing latent states for prediction and planning in partially observable domains. In *NeurIPS*, volume 34, pages 17518–17531, 2021.

[24] Zixu Zhao, Jiaze Wang, Max Horn, Yizhuo Ding, Tong He, Zechen Bai, Dominik Zietlow, Carl-Johann Simon-Gabriel, Bing Shuai, Zhuowen Tu, et al. Object-centric multiple object tracking. In *CVPR*, pages 16601–16611, 2023.

[25] Jian Li, Pu Ren, Yang Liu, and Hao Sun. Reasoning-enhanced object-centric learning for videos. In *SIGKDD*, pages 659–670, 2025.

[26] Rui Qian, Shuangrui Ding, Xian Liu, and Dahua Lin. Semantics meets temporal correspondence: Self-supervised object-centric learning in videos. In *ICCV*, pages 16675–16687, 2023.

[27] Nick Watters, Loic Matthey, Chris P Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for disentangled representations in vaes. In *ICLR Workshops*, 2019.

[28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[29] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PmLR, 2020.

[30] Xingping Dong, Jianbing Shen, and Ling Shao. Rethinking clustering-based pseudo-labeling for unsupervised meta-learning. In *ECCV*, pages 169–186. Springer, 2022.

[31] Tianran Ouyang, Xingping Dong, Mang Ye, Bo Du, Ling Shao, and Jianbing Shen. Semantic-aware pseudo-labeling for unsupervised meta-learning. *IEEE TPAMI*, 47(07):5475–5488, 2025.

[32] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *CVPR*, pages 3749–3761, 2022.

[33] Yang Linjie, Fan Yuchen, Fu Yang, and Xu Ning. The 3rd large-scale video object segmentation challenge - video instance segmentation track. 2021.

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[35] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, pages 1–31, 2024.

[36] Zhipeng Bao, Pavel Tokmakov, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert. Object discovery from motion-guided tokens. In *CVPR*, pages 22972–22981, 2023.

[37] Sadra Safadoust and Fatma Güney. Multi-object discovery by low-dimensional object motion. In *ICCV*, pages 734–744, 2023.

[38] Sandra Kara, Hejer Ammar, Julien Denize, Florian Chabot, and Quoc-Cuong Pham. Diod: Self-distillation meets object discovery. In *CVPR*, pages 3975–3985, 2024.

[39] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *ICLR*, 2025.

[40] Amir Mohammad Karimi Mamaghan, Samuele Papa, Karl Henrik Johansson, Stefan Bauer, and Andrea Dittadi. Exploring the effectiveness of object-centric representations in visual question answering: Comparative insights with foundation models. In *ICLR*, 2025.

[41] Andrea Dittadi, Samuele Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. Generalization and robustness implications in object-centric learning. In *ICML*, pages 5221–5285, 2022.

# A Training Details

Hyperparameters employed for training DynaProto are detailed in Table 7. We try our best to maintain consistency with prior work [13] to ensure a fair comparison. The hyperparameter settings for the Dynamic Slot Selection module and the number of slots follow prior work [16]. Additionally, details of Object Property Prediction task are presented separately in Appendix C. The main experiments on MOVi-C, MOVi-E, and YTVIS21 used 1, 2, and 4 NVIDIA RTX 6000 Ada GPUs with 48 GB memory, respectively, and required approximately 20, 22, and 24 hours per run.

Table 7: Hyperparameters of DynaProto for Main Results on MOVi-C, MOVi-E, and YTVIS21 Datasets

| **Hyperparameter** | **MOVi-C** | **MOVi-E** | **YTVIS21** |
|---|---|---|---|
| Training Steps | 100k | 100k | 100k |
| Batch Size | 64 | 64 | 64 |
| Training Clip Size | 4 | 4 | 4 |
| Learning Rate Warmup Steps | 2500 | 2500 | 2500 |
| Optimizer | Adam | Adam | Adam |
| Peak Learning Rate | 0.0004 | 0.0008 | 0.0008 |
| Exponential Decay | 100k | 100k | 100k |
| ViT Architecture | DINOv2 Small | DINOv2 Base | DINOv2 Base |
| Initialization | FixedLearnedInit | FixedLearnedInit | FixedLearnedInit |
| Patch Size | 14 | 14 | 14 |
| Feature Dimension ($D_{\text{feat}}$) | 384 | 768 | 768 |
| Gradient Norm Clipping | 0.05 | 0.05 | 0.05 |
| Precision | FP16 (Mixed) | FP16 (Mixed) | FP16 (Mixed) |
| **Image Specifications** | | | |
| Image / Crop Size | 336 | 336 | 518 |
| Cropping Strategy | Full | Full | Rand. Center Crop |
| Augmentations | – | – | Rand. Horizontal Flip |
| Image Tokens | 576 | 576 | 1369 |
| **Recurrent Slot Attention** | | | |
| Slots | 20 | 30 | 20 |
| Iterations (first / other frames) | 3 / 2 | 3 / 2 | 3 / 2 |
| Slot Dimension ($D_{\text{slots}}$) | 64 | 64 | 64 |
| **Dynamic Slot Selection** | | | |
| Coverage-Rate Threshold ($\rho$) | 0.8 | 0.8 | 0.8 |
| Coverage-Quality Threshold ($\sigma$) | 0.8 | 0.8 | 0.7 |
| Stabilizing Constant ($\epsilon$) | 1e-6 | 1e-6 | 1e-6 |
| **Predictor** | | | |
| Type | Transformer | Transformer | Transformer |
| Layers | 1 | 1 | 1 |
| Heads | 4 | 4 | 4 |
| **Decoder** | | | |
| Type | MLP | MLP | MLP |
| **Loss Parameters** | | | |
| Temperature ($\tau$) | 0.1 | 0.1 | 0.1 |
| Proto-Proto Term Weight ($\alpha$) | 1 | 1 | 1 |
| Prototypical Contrast Weight ($\beta$) | 0.5 | 0.5 | 0.5 |

Table 8: Comparison of temporal contrastive objectives.

| Method | Temporal Scope | Contrastive Pool | Complexity |
|---|---|---|---|
| SlotContrast [13] | Adjacent frames | Cross-video batch pool | $\mathcal{O}(B^2TK^2)$ |
| **DynaProto (Ours)** | **Full clip** | **Within-video prototypes** | $\mathcal{O}(\mathbf{BTK^2})$ |

## B Complexity Analysis of Clip-Level Prototypical Contrast

Here we compare the complexity of our formulation with adjacent-frame slot contrast. We analyze the dependence on clip length $T$, slot budget $K$, and mini-batch size $B$, while treating the embedding dimension $D$ as a constant. For simplicity, we assume each frame contains at most $K$ slots. This gives an upper bound for DynaProto, since the dynamic architecture activates only a subset of slots in practice.

**Adjacent-Frame Slot Contrast.** SlotContrast [13] contrasts slots between adjacent frames. For a clip of length $T$, this gives $T-1$ frame pairs. Within each pair, slots are compared all-to-all.

With mini-batch contrast, each frame contributes $BK$ slots across all videos. Each adjacent-frame comparison therefore requires a $(BK) \times (BK)$ similarity matrix, which gives

$$\mathcal{C}_{\text{SlotContrast}} \sim \mathcal{O}\left((T-1)(BK)^2\right) = \mathcal{O}(B^2TK^2). \tag{13}$$

This design provides cross-video negatives, but its cost grows quadratically with batch size. Its supervision is also local, since positives are defined only between neighboring frames.

**Clip-Level Prototypical Contrast.** Our method extends temporal consistency from adjacent frames to the full clip. Instead of directly contrasting slots across videos, we summarize each slot track within a video into a prototype and perform contrastive learning inside the clip.

For each video, we compute at most $K$ prototypes from $T$ frames. The aggregation cost is

$$\mathcal{C}_{\text{agg}} \sim \mathcal{O}(TK). \tag{14}$$

The proto-slot term compares $K$ prototypes with all $TK$ slot embeddings in the clip, which gives

$$\mathcal{C}_{\text{proto-slot}} \sim \mathcal{O}\left(K \cdot TK\right) = \mathcal{O}(TK^2). \tag{15}$$

The proto-proto term compares the $K$ prototypes within the clip

$$\mathcal{C}_{\text{proto-proto}} \sim \mathcal{O}(K^2). \tag{16}$$

The resulting per-video complexity is

$$\mathcal{C}_{\text{ours}}^{\text{video}} \sim \mathcal{O}(TK) + \mathcal{O}(TK^2) + \mathcal{O}(K^2) = \mathcal{O}(TK^2). \tag{17}$$

Computed independently within each video, the mini-batch complexity becomes

$$\mathcal{C}_{\text{ours}} \sim \mathcal{O}(BTK^2). \tag{18}$$

**Summary.** Compared with adjacent-frame slot contrast, DynaProto extends the temporal scope from local frame pairs to the full clip while avoiding cross-video slot comparisons. Its complexity therefore grows linearly rather than quadratically with batch size

$$\mathcal{O}(BTK^2) \quad \text{vs.} \quad \mathcal{O}(B^2TK^2). \tag{19}$$

Thus, the proposed prototypical formulation provides stronger clip-level temporal supervision with more favorable scaling for longer temporal ranges.

## C Details on Object Property Prediction

We investigate the utility of our learned representations for downstream applications by predicting object properties on the COCO dataset.

**Setup.** We follow the experimental design of Dittadi et al. [41] and Dinosaur [1]. We extract slot representations using our unsupervised model (frozen) and use them to supervise a predictor for object category and center-of-mass coordinates $(x, y)$.

**Architecture.** The downstream predictor is a single-hidden-layer MLP with a hidden dimension of 256. This model operates on a per-slot basis, applying the same weights independently and in parallel to each slot representation to output a vector of property predictions.

**Loss and Matching.** We employ cross-entropy loss for discrete classification (80 COCO categories) and Mean Squared Error (MSE) for continuous coordinate regression. The total loss is the sum of these terms. To align the set of predictions with the set of ground truth objects, we utilize the Hungarian algorithm to minimize the total matching loss.

**Data and Training Details.** The training set consists of 10,000 images randomly sampled from COCO, with an additional 1,000 images reserved for validation. Input images are first resized to $224 \times 224$ for consistency with Dinosaur [1], and then rescaled to $518 \times 518$ to align with our pre-trained model. We train the predictor for 15,000 steps using the Adam optimizer with a learning rate of $1 \times 10^{-3}$ and a batch size of 64. Evaluation is performed on the standard 5,000 validation images, excluding those containing no objects. We report the macro-averaged Top-1 and Top-5 accuracy for classification, and the coefficient of determination ($R^2$) for $x$ and $y$ coordinates.

# D Binding Switch Rate

We measure temporal consistency using the *Binding Switch Rate* (BSR) under *Best-Overlap* (BO) matching. We compute BSR only over foreground ground-truth instances.

**BO matching.** For a video $v$ with $T$ frames, let $y_{t,i}$ denote the mask of foreground ground-truth instance $i$ at frame $t$, and let $\hat{y}_{t,k}$ denote the mask predicted by slot $k$. For each frame $t$ and each foreground instance $i$ that is visible at $t$, we define the BO assignment

$$a_{t,i} \;=\; \arg \max_{k \in \{1,\dots,K\}} \mathrm{IoU}(y_{t,i}, \hat{y}_{t,k})\,. \tag{20}$$

**Binding Switch Rate.** We compute BSR over *valid transitions*, namely pairs of consecutive frames where foreground instance $i$ is visible in both frames. Let

$$\mathcal{T}_v \;=\; \{(t,i) \mid t \in \{2,\dots,T\},\; y_{t-1,i} \neq \emptyset,\; y_{t,i} \neq \emptyset\}\,. \tag{21}$$

BSR is then defined as the fraction of valid transitions where the matched slot identity changes:

$$\mathrm{BSR}(v) = \frac{1}{|\mathcal{T}_v|} \sum_{(t,i)\in\mathcal{T}_v} \mathbb{1}[a_{t,i} \neq a_{t-1,i}]\,, \tag{22}$$

and we set $\mathrm{BSR}(v) = 0$ when $|\mathcal{T}_v| = 0$.

**Dataset-level average.** We report the mean BSR over all test videos $\mathcal{V}$:

$$\mathrm{mBSR} = \frac{1}{|\mathcal{V}|} \sum_{v\in\mathcal{V}} \mathrm{BSR}(v). \tag{23}$$

# E Limitation and Future Work

Table 9: **Temporal Consistency over Different Video Lengths.** We report mBSR on YTVIS21, where lower values indicate better temporal consistency.

| | | DynaProto | | | | |
|---|---|---|---|---|---|---|
| Contrast Clip | | 2 | 4 | 6 | 8 | 12 |
| mBSR (%) ↓ | all | 9.10 | 8.68 | 7.34 | 7.22 | 8.50 |
| | short | 11.04 | 10.79 | 7.45 | 7.58 | 8.86 |
| | long | 7.46 | 6.85 | 6.85 | 6.52 | 7.93 |

By addressing dynamic object variation and enforcing consistency over broader temporal ranges, our method substantially advances temporally consistent OCL. Nevertheless, several limitations remain. First, our method primarily relies on semantic cues and does not explicitly exploit motion information. As a result, on synthetic datasets with strong motion cues, it underperforms VideoSAUR [12] on mBO. A promising direction is to better integrate semantic information with motion cues. Second, the current gating mechanism operates independently at each frame and does not fully exploit temporal information. How to design a gating mechanism that simultaneously accommodates dynamic object variation and temporal continuity remains an open question. Third, we observe that the current recurrent architecture performs worse on shorter videos with fewer than 30 frames, while performing better on longer videos with more than 40 frames, as shown in Table 9. We conjecture that this behavior arises from the recurrent processing scheme, under which slot representations gradually improve over time, giving longer videos an advantage. This limitation may be alleviated by bidirectional recurrence or alternative non-recurrent architectures.

# F Additional Examples

In this section, we first highlight our robustness to severe occlusions in Fig. 5. Next, supplementary comparisons on YTVIS21 (Fig. 6, Fig. 7), MOVi-C (Fig. 8), and MOVi-E (Fig. 9) demonstrate that our method excels in boundary delineation, segmentation accuracy, and temporal consistency, validating the quality of our learned slot representations. Finally, comparisons with OCL-initialized SAM2 (Fig. 10) confirm that our approach successfully handles challenging scenarios where prior arts struggle, even with SAM2 enhancement.

To provide a more intuitive visualization of our method's superiority, we include **GIF videos** for the YTVIS21 examples in the supplementary material.

For optimal visual clarity, only color assignments were adjusted where necessary.

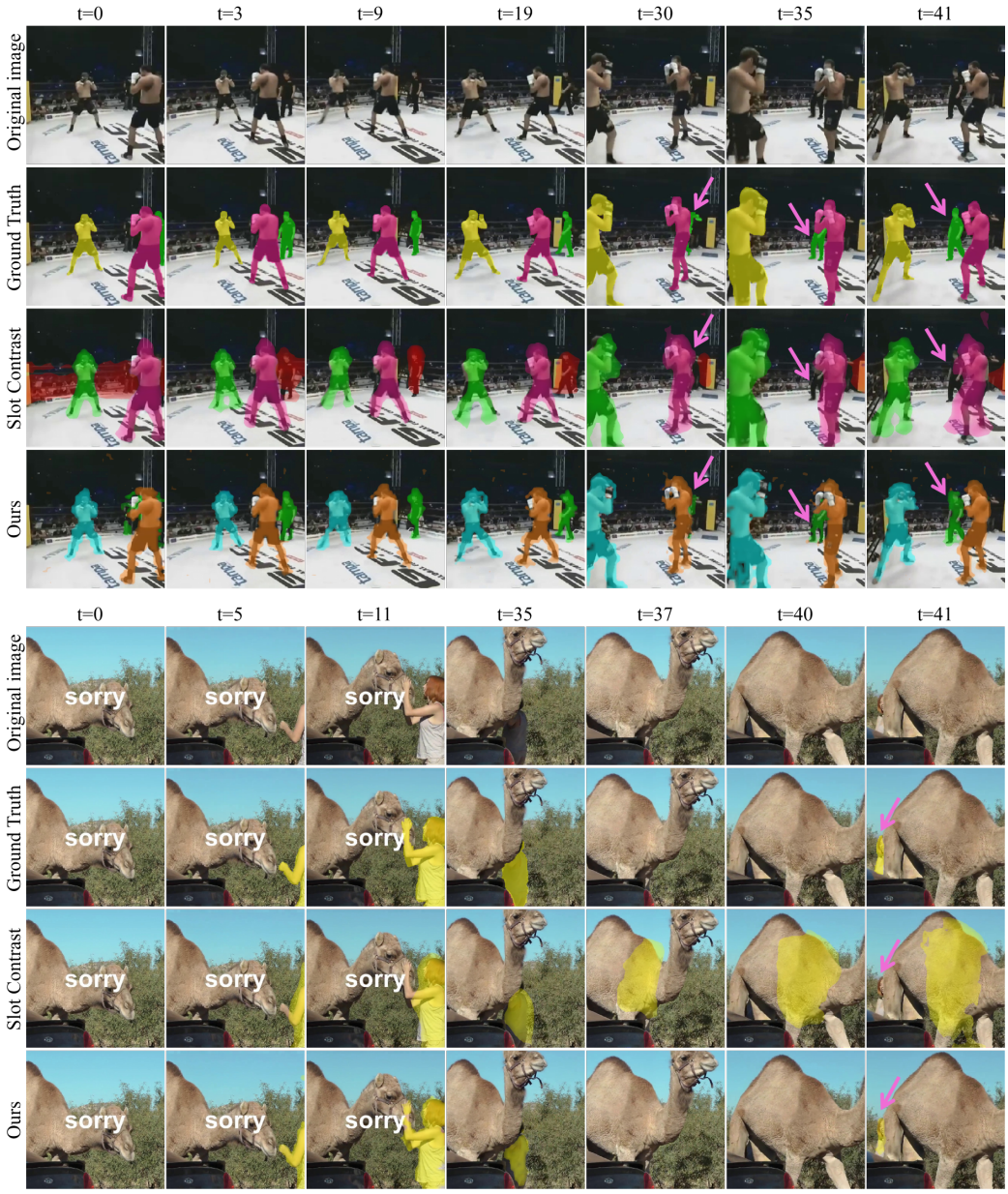



Figure 5: **Examples of object reappearance.** Arrows indicate the reappearing object, demonstrating our model's superior temporal consistency. It successfully maintains the same slot binding even after near-total occlusion.

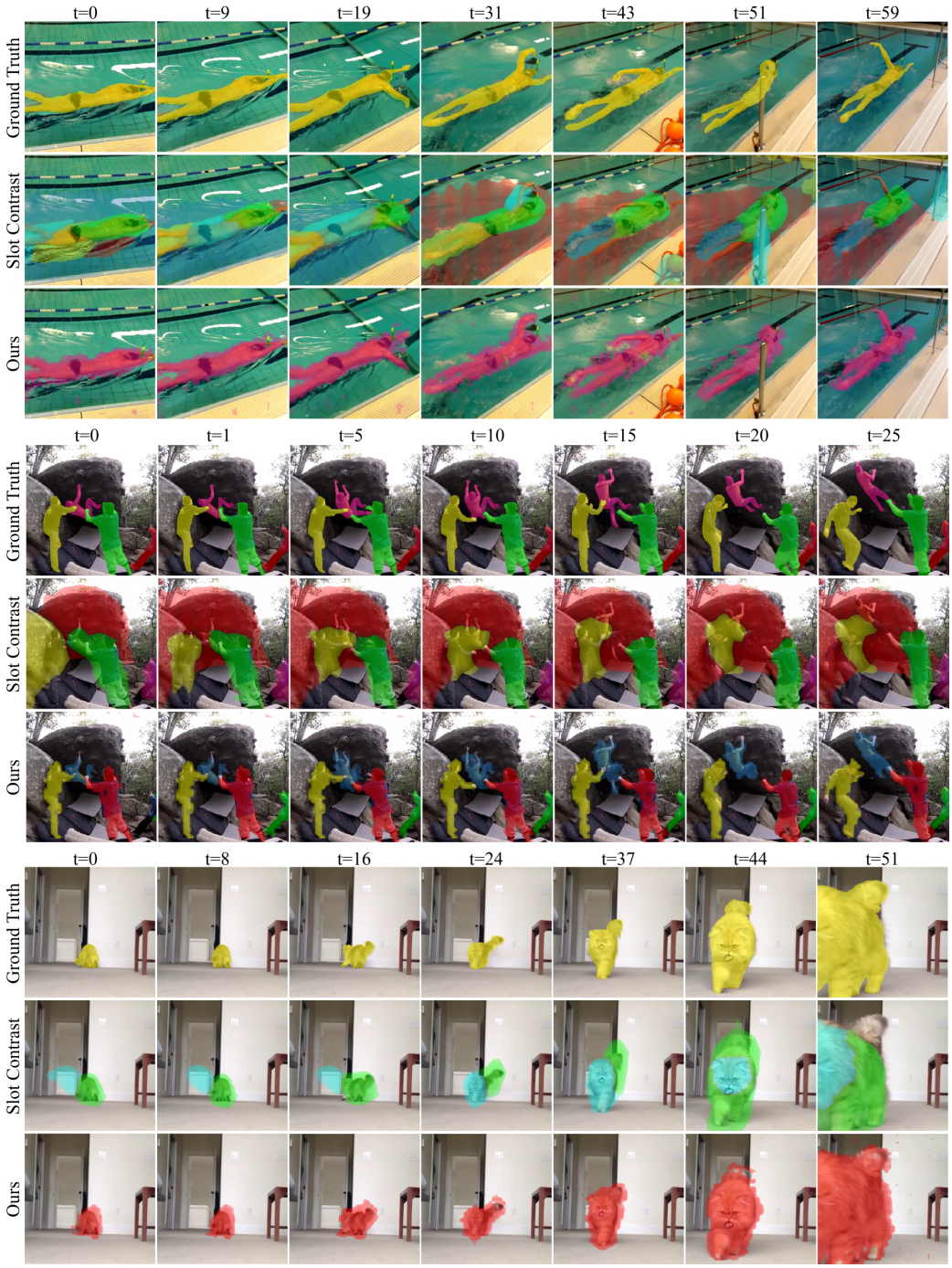



Figure 6: **Additional qualitative comparisons of temporally consistent object discovery on YTVIS21.**

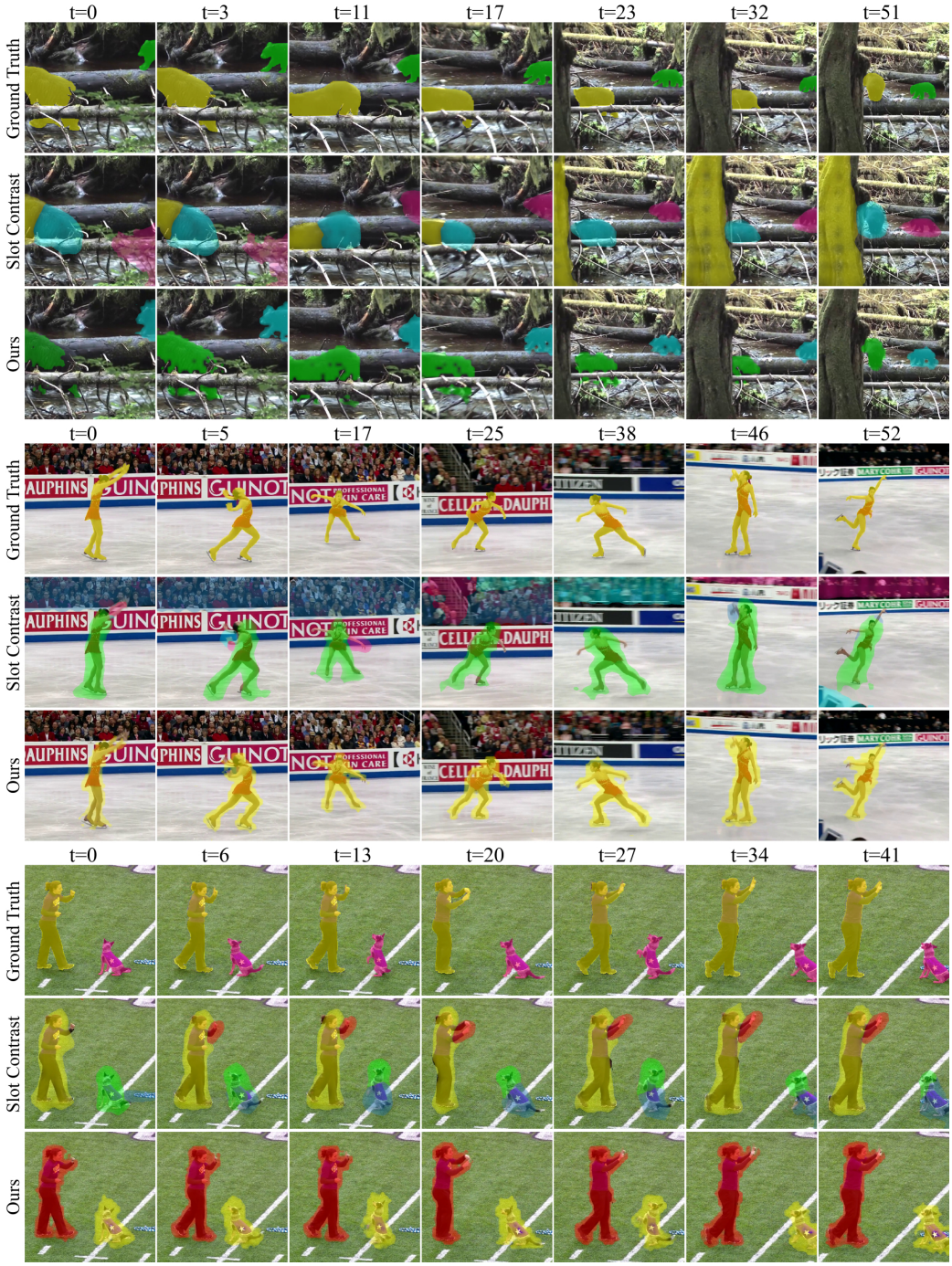



Figure 7: **Additional qualitative comparisons of temporally consistent object discovery on YTVIS21.**




Figure 8: **Additional qualitative comparisons of temporally consistent object discovery on MOVi-C.**

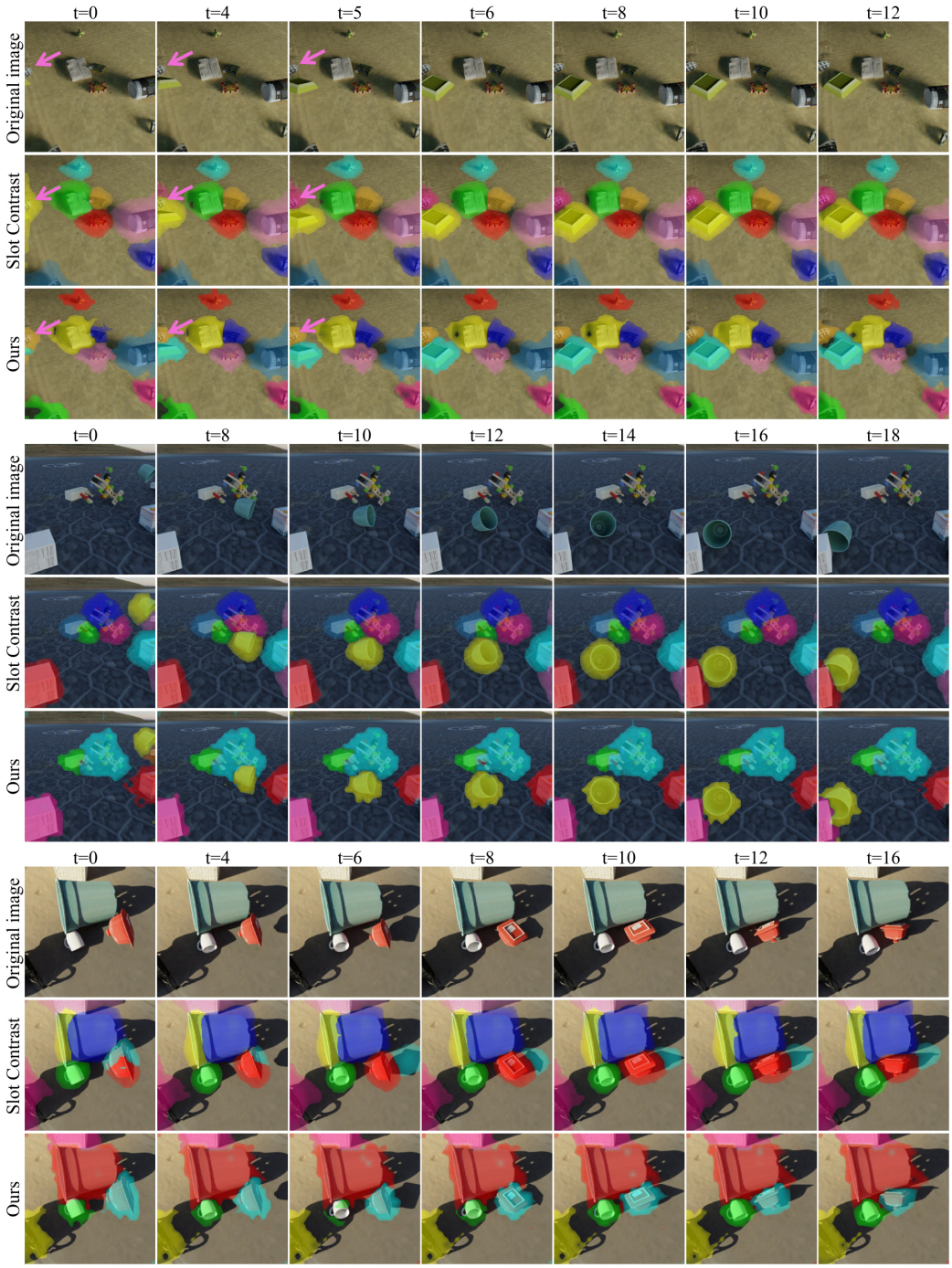



Figure 9: **Additional qualitative comparisons of temporally consistent object discovery on MOVi-E.**




Figure 10: **Additional qualitative comparisons between our method and SAM2-enhanced SlotContrast on YTVIS21.** Visualizations reveal that the baseline struggles with missing small instances, merging distinct objects, and failing to capture object emergence. In contrast, our method effectively addresses these limitations, demonstrating superior granularity and robustness.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All statements in the abstract and introduction accurately reflect the scope and contributions of this paper. Specifically, we propose DynaProto, which addresses the limitations of existing methods.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide additional discussion of the limitations of our method in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The innovation of our work lies at the practical level, guided by intuition. More rigorous theoretical justification remains an open direction for future research.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide experimental details in the paper and the Appendix to ensure the reproducibility of our work.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We use publicly available datasets. Our code and instructions for reproducing the main experimental results, including environment setup, data preparation, training, and evaluation, are publicly available at `https://github.com/ouyangtianran/DynaProto`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We provide the key experimental setups in the paper and the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Main results are reported as mean $\pm$ standard deviation over three random seeds, where the standard deviation is computed across independent runs with different random initializations.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report the compute resources for the main experiments in the appendix, including GPU type, memory, number of GPUs, and approximate training time.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conducted in the paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper focuses on upstream learning itself rather than specific downstream tasks, and therefore does not involve direct potential positive or negative societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: We properly cite the original papers and sources of the third-party datasets, models, and code used in this work. However, we do not explicitly list the corresponding licenses and terms of use in the manuscript.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper is a method paper and does not release new assets at submission time. We do not introduce a new dataset, and the implementation code will be released after acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.