CoSlot: Clip-Level Joint Slot Binding for Video Object-Centric Learning

Anonymous authors
Paper under double-blind review

Abstract

Video object discovery requires information from multiple frames while maintaining object identities across frames. In commonly used training setups with per-frame reconstruction, adding context frames also increases the cost of dense decoding. We propose CoSlot, which jointly binds features from multiple frames with a set of shared slots and combines two reconstruction objectives: shared prototypes reconstruct all visible frames using the binding assignments, while a spatial decoder reconstructs sampled frames from frame-level slot features. This design allows the number of frames used for binding to grow while keeping the number of spatially decoded frames fixed. Experiments show that combining the two reconstruction objectives improves object grouping in videos. In inference comparisons using fixed models, shorter independent chunks mainly reduce cross-frame consistency, and slot passing recovers most of the loss. Without a slot contrastive loss or explicit slot regularization terms, CoSlot achieves video FG-ARI scores of 72.2 and 81.9 on MOVi-C and MOVi-E, respectively. In the evaluated configurations, its cost per optimizer update on precomputed features is lower than that of SlotContrast and SRL. Experiments on YouTube-VIS also show the benefit of passing slots across chunks.

1 Introduction

Video object discovery requires both distinguishing objects within each frame and maintaining their identities across frames despite motion, occlusion, and appearance changes. Earlier scene-decomposition models introduced object-like latent representations (Eslami et al., 2016; Burgess et al., 2019; Greff et al., 2019; Engelcke et al., 2020). The binding problem concerns how distributed information is grouped into distinct entities (Greff et al., 2020); Slot Attention provides a competitive feature-to-slot mechanism (Locatello et al., 2020), which we extend to video clips.

Recurrent video models update slots frame by frame (Kipf et al., 2022; Elsayed et al., 2022; Manasyan et al., 2025). In their commonly used per-frame reconstruction configurations, longer training clips require more state updates and dense frame reconstructions, increasing decoding cost. A few earlier methods have shown that a single set of object representations can also process information from multiple frames jointly (Kabra et al., 2021; Fan et al., 2023). Yet, for Slot Attention-based methods, how multi-frame context affects object grouping in training and inference remains insufficiently studied. How to increase the number of jointly bound frames without a corresponding increase in dense decoding cost also remains underexplored.

We propose CoSlot, which iteratively binds frozen visual features with clip-level shared slots. Each slot generates a feature prototype shared across frames; the binding assignments and these prototypes reconstruct every visible frame. A spatial decoder reconstructs sampled frames from their frame-level slot features. The shared prototype constrains features assigned to the same slot across frames, while spatial reconstruction provides additional within-frame structural constraints. Together, the two losses train the binder with more frames while dense spatial decoding remains limited to a small sample.

We evaluate this design through training ablations and inference comparisons using fixed checkpoints. On MOVi-C, shared-prototype and spatial reconstruction improve object grouping in videos when combined. Shortening Reset chunks leaves image-level grouping largely stable but markedly reduces video-level consistency; GS passing recovers most of the loss. A sweep over training clip lengths

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

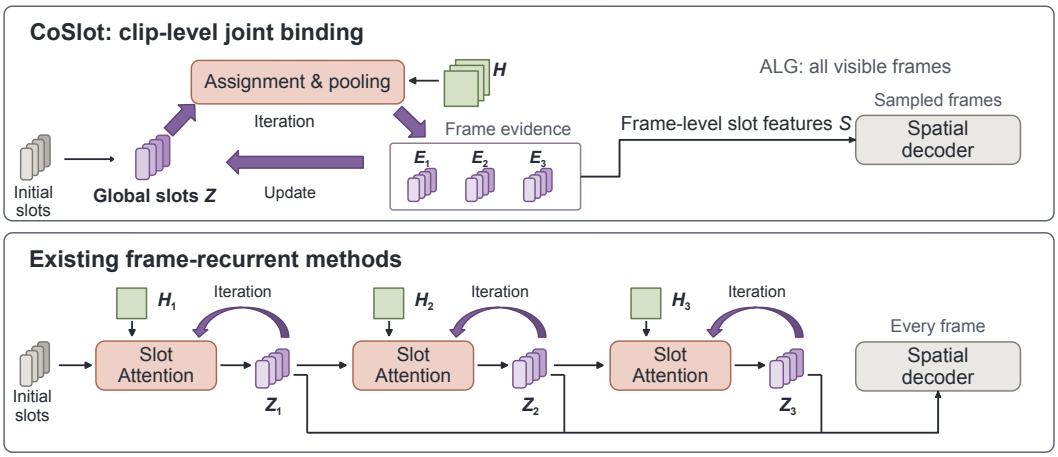

Figure 1: Temporal context and spatial decoding coverage. **Top:** CoSlot iteratively assigns the frozen features H of all visible frames to global slots Z, pools them per frame into frame evidence E_t, and updates Z; the final assignments give frame-level slot features S. ALG covers all visible frames, while spatial decoding covers only sampled frames during training. **Bottom:** frame-recurrent baselines run Slot Attention frame by frame, pass the slots Z_t forward, and spatially decode every frame in the training configurations considered.

further shows that the benefits of longer training context can appear in video FG-ARI even under single-frame Reset.

Our contributions are:

1. **Joint binding and reconstruction.** We combine shared-prototype reconstruction over all visible frames with spatial reconstruction of sampled frames, decoupling the number of frames used for binding from the number of densely decoded frames.
2. **Training and inference context.** Fixed-checkpoint comparisons distinguish the effects of the number of bound frames, temporal coverage, and GS passing. We also observe improved video consistency under single-frame Reset with longer training context.
3. **Grouping quality and training cost.** Experiments support binding longer clips while retaining one spatially reconstructed frame per clip. CoSlot achieves competitive video FG-ARI on MOVi-C/E, with lower memory and time per update than SlotContrast and SRL (Manasyan et al., 2025; Seong et al., 2026) on precomputed features in the evaluated MOVi-C and YouTube-VIS configurations (Table D.3).

2 Related Work

2.1 Object-Centric Scene Decomposition

Early generative models decomposed scenes into object-like latent representations (e.g., Eslami et al., 2016; Burgess et al., 2019; Greff et al., 2019; Engelcke et al., 2020; Lin et al., 2020). Slot Attention introduced iterative competitive feature-to-slot binding (Locatello et al., 2020), and later work extends it for object discovery in images (e.g., Biza et al., 2023; Jia et al., 2023; Kim et al., 2023; Fan et al., 2024).

2.2 Video Object-Centric Learning and Temporal Consistency

Early sequential generative models followed moving objects through time-indexed latent states (e.g., Kosiorek et al., 2018; Jiang et al., 2020). Recurrent slot-based models such as SAVi and STEVE carry object states from frame to frame (Kipf et al., 2022; Elsayed et al., 2022; Singh et al., 2022). DINOSAUR showed that reconstructing features from a frozen self-supervised backbone supports object discovery in complex natural images (Seitzer et al., 2023). VideoSAUR extended feature

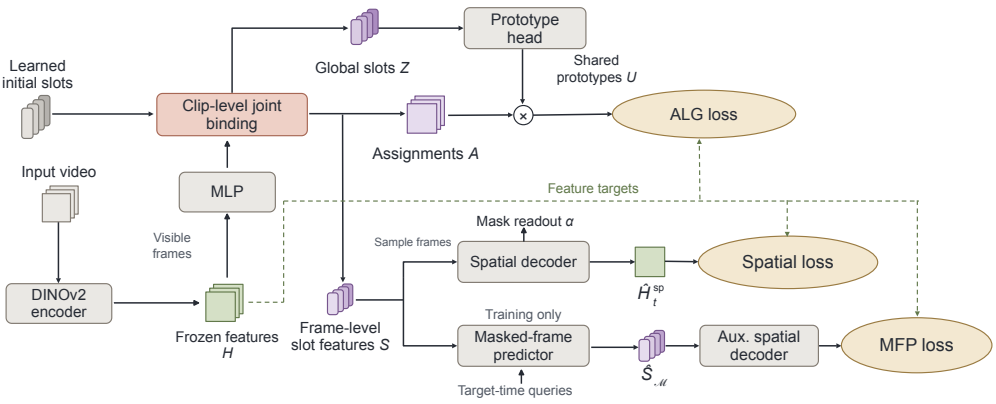

Figure 2: CoSlot overview. Joint binding outputs global slots Z, assignments A, and frame-level slot features S, which feed ALG and sampled-frame spatial reconstruction (Section 3.2) and the training-only MFP branch (Section 3.3). Dashed lines mark the frozen features H, the target of all three losses.

reconstruction to recurrent video models and used the similarity of patch features across frames as a temporal cue (Zadaianchuk et al., 2023). SlotContrast adds a contrastive objective between slots in adjacent frames to keep object representations consistent over time (Manasyan et al., 2025), and SRL lets encoder attention maps and decoder masks refine each other (Seong et al., 2026). SlotContrast combines recurrence with per-frame dense reconstruction; CoSlot jointly binds visible frames and spatially reconstructs only sampled frames. SlotContrast and SRL also add objectives whose targets come from the model's own slots or spatial maps, whereas all three CoSlot losses (ALG, Spatial, and MFP) reconstruct frozen backbone features.

Other work uses motion cues or semantic temporal correspondence, or models object dynamics over slots (e.g., Bao et al., 2023; Qian et al., 2023; Wu et al., 2023). Recent methods revise training signals and slot allocation (Moon et al., 2026; Moon & Heo, 2026), slot initialization, state transitions, and cross-frame matching (Zhao et al., 2025a;b; Li et al., 2026; Nguyen et al., 2026), or the structure of slot states (Tran et al., 2026; Zhao et al., 2026). Other designs bind slots within each frame and then relate them across frames (Aydemir et al., 2023), or process slot sequences with parallelized spatiotemporal binding or slot-structured state-space models (Singh et al., 2024; Jiang et al., 2024).

2.3 Joint Inference across Frames and Shared Object States

SIMONe addresses video generation and the disentanglement of generative factors. It represents object content and time-varying factors such as viewpoint with sequence-level object latents and frame-level global latents, respectively; it generates pixels using spatiotemporal coordinates and samples frames and pixels during training (Kabra et al., 2021). U-OVOL (Fan et al., 2023) addresses open-vocabulary object localization in videos. It applies shared Slot Attention to spatiotemporal features from a video backbone, reconstructs features through spatiotemporal broadcast decoding, and uses CLIP semantics for slot naming and merging. These methods target factorized generative representations and semantically guided video localization, respectively. CoSlot focuses on cross-frame instance grouping with frozen visual features. CoSlot jointly binds slots over frozen features extracted frame by frame, reconstructs all visible tokens with shared prototypes, and spatially decodes only sampled frames from frame-level slot features, without temporal coordinates. Relative to the methods above, we focus on how the two reconstruction objectives work together and how the number of bound frames, temporal span, and GS passing affect instance grouping.

3 METHOD

3.1 CLIP-LEVEL JOINT BINDING

Let $H = \{h_{t,p} \in \mathbb{R}^D\}$ denote frozen dense features of a video clip, where t indexes frames and p indexes patch tokens, with P tokens per frame. An MLP projects the frozen features to the slot dimension before binding; the original features remain the reconstruction targets. Let $\mathcal{V}$ be the set of frames visible to the current binding operation. CoSlot binds all visible tokens with a shared set of K slots:

$$Z, A, S = \mathrm{CoSlotBind}_\theta(H_\mathcal{V}; Z^0). \tag{1}$$

Here $Z = \{z_k\}_{k=1}^K$ comprises clip-level shared slots, referred to as global slots (GS); $A = \{a_{t,k,p}\}$ contains token-to-slot assignment weights; and $S = \{s_{t,k}\}$ comprises frame-level slot features (FS), pooled using A and supplied to the spatial decoder. The initial slots Z^0 are learnable parameters or can be taken from the global slots of the preceding chunk.

Joint binding follows the iterative competition of Slot Attention. At each iteration, the current slots determine assignments for all visible tokens, and features are pooled within each frame using those assignments. Within-frame attention across slots (SlotCoordination, SC) and aggregation of evidence across frames (TemporalEvidence, TE) then update the shared slots. The updated slots determine the assignments at the next iteration, repeatedly refining assignments over all visible frames. After the final iteration, features are pooled within each frame using the final assignments to obtain S. The binder does not use temporal positional encoding; the MFP predictor uses temporal encodings to distinguish masked target times, as described in Section 3.3. Appendix A.1 gives the full computation.

3.2 SHARED-PROTOTYPE AND SPATIAL RECONSTRUCTION

CoSlot trains the joint binder with two feature reconstruction losses: shared-prototype reconstruction covers all visible frames, whereas spatial reconstruction covers only sampled frames.

Shared-prototype reconstruction (ALG). Flatten the visible features and assignment weights into $X \in \mathbb{R}^{N\times D}$ and $A \in \mathbb{R}^{N\times K}$, where $N = |\mathcal{V}|P$. Each global slot generates a position-independent feature prototype, and the assignment weights directly reconstruct the input:

$$u_k = d_{\mathrm{alg}}(z_k), \qquad U = [u_1; \ldots; u_K], \qquad \widehat{X} = AU, \tag{2}$$

$$\mathcal{L}_{\mathrm{alg}} = \frac{1}{N}\|X - AU\|_F^2. \tag{3}$$

The prototypes U are shared across all visible frames. Features from multiple frames assigned to the same slot thus jointly update its prototype and their assignments, constraining the prototype from varying independently across frames. This loss can be computed from compact matrix statistics without explicitly producing per-slot, per-position feature outputs for every frame (Appendix A.1). On YouTube-VIS, the direct gradient from ALG through the output assignments A is multiplied by 0.01, while the gradient through the prototype path is unchanged. MOVi uses the standard gradient; Appendix A.2 specifies the rule.

Spatial reconstruction of sampled frames (Spatial). Sample a set of frames $\mathcal{S}$ uniformly from $\mathcal{V}$, and feed their frame-level slot features to a spatial broadcast MLP decoder:

$$(\widehat{H}_t^{\mathrm{sp}}, \alpha_t) = \mathrm{SpatialReadout}_\phi(S_t), \qquad t \in \mathcal{S}. \tag{4}$$

For each S_t, the spatial decoder predicts per-slot features and decoder masks α_t, which it mixes to reconstruct the frame features $\widehat{H}_t^{\mathrm{sp}}$. The spatial loss $\mathcal{L}_{\mathrm{sp}}$ is the mean feature reconstruction error over uniformly sampled frames, an unbiased estimate of the mean spatial loss over all visible frames. For a fixed binding context and model parameters, changing the number of sampled frames changes the computation and the sampling estimate, but not the mean objective being estimated. Standard object discovery evaluation uses α; Appendix A.2 provides the full definition. The number of slot–position pairs decoded by the visible-frame spatial branch per update falls from $K|\mathcal{V}|P$ to $K|\mathcal{S}|P$; the remaining visible frames still receive reconstruction supervision through ALG. Spatial decoding uses the corresponding frame-level slot features and spatial positional encodings, without temporal coordinates. At inference, spatial decoding is applied to frames for which masks are required.

3.3 Masked-Frame Prediction and Training Objective

During training, we remove a set of target frames $\mathcal{M}$ from the binding input and reconstruct their features through masked-frame prediction (MFP). The predictor uses visible-frame slot features $S_{\mathcal{V}}$ and their temporal encodings as context. Queries $Q_{\mathcal{M}}$ with target-time encodings predict target-frame slot features, which an independent auxiliary spatial decoder uses to reconstruct the target features. Prediction may draw on visible frames before and after a target frame (He et al., 2022; Tong et al., 2022).

$$\widehat{S}_{\mathcal{M}} = \mathrm{Predict}_{\psi}(S_{\mathcal{V}}, Q_{\mathcal{M}}), \qquad (\widehat{H}_t^{\mathrm{mask}}, \alpha_t^{\mathrm{mask}}) = \mathrm{SpatialReadout}_{\phi_{\mathrm{mp}}}(\widehat{S}_t). \tag{5}$$

The auxiliary decoder has the same type of architecture as the spatial decoder in Section 3.2 but separate parameters. Prediction error yields $\mathcal{L}_{\mathrm{mask}}$; Appendix A.2 describes target-frame sampling for decoding and auxiliary loss scaling. The full training objective is

$$\mathcal{L} = \lambda_{\mathrm{alg}}\mathcal{L}_{\mathrm{alg}} + \lambda_{\mathrm{sp}}\mathcal{L}_{\mathrm{sp}} + \lambda_{\mathrm{mask}}\mathcal{L}_{\mathrm{mask}}. \tag{6}$$

All three losses are computed in the frozen feature space, over all visible tokens, sampled visible frames, and sampled target frames, respectively. MFP is used only during training; inference runs joint binding and spatial decoding.

3.4 Chunked Inference and Global Slot Passing

At inference, input frames are no longer masked. The same checkpoint can either perform joint binding over the full video in one offline operation (Whole, with access to both past and future frames) or partition the evaluation frames into ordered chunks $\mathcal{C}_1, \ldots, \mathcal{C}_J$. A chunk may comprise contiguous or interleaved sampled frames. Binding for chunk j is

$$Z_j, A_j, S_j = \mathrm{CoSlotBind}_{\theta}(H_{\mathcal{C}_j}; Z_j^0). \tag{7}$$

With independent chunk inference (Reset), every chunk starts from the default state Z_{init}. With global slot passing (GS passing), the global slots from one chunk initialize the next: $Z_1^0 = Z_{\mathrm{init}}$ and $Z_{j+1}^0 = Z_j$.

Contiguous and interleaved chunking change which frames are included in a binding operation, whereas slot passing initializes each subsequent chunk from the preceding chunk's state. Video-level metrics are computed after restoring the original frame order. Let T_{train} denote training clip length, $L_{\mathrm{bind}} = |\mathcal{C}_j|$ the number of frames in one binding operation, and T_{eval} evaluation video length. For a fixed checkpoint, L_{bind} and the frames it includes can be chosen independently of T_{train}. When $L_{\mathrm{bind}} = 1$, Reset processes each frame independently, whereas GS passing uses the preceding slot state.

4 Experiments

4.1 Experimental Setup

Datasets and metrics. We evaluate on the synthetic video datasets MOVi-C and MOVi-E (Greff et al., 2022) and the real-world YouTube-VIS 2021 dataset (Yang et al., 2019). MOVi-C contains moving objects with realistic textures; MOVi-E additionally introduces camera motion and more crowded scenes. We follow SlotContrast for dataset splits and metric computation (Appendix B.2). We report foreground adjusted Rand index (FG-ARI) and mean best overlap (mBO), both as percentages. Video-level metrics require an object to retain the same slot across frames. Image-level metrics are computed per frame and then averaged, so they are unaffected by permutations of slot indices across frames. Except for the binding attention A columns explicitly marked in Table 1, all performance metrics in the main text use decoder masks α.

Implementation and baselines. CoSlot uses precomputed features from a frozen DINOv2 backbone (Oquab et al., 2024). The backbone, input resolution, and slot configuration match SlotContrast and SRL. We train on 24-frame clips for MOVi and 4-frame clips for YouTube-VIS, sampling one

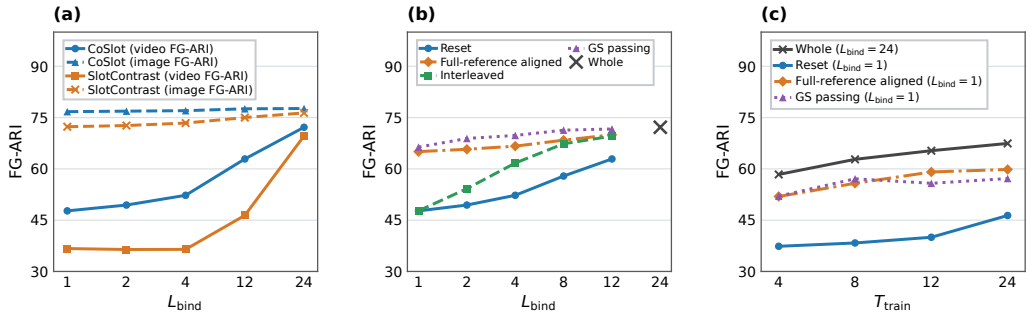

Figure 3: Temporal-context diagnostics on MOVi-C. (a) Image-level and video-level FG-ARI under independent contiguous chunks. SlotContrast uses its released checkpoint; CoSlot reports three-seed results. Chunk length L equals L_{bind} for CoSlot. (b) Comparison of binding scope, temporal coverage, and GS passing with each CoSlot checkpoint fixed; Full-reference aligned uses each checkpoint's own Whole predictions to help reindex slots. CoSlot curves in (a) and (b) show three-seed means. (c) Training-clip-length sweep with a fixed spatial decoding budget and MFP disabled (one fixed random seed); resource measurements follow Appendix B.3.1.

visible frame per clip for spatial reconstruction. The main CoSlot results and inference-protocol comparisons report the mean $\pm$ sample standard deviation over three training seeds. The training-length, component, and budget ablations in the main text use the same fixed random seed. Appendix B.1 gives the full configuration. The main table uses the baseline set from SlotContrast and adds SRL. Our analysis primarily compares with SlotContrast, which updates slots recurrently from frame to frame and relies mainly on an explicit slot contrastive loss for temporal consistency. CoSlot instead uses joint binding and shared-prototype reconstruction. SRL retains the SlotContrast losses and adds slot regularization and mutual refinement between encoder and decoder; its additional constraints primarily target mask quality and slot redundancy. We include it as a stronger reference. Where corresponding checkpoints are available, we include both methods in qualitative and training-cost comparisons.

4.2 Training Context and Inference Protocols

All results in Figure 3 are evaluated on complete 24-frame MOVi-C videos. Appendix C examines the interaction between components and single-frame inference and provides the corresponding real-video experiment.

Within-frame discovery and independent-chunk behavior. CoSlot's mean video FG-ARI increases monotonically across the tested chunk lengths, whereas image FG-ARI changes little. SlotContrast also loses substantially more video than image FG-ARI with shorter chunks (Figure 3(a)). This divergence indicates that shorter independent chunks mainly disrupt cross-frame consistency while largely preserving within-frame grouping. CoSlot's mean video FG-ARI exceeds the released SlotContrast result at every tested length. Single-frame Reset uses neither multi-frame binding nor historical state. Both methods use learnable slot initialization, but it serves different roles in training. SlotContrast uses it to start binding in the first frame, with losses from later frames backpropagating through the recurrent chain. CoSlot uses it to start joint binding over the entire visible clip, under the combined constraints of cross-frame ALG and sampled-frame spatial reconstruction. This distinction offers a possible explanation for the more stable slot assignments under single-frame Reset.

State passing and number of bound frames. GS passing raises mean video FG-ARI over Reset at every tested chunk length, recovering much of the loss even with single-frame binding and approaching Whole with longer chunks (Figure 3(b)). At a fixed multi-frame binding size, Interleaved also outperforms contiguous Reset, showing that temporal coverage matters in addition to frame count. GS passing also exceeds Full-reference aligned in mean video FG-ARI at every tested length. Paired over matching seeds, single-frame GS passing improves video FG-ARI over single-frame Reset by 18.6 ± 3.7 percentage points, and four-frame Interleaved improves over four-frame Reset by 9.4 ± 2.4 points.

Table 1: Ablation of training constraints and slot-update components on MOVi-C. Higher is better for all quality metrics. All results use one fixed training seed. Dashes indicate no trained spatial decoding head for Only-ALG.

Variant	α decoder masks				A binding masks	
	Video FG-ARI	Image FG-ARI	Video mBO	Image mBO	Video FG-ARI	Video mBO
Full	71.8	77.6	31.3	35.4	73.1	37.2
w/o MFP	67.4	74.3	30.4	34.9	68.7	35.9
w/o TE	69.8	76.6	30.9	35.4	71.0	35.8
w/o SC	70.9	77.6	32.3	36.2	71.9	37.2
Core (GS)	66.6	73.8	30.8	35.8	67.6	34.5
Core (FS)	60.7	74.1	27.5	33.7	62.2	32.9
Only-Spatial	50.6	70.8	26.1	34.6	51.3	28.9
Only-ALG	—	—	—	—	40.2	22.0
Only-ALG (FS)	—	—	—	—	49.5	26.4



Figure 4: Qualitative MOVi-C ablation on video 002, frame 13. Only-ALG-GS shows binding masks A because it has no trained spatial decoder; Only-Spatial, Core-GS, and Full show decoder masks α. The marked regions compare merged or fragmented foreground parts in the ablations with the corresponding Full prediction.

Increasing training context under a fixed spatial decoding budget. Video FG-ARI increases monotonically across the tested training lengths under both Whole and single-frame Reset (Figure 3(c)). Image FG-ARI under single-frame Reset changes much less and varies nonmonotonically. Thus, in this single-seed sweep, the benefit of longer training context persists when frames are processed independently at inference. From 4 to 24 training frames, model-side time per update and peak allocated memory rise by factors of 1.45 and 2.04, a smaller proportional increase than the sixfold clip length (CoSlot w/o MFP rows of Table D.2).

4.3 Ablations of Shared-Prototype and Spatial Reconstruction

We ablate shared-prototype reconstruction, spatial reconstruction of sampled frames, the spatial decoding budget, and auxiliary components on MOVi-C.

Full uses ALG with GS prototypes, Spatial, MFP, TE, and SC. GS is the default ALG prototype source; FS is marked explicitly. Three single-component ablations remove MFP, TE, or SC; Core removes all three. Both Core variants spatially decode from frame-level slot features and differ only in whether the ALG prototypes come from GS or FS. Only-Spatial and Only-ALG remove ALG and Spatial, respectively, from Core. Removing TE disables only the learned aggregation of evidence across frames; basic temporal pooling remains. The A columns report binding attention masks, which directly reflect the assignments and are the only masks Only-ALG produces.

Shared-prototype reconstruction and spatial reconstruction. Adding ALG to sampled-frame spatial reconstruction improves video FG-ARI more than image FG-ARI (Core (GS) versus Only-Spatial). Figure 4 gives a single-frame view: Only-ALG-GS assigns broad binding regions, while Only-Spatial separates local parts but also fragments some foreground objects. Core-GS and Full recover more complete object regions in this example, although neither exactly matches the ground truth. In Core, GS prototypes improve video FG-ARI over FS prototypes for both α and A, while

Table 2: Standard object discovery results. CoSlot reports mean ± sample standard deviation over three training seeds. A dash denotes a value not reported in the source main table.

	MOVi-C		MOVi-E		YouTube-VIS 2021	
Method	FG-ARI ↑	mBO ↑	FG-ARI ↑	mBO ↑	FG-ARI ↑	mBO ↑
SAVi	22.2	13.6	42.8	16.0	—	—
STEVE	36.1	26.5	50.6	26.6	15.0	19.1
VideoSAUR	64.8	38.9	73.9	35.6	28.9	26.3
VideoSAURv2	—	—	77.1	34.4	31.2	29.7
SlotContrast	69.3	32.7	82.9	29.2	38.0	33.7
SRL	74.3	34.5	81.9	29.3	42.9	35.6
CoSlot	72.2 ± 0.4	31.5 ± 0.5	81.9 ± 0.3	26.7 ± 0.4	30.2 ± 0.6	31.6 ± 0.5

image FG-ARI changes little; with ALG alone, the ordering reverses for A. One possible explanation is that, without the within-frame structural constraints supplied by sampled-frame spatial reconstruction, the assignments are driven only by feature reconstruction error. Prototypes shared across frames then behave more like similarity clusters over clip features, potentially grouping similar-looking local regions into the same slot without necessarily forming complete objects.

Budget choice of one spatially reconstructed frame. With Full CoSlot's 24-frame training context and one MFP target frame decoded, video FG-ARI changes little and video mBO improves modestly as spatial reconstruction increases from one to two frames; neither metric improves further at four frames (Table D.1). Peak allocated memory and time per update rise, with the four-frame setting taking about 5.4 times as long per update as the one-frame setting. Based on this single-seed comparison, we use one visible frame for spatial reconstruction by default.

Masked-frame prediction and slot-update components. Removing MFP lowers video FG-ARI under both Whole binding (Table 1) and single-frame GS passing (Appendix C.1). This ablation removes input masking and target-frame prediction together. TE and SC also contribute to video FG-ARI in the Full comparison. Their effects do not extend uniformly to overlap: removing SC raises α video mBO and leaves A video mBO unchanged.

4.4 Object Discovery Results and Training Cost

Table 2 compares object discovery quality on complete videos. CoSlot binds the full video jointly in an offline setting; the baselines retain their respective frame-by-frame recurrent configurations. SRL results come from its original paper, and the other baseline results come from SlotContrast's main table.

CoSlot is competitive with SlotContrast and SRL on MOVi video FG-ARI, though its mBO is lower than both on MOVi-C and MOVi-E. On YouTube-VIS 2021, both video FG-ARI and mBO trail these baselines; its mBO exceeds VideoSAUR and VideoSAURv2. CoSlot is also cheaper to train: measured on precomputed frozen features with each method's native training graph, it has lower peak allocated memory and time per update than SlotContrast and SRL on both MOVi-C and YouTube-VIS, although on MOVi-C it trains on 24-frame clips while the baselines use four. On YouTube-VIS, where all methods use four frames, CoSlot needs about 59% of SlotContrast's time per update (Table D.3; protocol in Appendix B.3.2).

The state formed by joint binding can also be reused across chunks of real videos. Appendix C.2 examines the first 48 frames of 39 validation videos that are at least 48 frames long. Relative to independently binding two 24-frame chunks, GS passing and joint binding over 48 frames improve video FG-ARI for all three training seeds, while image-level metrics change little. These checkpoints were trained on four-frame clips, showing that they can use longer inference context through state passing or joint binding.

Frozen features also affect grouping and mask overlap quality. In Appendix E, PCA visualizations of DINOv2 features and the output masks show similar background stripes. After retraining with DINOv3 using one seed and the same training hyperparameters, video mBO increases on both MOVi-C and MOVi-E, while video FG-ARI decreases on both.

5 Discussion and Limitations

Discussion. Our experiments indicate that cross-frame consistency depends on both training and inference context. Longer training clips improve video FG-ARI even under independent single-frame Reset, suggesting that training learns more consistent cross-frame slot assignments without relying on inference-time state. For a fixed checkpoint, the organization of temporal context at inference remains consequential: Whole jointly binds all visible frames, whereas GS passing carries compact slot states between chunks. These mechanisms use context differently and can benefit models trained on shorter clips when inference spans longer videos (Section 4.2; Appendix C.2). The results also support choosing binding scope separately from dense decoding. With one visible frame spatially reconstructed per clip, longer training clips improve video FG-ARI while the measured endpoint increases in model-side time per update and peak allocated memory are proportionally smaller than the increase in clip length (Figure 3(c)). Decoding additional visible frames yields little further video FG-ARI gain and a modest mBO gain, at greater resource cost (Section 4.3).

Limitations and Future Work. CoSlot's video mBO on MOVi-C/E remains below SlotContrast and SRL. A possible explanation is that shared prototypes absorb similar feature patterns that persist at fixed positions across frames, potentially carrying frozen-backbone background stripes into binding and masks. The feature visualizations and backbone comparison in Appendix E are consistent with a feature contribution but do not establish this prototype mechanism. On YouTube-VIS, CoSlot trails SlotContrast and SRL (Table 2), and extending training clips there does not consistently improve performance (Appendix C.2). CoSlot uses a fixed number of slots, and GS passing across chunks is applied only at inference; without explicit state management for objects entering or leaving a video, these choices may limit handling of frequent entry and exit. Future work could train state passing between sequential chunks while retaining parallel joint binding within each chunk, add a contrastive objective between adjacent global slot states, and manage object entry and exit explicitly. Training-length and component ablations are primarily single-seed comparisons.

6 Conclusion

CoSlot jointly binds visible-frame features with clip-level shared slots, reconstructs all visible features with shared prototypes, and spatially reconstructs sampled visible frames. The diagnostics show complementary roles for training and inference context: longer training clips improve cross-frame consistency even under single-frame Reset, while binding scope and slot-state passing still affect a fixed model's predictions. Separating the number of frames used for binding from the number densely decoded makes longer training context practical with one visible frame spatially reconstructed per clip. CoSlot achieves competitive video FG-ARI on MOVi-C/E and demonstrates real-video feasibility on YouTube-VIS. In the evaluated configurations, its model-side memory and time per optimizer update on precomputed features are lower than SlotContrast and SRL.

AI use statement

In this work, we used generative AI tools to brainstorm and provide feedback on the research methodology, to help design experiments, to help interpret results, and to create and edit software code for experiments, evaluation, and figure generation. We also used them to edit the paper text for readability and to format references. We have not used generative AI tools to generate or clean data sets, and proofs and translation are not applicable to this work. We have reviewed all AI-assisted work: the authors checked AI-suggested ideas and experimental designs against the literature and our own experiments, checked and tested AI-assisted code, and traced every reported number to its training configuration, checkpoint, and evaluation output. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Reproducibility statement

Section 3 and Appendix A specify the model, losses, and binding and decoding procedures. Appendix B lists the training configurations and the evaluation, cost, and resource protocols, and Appendices C–D describe the protocols behind each additional analysis. All experiments use public datasets (MOVi-C, MOVi-E, and YouTube-VIS).

REFERENCES

Görkay Aydemir, Weidi Xie, and Fatma Güney. Self-supervised object-centric learning for videos. In *NeurIPS*, 2023. URL https://arxiv.org/abs/2310.06907.

Zhipeng Bao, Pavel Tokmakov, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert. Object discovery from motion-guided tokens. In *CVPR*, 2023. URL https://arxiv.org/abs/2303.15555.

Ondrej Biza, Sjoerd van Steenkiste, Mehdi S. M. Sajjadi, Gamaleldin F. Elsayed, Aravindh Mahendran, and Thomas Kipf. Invariant slot attention: Object discovery with slot-centric reference frames. In *ICML*, 2023. URL https://arxiv.org/abs/2302.04973.

Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised Scene Decomposition and Representation. *arXiv preprint arXiv:1901.11390*, 2019. URL https://arxiv.org/abs/1901.11390.

Gamaleldin Fathy Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael Curtis Mozer, and Thomas Kipf. SAVi++: Towards End-to-End Object-Centric Learning from Real-World Videos. In *NeurIPS*, 2022. URL https://openreview.net/forum?id=fT9W53lLxNS.

Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations. In *ICLR*, 2020. URL https://openreview.net/forum?id=BkxfaTVFwH.

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, Infer, Repeat: Fast Scene Understanding with Generative Models. In *NeurIPS*, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/52947e0ade57a09e4a1386d08f17b656-Abstract.html.

Ke Fan, Zechen Bai, Tianjun Xiao, Dominik Zietlow, Max Horn, Zixu Zhao, Carl-Johann Simon-Gabriel, Mike Zheng Shou, Francesco Locatello, Bernt Schiele, Thomas Brox, Zheng Zhang, Yanwei Fu, and Tong He. Unsupervised open-vocabulary object localization in videos. In *ICCV*, pp. 13701–13709, 2023. doi: 10.1109/ICCV51070.2023.01264. URL https://doi.org/10.1109/ICCV51070.2023.01264.

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*, pp. 23062–23071, 2024. doi: 10.1109/CVPR52733.2024.02176. URL https://doi.org/10.1109/CVPR52733.2024.02176.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *ICML*, pp. 2424–2433, 2019. URL https://proceedings.mlr.press/v97/greff19a.html.

Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. On the Binding Problem in Artificial Neural Networks. *arXiv preprint arXiv:2012.05208*, 2020. URL https://arxiv.org/abs/2012.05208.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A Scalable Dataset Generator. In *CVPR*, 2022. URL https://arxiv.org/abs/2203.03570.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pp. 16000–16009, 2022. doi: 10.1109/CVPR52688.2022.01553. URL https://openaccess.thecvf.com/content/CVPR2022/html/He_Masked_Autoencoders_Are_Scalable_Vision_Learners_CVPR_2022_paper.html.

Baoxiong Jia, Yu Liu, and Siyuan Huang. Improving object-centric learning with query optimization. In *ICLR*, 2023. URL https://arxiv.org/abs/2210.08990.

Jindong Jiang, Sepehr Janghorbani, Gerard de Melo, and Sungjin Ahn. SCALOR: Generative World Models with Scalable Object Representations. In *ICLR*, 2020. URL https://arxiv.org/abs/1910.02384.

Jindong Jiang, Fei Deng, Gautam Singh, Minseung Lee, and Sungjin Ahn. Slot state space models. In *NeurIPS*, 2024. doi: 10.52202/079017-0370. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/158ac5698e36a01ee5ca9e6732685b34-Abstract-Conference.html.

Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matthew Botvinick, Alexander Lerchner, and Christopher P. Burgess. SIMONe: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. In *NeurIPS*, 2021. URL https://openreview.net/forum?id=YSzTMntO1KY.

Jinwoo Kim, Janghyuk Choi, Ho-Jin Choi, and Seon Joo Kim. Shepherding slots to objects: Towards stable and robust object-centric learning. In *CVPR*, 2023. URL https://arxiv.org/abs/2303.17842.

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. URL https://openreview.net/forum?id=aD7uesX1GF_.

Adam Kosiorek, Hyunjik Kim, Yee Whye Teh, and Ingmar Posner. Sequential Attend, Infer, Repeat: Generative Modelling of Moving Objects. In *NeurIPS*, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/7417744a2bac776fabe5a09b21c707a2-Abstract.html.

Zhiyuan Li, Rongzhen Zhao, Wenyan Yang, Wenshuai Zhao, Pekka Marttinen, and Joni Pajarinen. Rethinking temporal consistency in video object-centric learning: From prediction to correspondence. *arXiv preprint arXiv:2605.03650*, 2026. URL https://arxiv.org/abs/2605.03650v1.

Zhixuan Lin, Yi-Wu Fu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. SPACE: Unsupervised Object-Oriented Scene Representation via Spatial Attention and Decomposition. In *ICLR*, 2020. URL https://openreview.net/forum?id=rkl03ySYDH.

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*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/8511df98c02ab60aea1b2356c013bc0f-Paper.pdf.

Anna Manasyan, Maximilian Seitzer, Filip Radovic, Georg Martius, and Andrii Zadaianchuk. Temporally consistent object-centric learning by contrasting slots. In *CVPR*, pp. 5401–5411, 2025. doi: 10.1109/CVPR52734.2025.00508. URL https://doi.org/10.1109/CVPR52734.2025.00508.

WonJun Moon and Jae-Pil Heo. Selective synergistic learning for video object-centric learning. *arXiv preprint arXiv:2606.15527*, 2026. URL https://arxiv.org/abs/2606.15527.

WonJun Moon, Hyun Seok Seong, and Jae-Pil Heo. Reconstruction-guided slot curriculum: Addressing object over-fragmentation in video object-centric learning. *arXiv preprint arXiv:2603.22758*, 2026. URL https://arxiv.org/abs/2603.22758.

Duc Nguyen, Sieu Tran, Hao Vo, Khoa Vo, Duy Minh Ho Nguyen, Nghi D. Q. Bui, Anh Nguyen, Long Mai, and Ngan Le. TSA: Temporal slot activation for persistent object-centric video representation. *arXiv preprint arXiv:2606.13714*, 2026. URL https://arxiv.org/abs/2606.13714.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

Rui Qian, Shuangrui Ding, Xian Liu, and Dahua Lin. Semantics meets temporal correspondence: Self-supervised object-centric learning in videos. In *ICCV*, 2023. URL https://arxiv.org/abs/2308.09951.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *ICLR*, 2023. URL https://openreview.net/forum?id=b9tUk-f_aG.

Hyun Seok Seong, WonJun Moon, and Jae-Pil Heo. From vicious to virtuous cycles: Synergistic representation learning for unsupervised video object-centric learning. *arXiv preprint arXiv:2602.03390*, 2026. doi: 10.48550/ARXIV.2602.03390. URL https://doi.org/10.48550/arXiv.2602.03390.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *NeurIPS*, 2022. URL https://openreview.net/forum?id=eYfIM88MTUE.

Gautam Singh, Yue Wang, Jiawei Yang, Boris Ivanovic, Sungjin Ahn, Marco Pavone, and Tong Che. Parallelized spatiotemporal slot binding for videos. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *ICML*, pp. 45707–45733, 2024. URL https://proceedings.mlr.press/v235/singh24g.html.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/416f9cb3276121c42eebb86352a4354a-Abstract-Conference.html.

Sieu Tran, Duc Nguyen, Hao Vo, Khoa Vo, and Ngan Le. Dual-state slot attention: Decoupling appearance and identity for video object-centric learning. *arXiv preprint arXiv:2606.12601*, 2026. URL https://arxiv.org/abs/2606.12601.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. In *ICLR*, 2023. URL https://openreview.net/forum?id=TFbwV6I0VLg.

Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. URL https://arxiv.org/abs/1905.04804.

Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. In *NeurIPS*, 2023. URL https://arxiv.org/abs/2306.04829.

Rongzhen Zhao, Jian Li, Juho Kannala, and Joni Pajarinen. Predicting video slot attention queries from random slot-feature pairs. *arXiv preprint arXiv:2508.01345*, 2025a. URL https://arxiv.org/abs/2508.01345.

Rongzhen Zhao, Wenyan Yang, Juho Kannala, and Joni Pajarinen. Smoothing slot attention iterations and recurrences. *arXiv preprint arXiv:2508.05417*, 2025b. doi: 10.48550/ARXIV.2508.05417. URL https://doi.org/10.48550/arXiv.2508.05417.

Rongzhen Zhao, Zhiyuan Li, Juho Kannala, and Joni Pajarinen. Internalizing temporal consistency in video object-centric learning without explicit regularization. *arXiv preprint arXiv:2605.31508*, 2026. URL https://arxiv.org/abs/2605.31508.

A Model and Loss Details

A.1 Binding Iterations and Compact Computation of the Algebraic Reconstruction Loss

Binding iterations. Following the notation of Section 3.1, let i index binding iterations rather than video time. The current global slots determine token compatibility scores and assignment weights:

$$\ell_{t,k,p}^{(i)} = \frac{q(z_k^{(i)})^\top \kappa(h_{t,p})}{\sqrt{d}}, \qquad a_{t,k,p}^{(i)} = \mathrm{softmax}_k(\ell_{t,k,p}^{(i)}), \tag{8}$$

where q, κ, v are query, key, and value mappings, respectively, and d is the query/key dimension. Normalize the assignment weights over patch positions within each frame to pool within-frame evidence:

$$\bar{a}_{t,k,p}^{(i)} = \frac{a_{t,k,p}^{(i)}}{\sum_{p'} a_{t,k,p'}^{(i)}}, \qquad e_{t,k}^{(i)} = \sum_p \bar{a}_{t,k,p}^{(i)} v(h_{t,p}). \tag{9}$$

The within-frame evidence is refined through attention across slots. The current slot then acts as a query to aggregate evidence over the visible frames:

$$\widetilde{E}_t^{(i)} = \mathrm{SlotCoordination}(E_t^{(i)}), \qquad g_k^{(i)} = \mathrm{TemporalEvidence}\Big(z_k^{(i)}, \{\widetilde{e}_{t,k}^{(i)}\}_{t\in\mathcal{V}}\Big), \tag{10}$$

$$z_k^{(i+1)} = \mathrm{Update}_{\mathrm{GRU/MLP}}\left(z_k^{(i)}, g_k^{(i)}\right). \tag{11}$$

$\mathrm{SlotCoordination}$ acts on the K evidence vectors within a frame; $\mathrm{TemporalEvidence}$ aggregates the corresponding evidence across frames, conditioned on the slot. Without SlotCoordination, $\widetilde{E}_t = E_t$. Without TE, an equal-weight mean of the per-slot evidence over valid frames replaces learned temporal attention: $g_k^{(i)} = |\mathcal{V}|^{-1} \sum_{t\in\mathcal{V}} \widetilde{e}_{t,k}^{(i)}$. The result still enters the subsequent GRU/MLP slot update. The updated Z enters the next round of token competition. After the iterations, the final global slots determine the output assignment weights A; the frame-level slot features are pooled using the normalized assignments:

$$s_{t,k} = \frac{\sum_p a_{t,k,p}\, v(h_{t,p})}{\sum_p a_{t,k,p}}. \tag{12}$$

Thus, the data flow for spatial readout is $Z \to A \to S \to \alpha$. SlotCoordination and TemporalEvidence participate in iterative updates of the global slots, while the output S is read from the within-frame features using the final assignments.

Compact computation of the algebraic reconstruction loss. The squared error in Section 3.2 expands as

$$\|X - AU\|_F^2 = \|X\|_F^2 - 2\,\mathrm{tr}(U^\top A^\top X) + \mathrm{tr}(U^\top A^\top A U). \tag{13}$$

Let $B = A^\top X \in \mathbb{R}^{K\times D}$ and $C = A^\top A \in \mathbb{R}^{K\times K}$. The last two terms can then be computed as $\langle U, B\rangle_F$ and $\langle C, UU^\top\rangle_F$, respectively. The implementation needs only the sum of squared features and these compact matrix statistics, without constructing per-slot, per-position feature outputs. Storage for assignment weights is $O(NK)$; supervision over all tokens is computed through matrix statistics of A and the input features.

A.2 Additional Decoding and Training Details

Expanded spatial readout. Let ξ_p be a spatial positional encoding. For each frame-level slot feature, the broadcast MLP produces features and mask logits:

$$(b_{t,k,p}, \eta_{t,k,p}) = d_\phi(s_{t,k}, \xi_p), \qquad \alpha_{t,k,p} = \mathrm{softmax}_k(\eta_{t,k,p}), \tag{14}$$

$$\widehat{h}^{\mathrm{sp}}_{t,p} = \sum_{k=1}^{K} \alpha_{t,k,p} b_{t,k,p}, \qquad \mathcal{L}_{\mathrm{sp}} = \frac{1}{mP} \sum_{t\in\mathcal{S}} \sum_{p=1}^{P} \left\| h_{t,p} - \widehat{h}^{\mathrm{sp}}_{t,p} \right\|_2^2, \qquad m = |\mathcal{S}|. \tag{15}$$

For a fixed set of visible frames and model, uniformly sampling m frames from $\mathcal{V}$ and averaging their losses gives

$$\mathbb{E}_{\mathcal{S}}[\mathcal{L}_{\mathrm{sp}}] = \frac{1}{|\mathcal{V}|P} \sum_{t\in\mathcal{V}} \sum_{p=1}^{P} \left\| h_{t,p} - \widehat{h}^{\mathrm{sp}}_{t,p} \right\|_2^2. \tag{16}$$

Sampling therefore gives an unbiased estimate of the mean spatial loss over all visible frames. The decoder input here is always the frame-level slot features S_t.

MFP target sampling and auxiliary loss. No frame in $\mathcal{M}$ enters binding. The predictor takes visible-frame slot features with their temporal encodings added as context, and learnable mask queries with target-frame temporal encodings added as queries. It performs cross-attention for each slot and uses visible evidence both before and after the target frame; temporal information distinguishes the times associated with visible-frame features and target queries. Let $\mathcal{D} \subseteq \mathcal{M}$ be the subset of targets actually used for decoding supervision: n_{target} determines the number of masked frames, and $n_{\mathrm{decode}} = |\mathcal{D}|$ determines the auxiliary decoding budget.

The auxiliary decoder uses the broadcast and mixture operations above, with predicted slot features $\widehat{s}_{t,k}$ as its inputs and ϕ_{mp} as its parameters, to obtain $\widehat{h}^{\mathrm{mask}}_{t,p}$. It has the same type of architecture as the visible-frame decoder but separate parameters. Let sg denote stop-gradient, and define

$$c = \mathrm{sg}\left[\max\left(\frac{\mathrm{RMS}(H_{\mathcal{D}})}{r_{\mathrm{ref}}}, 1 \right) \right], \tag{17}$$

The auxiliary loss is then

$$\mathcal{L}_{\mathrm{mask}} = \frac{1}{|\mathcal{D}|P} \sum_{t\in\mathcal{D}} \sum_{p=1}^{P} \left\| \frac{\widehat{h}^{\mathrm{mask}}_{t,p} - h_{t,p}}{c} \right\|_2^2. \tag{18}$$

r_{ref} is a fixed reference RMS. During backpropagation, c is treated as a constant to limit the effect of high-magnitude targets on the auxiliary loss. Masked block size, the number of decoded targets, and predictor configuration are given in Appendix B.1.

ALG branch gradients. Let $L(A, U)$ be the mean ALG reconstruction error and sg the stop-gradient operator. MOVi datasets use standard backpropagation through $L(A, U)$. For YouTube-VIS, optimization uses

$$L_{\mathrm{split}} = L(\mathrm{sg}(A), U) + w_A L(A, \mathrm{sg}(U)), \qquad w_A = 0.01. \tag{19}$$

Table B.1: CoSlot training configurations.

Setting	MOVi-C	MOVi-E	YouTube-VIS 2021
Training			
Training steps	100k	150k	40k
Effective clip batch	128	128	64
Physical batch / devices / accumulation	64 / 2 / 1	64 / 2 / 1	64 / 1 / 1
Training clip length	24	24	4
Optimizer	Adam	Adam	Adam
Peak learning rate	0.002	0.002	0.002
Learning-rate warmup	1000 steps	1000 steps	1000 steps
Exponential decay	100k	300k	100k
Gradient norm clipping	1.0	1.0	1.0
Vision Backbone			
ViT architecture	DINOv2 Small	DINOv2 Base	DINOv2 Base
Patch size	14	14	14
Feature dimension	384	768	768
Image Specifications			
Image / crop size	336	336	518
Cropping strategy	Full	Full	Center crop
Image tokens	576	576	1369
Slot Attention and Architecture			
Slots	11	15	7
Slot dimension	64	128	64
Iterations	3	3	3
Feature projector	MLP, hidden 768	MLP, hidden 768	MLP, hidden 768
SlotCoordination	8 heads	8 heads	8 heads
TemporalEvidence	1 head	1 head	1 head
Decoder	MLP	MLP	MLP
CoSlot Parameters			
ALG / Spatial / MFP loss weight	1.0 / 1.0 / 1.0	1.0 / 1.0 / 1.0	1.0 / 1.0 / 1.0
ALG weight anneal (cosine, floor 0.3)	5k–20k	20k–80k	5k–20k
Spatial reconstruction frames	1	1	1
MFP targets / block / decoded	4 / 2 / 1	4 / 2 / 1	1 / 1 / 1

Both evaluations of the loss have identical forward values, so the recorded scalar is $(1 + w_A)L$. In real videos, subdividing objects according to local feature similarity may also reduce reconstruction error; $w_A = 0.01$ is intended to weaken this tendency through ALG's direct assignment path. The scaling does not apply to the prototype path: reconstruction supervision from all visible frames still reaches the binding network through the shared prototypes and global slots.

B Experimental Setup and Evaluation Protocols

B.1 Training Configurations and Comparison Settings

Table B.1 lists CoSlot's training configurations, all using NVIDIA RTX 5090 GPUs. Training and evaluation read cached features from frozen backbones, without data augmentation. Multi-seed results use three independent training seeds and report the mean and sample standard deviation. Single-seed experiments use the same fixed random seed, which is also one of those three seeds. Paired gains are computed by subtracting unrounded results for matching seeds before calculating their mean and sample standard deviation.

YouTube-VIS is trained for 40k steps; its validation metrics had already plateaued before that point. The gradient weight for its ALG output-assignment branch is 0.01, versus 1.0 for MOVi; Appendix A.2 defines this branch gradient precisely. The spatial and MFP decoders each have three hidden layers of width 1024. The ALG prototype head has one hidden layer of width twice the slot dimension. The reference target RMS is 4.2. Training uses bf16 mixed precision, and evaluation uses float32.

All Figure 3(c) configurations have an effective batch size of 128 clips. Lengths 4/12/24 use two GPUs with 64 clips per GPU and one accumulation step; length 8 uses one GPU with 64 clips and two accumulation steps. All use 100k training steps, spatially decode one frame per clip, and disable MFP. Loss normalization does not change with training length (ALG is averaged over visible tokens), and the frozen features and training set are the same. Both Core (GS prototypes) and Core (FS prototypes) in Table 1 are trained with 128 clips per update.

B.2 Evaluation Protocols

Data and metric computation. We follow SlotContrast's dataset splits: MOVi-C and MOVi-E use all 250 videos in their official validation sets, with 24 frames per video. For YouTube-VIS 2021, we reserve the last 300 videos by video ID from the official training set as validation data, use the remainder for training, and evaluate 200 of those validation videos at their full lengths. At evaluation, MOVi frames are resized directly to 336×336, while a central square is cropped from each YouTube-VIS frame and resized to 518×518. Predicted masks are upsampled to the same resolution before comparison with ground truth.

Inference context. Chunked inference partitions T frames into $J = T/L$ chunks of length L. The protocols are defined as follows:

- **Whole:** Perform one joint binding operation over the full video. Observations before and after the current frame are available, making this offline inference.
- **Reset / GS passing:** Split the video into contiguous chunks in temporal order. Under Reset, every chunk starts from the learned slot initialization; under GS passing, the global slots from the preceding chunk initialize the next one. With $L = 1$, GS passing uses only the current frame and prior state.
- **Interleaved:** Chunk s contains frames $s, s + J, s + 2J, \ldots$. Chunks are inferred independently, with each spanning the full temporal range.
- **Full-reference aligned:** Use the already computed Whole predictions as a reference to optimally match the slot indices of independently inferred chunks by mask overlap before computing video-level metrics. This uses no ground truth and does not change within-frame partitions.

The 48-frame YouTube-VIS comparison uses the first 48 frames of 39 validation videos that are at least 48 frames long. It compares independent inference on two 24-frame chunks, global slot passing across two 24-frame chunks, and joint binding over 48 frames.

B.3 Measurement Protocol for Training Cost on Precomputed Features

All measurements use one RTX 5090, cached DINOv2 features, and BF16, with TF32 disabled. Timing covers only the training forward pass, backward pass, optimizer update, and necessary synchronization; it excludes the backbone, data pipeline, and work outside training. Each configuration completes 1000 stabilization updates and 50 timing warm-up updates before 200 timed updates. We report the median of the means of four groups of 50 updates, and PyTorch peak allocated memory (GiB) within the timing window.

B.3.1 Resource Protocol for CoSlot Training Configurations

For Figure 3(c) and Sections D.1 and D.2, the physical batch size is 64, with two gradient accumulation steps, giving an effective batch size of 128 clips. Memory includes two resident input buffers. Resource measurements use a newly initialized training graph with matching structure and are run separately from the quality experiments. The 4/24-frame endpoints of Figure 3(c) and the CoSlot no-MFP rows of Table D.2 share resource records.

B.3.2 Cross-Method Cost-per-Update Protocol

Each method is measured independently with a physical batch size of 16 and a small number of cached batches, retaining its native training graph and active losses. On MOVi-C, SlotContrast and SRL use four-frame configurations and CoSlot uses 24 frames; on YouTube-VIS, all use four frames.

This protocol matches CoSlot's actual training setup: CoSlot uses no data augmentation, and features for each frame are extracted only once before training, so feature-extraction cost does not depend on training clip length. The released configurations of SlotContrast and SRL use data augmentation and run DINOv2 on the fly during training; Table D.3 excludes that encoding cost.

SlotContrast and CoSlot use their loss terms throughout training, whereas SRL activates some losses in stages as training progresses, so its cost per update can vary by training stage (Seong et al., 2026). All three methods are timed at the same point in training using the stabilization, warm-up, and timing windows defined above.

C Additional Analyses of Temporal Context and Model Components

C.1 MOVi-C Components and Single-Frame Inference Protocols

With $T_{\text{eval}} = 24$ fixed, we compare Whole binding, Chunk-1 Reset, Chunk-1 Full-reference aligned, and Chunk-1 GS passing for four component checkpoints. All results are video FG-ARI based on decoder masks α. Full-reference aligned uses each checkpoint's own Whole predictions to reindex its single-frame Reset slots. This protocol matches the training-length comparison in Figure 3(c). The Full versus w/o MFP comparison covers the entire masking-and-prediction training scheme; Core versus Only-Spatial compares the presence or absence of ALG. Core also disables MFP, TE, and SC. Every row uses the same fixed random seed; the three-seed Chunk-1 GS passing result for Full is 66.3 ± 2.5 (Figure 3(b)).

ALG, MFP, and state passing. Compared with Only-Spatial, Core improves substantially under both Whole and single-frame Full-reference aligned inference, but its result with GS passing is lower than Only-Spatial's. The latter also performs better with GS passing than under its own Whole inference. Thus, ALG's effect in this setting varies with the inference protocol, and Whole is not an upper bound on performance for every configuration. Relative to w/o MFP, Full raises video FG-ARI under GS passing from 57.2 to 67.7; MFP training, including masking and prediction, substantially strengthens single-frame state passing.

Training configuration and inference protocol. Results vary substantially across training configurations under the same inference protocol. For a fixed checkpoint, switching from Reset to Full-reference aligned or GS passing also produces substantial changes, and the configurations benefit from state passing to different extents: Full and Only-Spatial gain 22.3 and 25.6 percentage points, respectively, while w/o MFP and Core gain 10.8 and 9.1 points. Together with Figure 3(c), these results indicate that training configuration and inference protocol both affect slot assignments and cross-frame consistency.

C.2 Training and Inference Context on YouTube-VIS 2021

The training-length sweep evaluates each 40k-step checkpoint, trained with the same fixed random seed, over the full lengths of 200 validation videos. The 48-frame inference-protocol experiment evaluates three independently trained four-frame Full checkpoints on the first 48 frames of 39 validation videos that are at least 48 frames long.

Training length. With 40k training steps and full-video evaluation fixed, we compare training contexts of 4/8/12 frames.

Four- and eight-frame training give similar results, whereas the twelve-frame result is lower. We do not observe the improvement with longer training clips seen on MOVi-C. The main results retain four-frame training, matching the two principal baselines.

48-frame inference protocols. We compare two independent 24-frame chunks, two 24-frame chunks with GS passing, and a single 48-frame joint binding operation.

Relative to independent chunks, GS passing and 48-frame joint binding raise video FG-ARI by 3.3 ± 1.2 and 4.4 ± 1.1 percentage points, respectively (paired over matching seeds), and both gains

Table C.1: MOVi-C components and single-frame inference protocols. Values are video FG-ARI from decoder masks α for one fixed seed.

MOVi-C variant	Whole-24	Chunk-1 Reset	Chunk-1 Full-reference aligned	Chunk-1 GS passing
Full	71.8	45.4	65.0	67.7
w/o MFP	67.4	46.4	59.8	57.2
Core	66.6	46.6	59.0	55.7
Only-Spatial	50.6	35.0	42.8	60.6

Table C.2: YouTube-VIS training length. Each row uses one fixed-seed 40k-step checkpoint evaluated over complete videos.

T_{train}	Video FG-ARI ↑	Video mBO ↑
4	30.4	31.1
8	30.9	31.3
12	28.7	30.9

are positive for every seed. Image-level metrics change little; the benefit is chiefly in cross-frame consistency. Thus, checkpoints trained on four-frame clips can use longer video context through state continuation and a larger binding scope.

Table C.3: YouTube-VIS 48-frame inference protocols. Values are three-seed means ± sample standard deviations on the first 48 frames of 39 validation videos.

Protocol	L_{bind}	Video FG-ARI ↑	Image FG-ARI ↑	Video mBO ↑	Image mBO ↑
Independent contiguous chunks (Contiguous reset)	24	29.3 ± 0.2	41.0 ± 1.7	30.8 ± 0.6	38.5 ± 0.4
Global slot passing (Contiguous GS passing)	24	32.5 ± 1.1	42.1 ± 2.1	32.5 ± 0.3	38.5 ± 0.1
Joint binding over the full sequence (Whole)	48	33.7 ± 1.1	41.6 ± 2.9	33.2 ± 0.3	38.6 ± 0.3

D Spatial Decoding Budget and Training Cost

Sections D.1 and D.2 use the same fixed random seed. In these two sections, quality is evaluated on complete 24-frame MOVi-C videos, and resource measurements follow Appendix B.3.1.

D.1 Spatial Decoding Budget

We fix Full CoSlot's training context ($T_{\text{train}} = 24$) and the number of MFP target frames decoded, varying only the number of visible frames spatially reconstructed per update. All configurations are evaluated using joint binding over the full video (Whole).

Table D.1: Spatial decoding budget for Full CoSlot. The number of MFP target frames decoded is fixed at one. Quality is evaluated using Whole inference on 24-frame MOVi-C videos.

Spatial frames	MFP decoded frames	Peak allocated (GiB) ↓	ms/update ↓	Video FG-ARI ↑	Video mBO ↑
1	1	13.95	332.71	71.8	31.3
2	1	16.77	432.83	72.3	32.3
4	1	25.02	1813.15	72.2	32.0

Video FG-ARI differs by at most 0.5 percentage points across the three configurations, and video mBO by at most 1.0 point. Peak allocated memory and time per update increase with the number of decoded frames; the four-frame configuration takes about 5.4 times as long per update as the one-frame configuration.

D.2 Dense Readout Reference Adapted from U-OVOL

U-OVOL-style dense is an adaptation of the U-OVOL readout design to this study's frozen per-frame DINOv2 features and object discovery task (Fan et al., 2023). It reconstructs the full feature grids of all training frames from clip-level global slots and spatiotemporal coordinates. Table D.2 also includes CoSlot with MFP disabled under the same measurement protocol. Models with $T_{\text{train}} = 4$ are evaluated using six contiguous four-frame chunks with global slot passing (6×4, GS passing); models with $T_{\text{train}} = 24$ use Whole.

The two readouts are complete multi-frame training configurations that differ in spatial readout input, the number of densely decoded frames per update, and their training objectives. U-OVOL-style dense decodes every training frame from global slots and spatiotemporal coordinates. CoSlot decodes one sampled frame from frame-level slot features and applies shared-prototype reconstruction to all visible tokens. The contribution of sampled-frame spatial reconstruction when combined with shared-prototype reconstruction within CoSlot is tested separately by the Only-Spatial versus Core comparison in Table 1.

Under four-frame training and the same inference protocol, U-OVOL-style dense has higher video FG-ARI (61.1 versus 57.3), whereas CoSlot w/o MFP has higher video mBO (27.6 versus 25.0). CoSlot w/o MFP uses about 29% of its peak memory and 32% of its time per update. When training clips extend to 24 frames, U-OVOL-style dense runs out of memory (OOM) during stabilization with

Table D.2: Reference implementation of a dense readout adapted from U-OVOL. Configuration and results panels list the same four runs in the same order. The two readouts are distinct complete training configurations. OOM denotes an out-of-memory failure during stabilization under the same hardware and physical batch size.

(a) Configuration

Method / readout	Spatial readout input	T_{train}	Dense frames per update	Inference protocol
U-OVOL-style dense	GS + spatiotemporal coordinates	4	4	6×4, GS passing
U-OVOL-style dense	GS + spatiotemporal coordinates	24	24	Whole
CoSlot w/o MFP	FS	4	1	6×4, GS passing
CoSlot w/o MFP	FS	24	1	Whole

(b) Resources and quality

Method / readout	Peak allocated (GiB) ↓	ms/update ↓	Video FG-ARI ↑	Video mBO ↑
U-OVOL-style dense (4 frames)	17.66	435.25	61.1	25.0
U-OVOL-style dense (24 frames)	OOM	—	—	—
CoSlot w/o MFP (4 frames)	5.17	141.02	57.3	27.6
CoSlot w/o MFP (24 frames)	10.55	203.95	67.4	30.4

the same hardware and physical batch size; CoSlot w/o MFP has peak allocated memory of 10.55 GiB and achieves 67.4 Whole video FG-ARI.

D.3 Cross-Method Training Cost

Following the internal spatial-budget comparison and the dense-readout reference, we compare CoSlot with SlotContrast and SRL as complete training methods. Table D.3 follows the protocol with a physical batch size of 16 in Appendix B.3.2. CoSlot uses less peak allocated memory and less time per update on both datasets: MOVi-C compares CoSlot's 24-frame training with four-frame baselines, while all methods use four frames on YouTube-VIS.

Table D.3: Model training cost per optimizer update on precomputed features. Physical batch size is 16; all methods read cached frozen features, and backbone extraction and data I/O are excluded.

Dataset	Method	T_{train}	Peak allocated (GiB) ↓	ms/update ↓
MOVi-C	SlotContrast	4	5.86	89.13
MOVi-C	SRL	4	5.96	141.63
MOVi-C	CoSlot Full	24	3.23	67.57
YouTube-VIS 2021	SlotContrast	4	10.65	149.18
YouTube-VIS 2021	SRL	4	21.07	286.79
YouTube-VIS 2021	CoSlot Full	4	4.15	87.98

E Frozen Features and Backbone Replacement

CoSlot's binding and reconstruction both operate directly on frozen features, so the spatial structure of the input features enters both grouping and readout. We first compare stripes visible in DINOv2 features and masks, then replace the backbone with DINOv3 without retuning the training hyperparameters to examine how frozen features affect grouping and overlap quality.

E.1 PCA of Frozen Features and Stripes in Masks



Figure E.1: Frozen-feature PCA for MOVi-C validation video 016, frame 4. From left to right: input image, ground-truth overlay, DINOv2 features, and DINOv3 features from the same frame. The DINOv2 projection shows conspicuous vertical variation in the background; the DINOv3 projection is smoother in this example. PCA colors are not object labels.



Figure E.2: Mask comparison on validation video 016, frame 4, for MOVi-C (upper block) and MOVi-E (lower block). Within each block, the columns are DINOv2 binding masks A, DINOv2 decoder masks α, DINOv3 A, and DINOv3 α. Stripe-like background partitions are prominent in the DINOv2 masks and less visible in the DINOv3 masks in these examples. Slot colors are local to each prediction.

Background stripes in the feature PCA appear with similar patterns in A and α, suggesting that the spatial structure of frozen features carries through to binding and readout.

E.2 DINOv3 Replacement under the Same Training Setup

On MOVi-C/E, we replace the frozen backbone with DINOv3 (Siméoni et al., 2025) at the corresponding model scale and train CoSlot from scratch with the same architecture, loss weights, and

Table E.1: Comparison of frozen DINOv2 and DINOv3 features.

Dataset	Frozen backbone	α video		α image	
		FG-ARI ↑	mBO ↑	FG-ARI ↑	mBO ↑
MOVi-C	DINOv2 (three-seed α)	72.2 ± 0.4	31.5 ± 0.5	77.6 ± 0.1	35.4 ± 0.4
MOVi-C	DINOv3 (single seed)	68.2	38.0	74.6	43.3
MOVi-E	DINOv2 (three-seed α)	81.9 ± 0.3	26.7 ± 0.4	83.6 ± 0.2	29.7 ± 0.4
MOVi-E	DINOv3 (single seed)	76.3	28.6	77.8	32.2

Dataset	Frozen backbone	A video	
		FG-ARI ↑	mBO ↑
MOVi-C	DINOv2 (single-seed A)	73.1	37.2
MOVi-C	DINOv3 (single seed)	72.6	46.8
MOVi-E	DINOv2 (single-seed A)	81.9	30.6
MOVi-E	DINOv3 (single seed)	78.4	30.6

training setup. The DINOv2 α columns report the mean $\pm$ sample standard deviation over three seeds, whereas DINOv3 uses a single seed; the A columns report single-seed results for both backbones.

With the CoSlot architecture and training hyperparameters held fixed, replacing DINOv2 with DINOv3 raises α video mBO from 31.5 to 38.0 on MOVi-C and from 26.7 to 28.6 on MOVi-E, while video FG-ARI falls from 72.2 to 68.2 and from 81.9 to 76.3, respectively. Video mBO based on A rises from 37.2 to 46.8 on MOVi-C and remains unchanged on MOVi-E. In this comparison under fixed training settings, the choice of frozen features affects both grouping and overlap quality.

F Qualitative Results and Failure Cases

F.1 MOVi-C Ablation Across Time

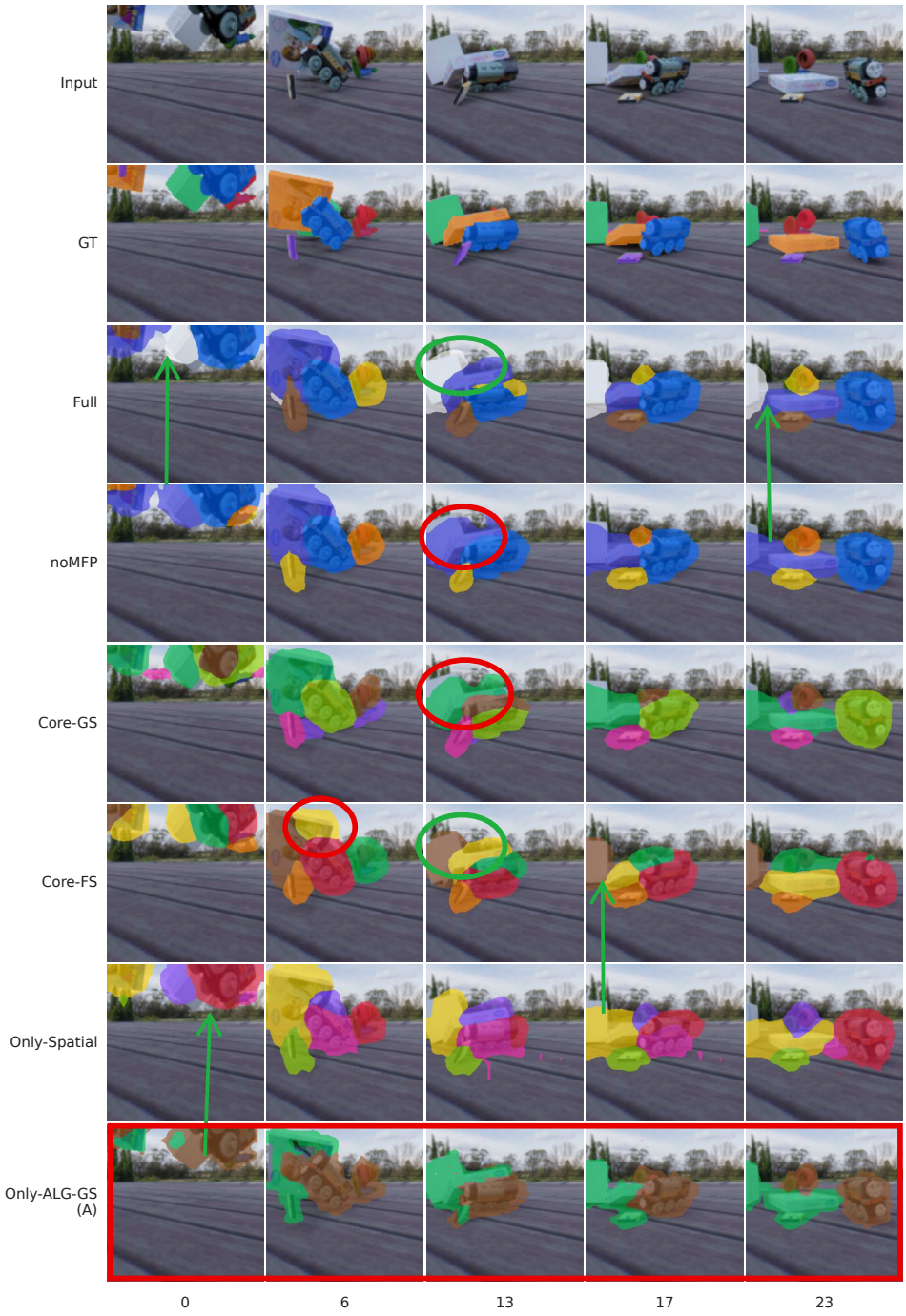

Figure F.1: Five frames of MOVi-C video 004 under whole-video inference. Rows show input, ground truth, Full, w/o MFP, Core-GS, Core-FS, Only-Spatial, and Only-ALG-GS. Only-ALG-GS displays binding masks A; all other predicted rows display decoder masks α. The marked regions compare object boundaries and assignments across the six trained variants at the same frames.

F.2 Cross-Method Comparisons on MOVi-C

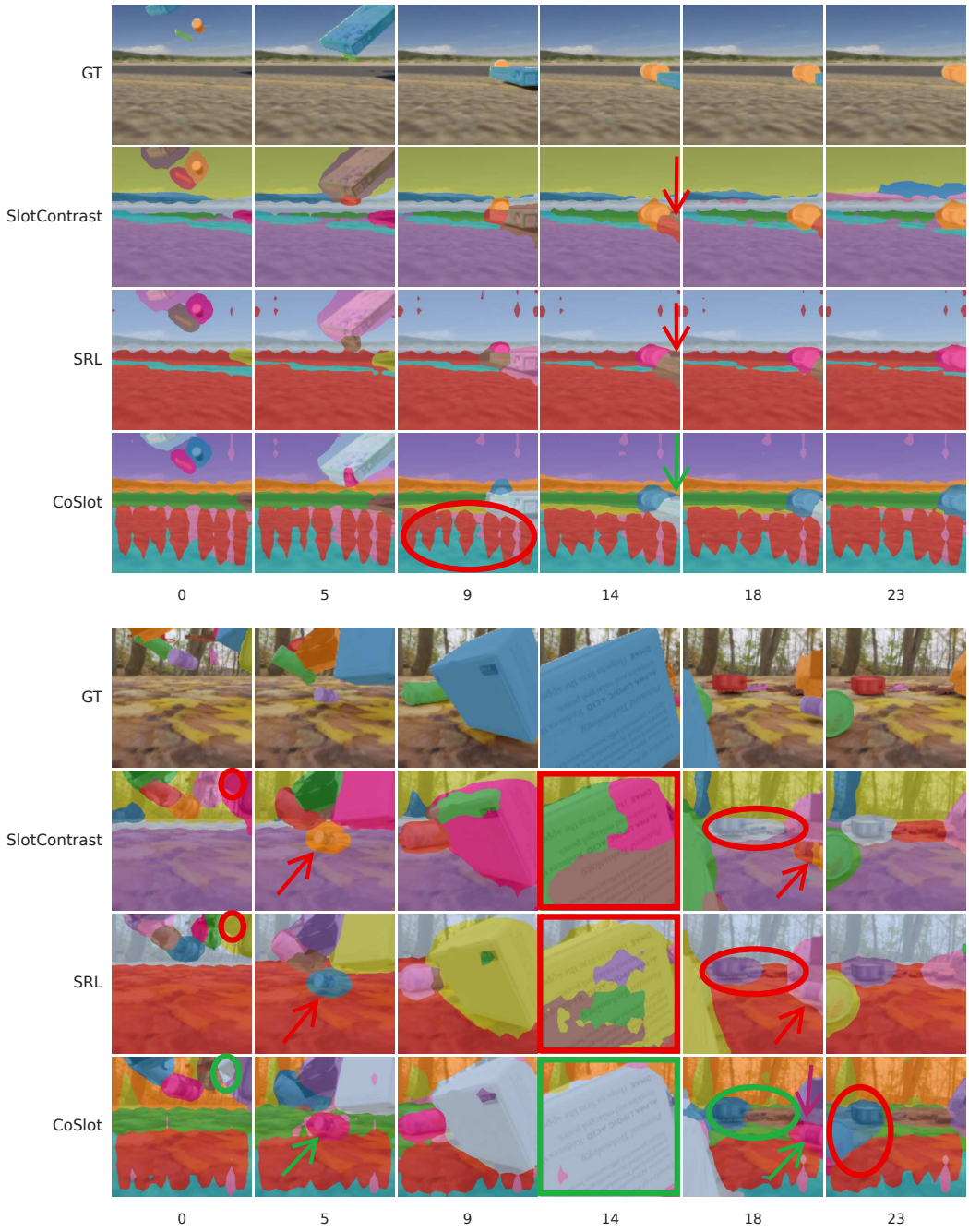

Figure F.2: Selected frames from MOVi-C videos 029 (upper) and 126 (lower), evaluated with whole-video CoSlot inference. Each block compares ground truth, SlotContrast, SRL, and CoSlot. Arrows and outlines identify regions for comparing assignments across frames. The CoSlot row of 029 also makes stripe-like background partitions visible; 126 includes a large foreground occlusion.

F.3 Occlusion Example on MOVi-C: SRL and CoSlot

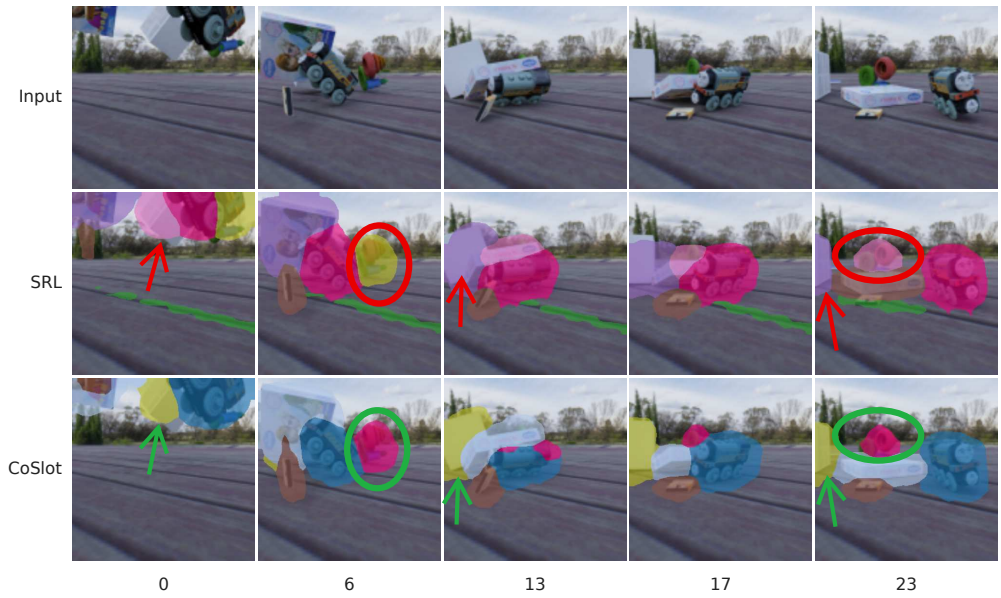

Figure F.3: MOVi-C video 004 at frames 0, 6, 13, 17, and 23. Rows show input frames and the decoder masks α from SRL and CoSlot. Arrows and circles follow a partially occluded foreground object: its displayed slot changes in SRL, while CoSlot retains the same slot color across the later frames.

F.4 MOVi-E Qualitative Results and Failure Cases

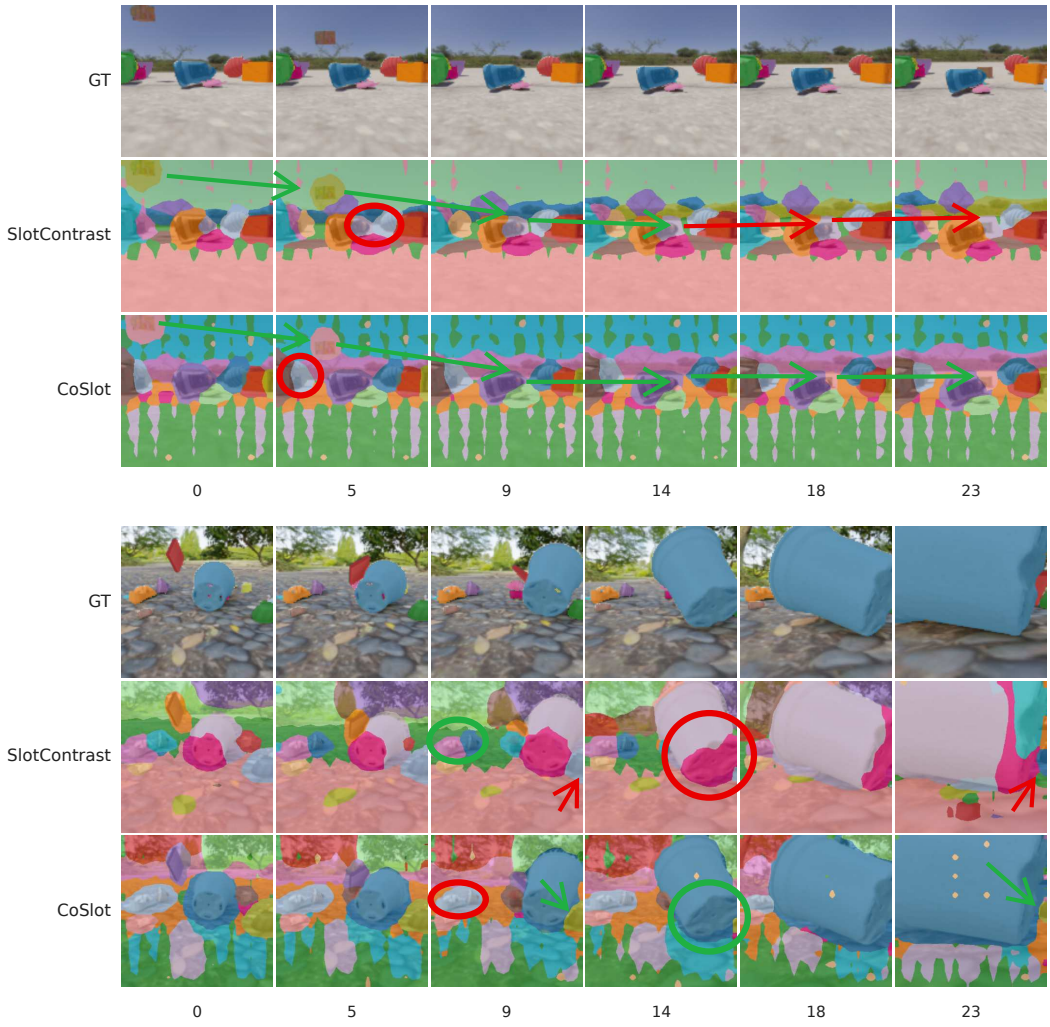

Figure F.4: Selected frames from MOVi-E videos 048 (upper) and 139 (lower). Each block shows ground truth, SlotContrast, and CoSlot. Video 048 contains several nearby small objects, and the marked regions highlight differing mask boundaries. In video 139, the marked large object stays mostly in one CoSlot mask, while some nearby small regions remain ambiguous.

These examples show that keeping a large object together does not settle the segmentation of small neighboring objects or textured backgrounds.

F.5 YouTube-VIS 2021 Qualitative Results and Failure Cases

The three real-video examples compare the same four rows: ground truth, SlotContrast, SRL, and CoSlot. Slot colors are local to each method. Circles mark regions worth inspecting, not a uniform success or failure code. These selected videos illustrate behavior under the YouTube-VIS evaluation protocol and do not establish an overall ranking.

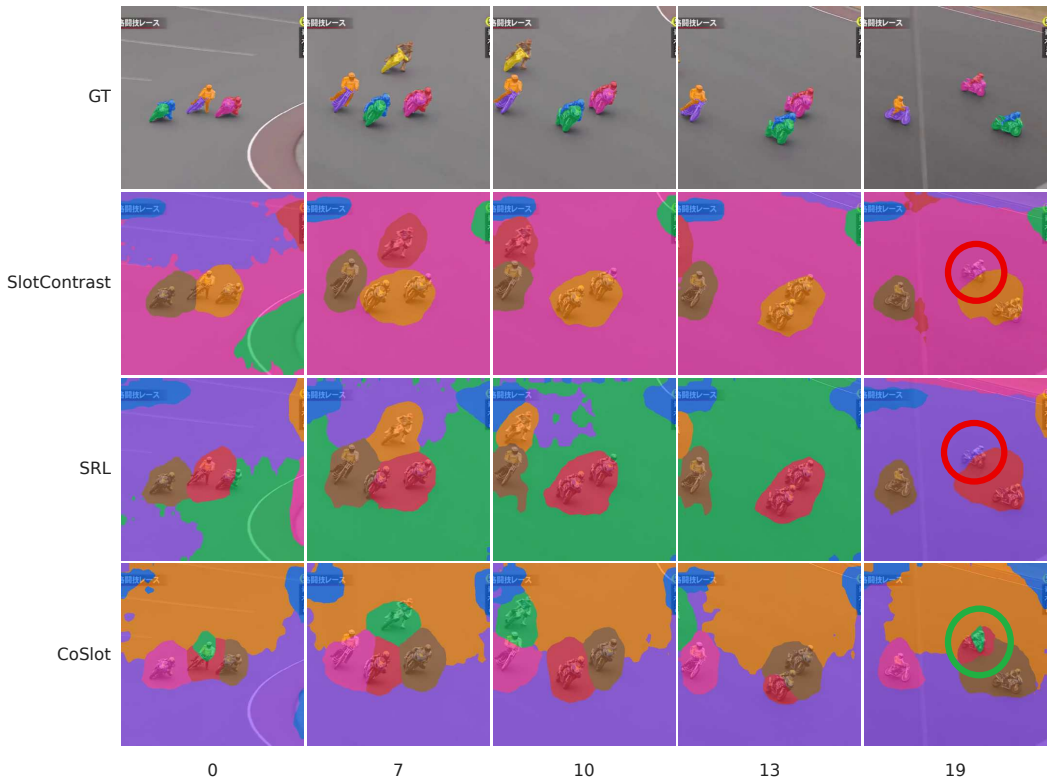

Figure F.5: YouTube-VIS 2021 video 2713, showing moving athletes across five selected frames. The marked last frame highlights a small foreground region; broad background slots are also visible in the predictions.

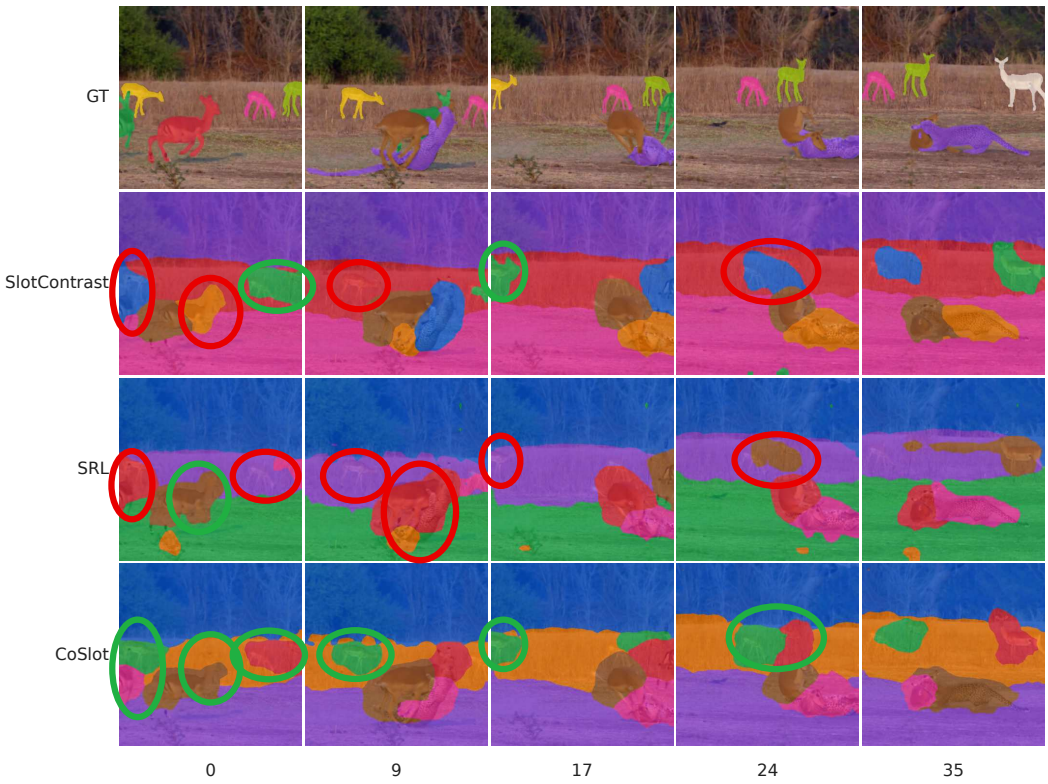

Figure F.6: YouTube-VIS 2021 video 2746, with several moving animals. The marked regions highlight differences in foreground coverage, while broad horizontal partitions of the scene remain visible in the predicted masks.

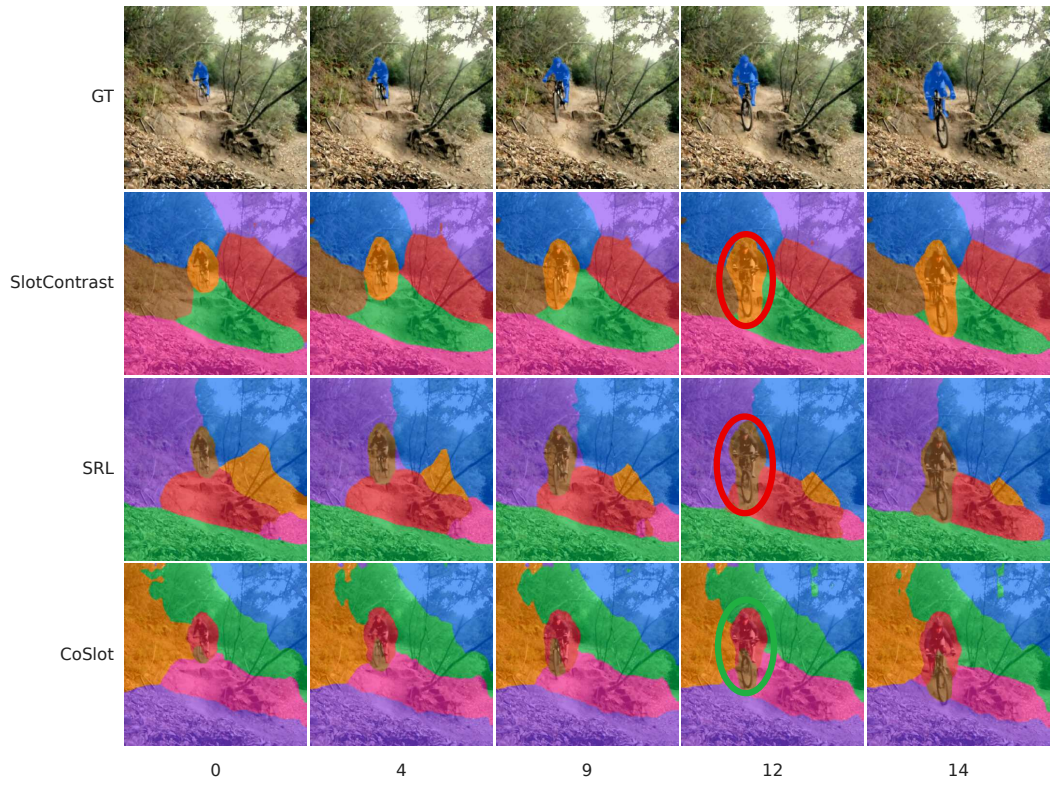

Figure F.7: YouTube-VIS 2021 video 2764, showing a cyclist on a trail. At the marked frame, CoSlot assigns the rider and bicycle to separate masks, consistent with the ground-truth distinction. The baseline rows provide a comparison of the same foreground region.