WEAK SUPERVISION HELPS JEPA

Anonymous authors
Paper under double-blind review

ABSTRACT

When a hand picks up a cup, understanding the action requires distinguishing the hand from the cup while capturing their interaction. Global video self-supervision leaves these object distinctions implicit. We show that weak object supervision improves the quality and video-data efficiency of LeVJEPA representations. Masks generated offline by a frozen SAM2 model guide a local instance-separation loss on retained patch features. The loss brings together features from the same object and separates them from other objects and the background. The original global alignment and SIGReg objectives remain unchanged. Across matched comparisons with ViT-Tiny, ViT-Small, and ViT-Large, instance supervision improves action recognition and mask propagation in every evaluated LeVJEPA setting. In our shared-head ablations, instance separation alone gives the best measured results among the tested auxiliary object-level losses. With the same ViT-Small backbone, our model trained on fewer videos exceeds the baseline trained on the larger dataset on both DAVIS and YouTube-VOS, with similar SSv2 accuracy. As an additional test, we apply the local loss to predictive V-JEPA2 and improve mask propagation while keeping its original prediction task. Both versions reuse the encoder's existing features and token budget. The auxiliary head is removed after pretraining, leaving encoder inference cost unchanged. These results establish the value of local object supervision during sparse video pretraining and show better use of student training videos with a fixed pretrained mask generator.

1 INTRODUCTION

Understanding actions requires representations that distinguish interacting objects and capture how their relationships change over time. Video JEPA methods learn from videos without action labels (Assran et al., 2025; Kuhn et al., 2026), while object membership remains implicit in their training objectives. We make this information explicit through a local instance-separation loss. Our main result is that this simple addition improves both action recognition and object correspondence during sparse video pretraining, while preserving the original JEPA objective and encoder token budget.

Our contribution concerns object supervision in encoders that learn from video frames jointly. Object-centric LeJEPA (Geusen & Konukoglu, 2026) studies image pretraining on still images from COCO. It combines patch-level instance separation with alignment and SIGReg on pooled object features, and its DAVIS evaluation tests how those image features transfer to video correspondence. We study object supervision during video pretraining itself. Local features now combine temporal context with partial observations of moving objects, and we evaluate them on both action recognition and matching across frames.

LeVJEPA makes this distinction concrete. It shares alignment and SIGReg with image LeJEPA, but uses block-causal temporal attention and drops 95% of patch tokens before the transformer (Balestriero & LeCun, 2025; Kuhn et al., 2026). Object supervision therefore acts on a small, changing set of retained patches. We use valid local pairs wherever they remain, with the original uniform token sampling. In the ViT-Tiny run on 50K videos, positive pairs remain in 9.38% of frame–view groups, yet this sparse local signal improves all four evaluation benchmarks.

Our method preserves LeVJEPA's global objective and adds instance separation to retained patch features. Masks from a frozen SAM2 model (Ravi et al., 2024) define the pairs: patches from one object are positives, while patches from other objects or background are negatives. We use a

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.

supervised contrastive loss (Khosla et al., 2020) and add it to the base objective:

$$\mathcal{L} = \mathcal{L}_{\text{LeVJEPA}} + \lambda_{\text{inst}} \mathcal{L}_{\text{inst}}. \tag{1}$$

Global view agreement and local object membership are learned from the same encoder pass. The input views and token budget stay unchanged, and the auxiliary head is removed after pretraining.

Instance supervision improves SSv2, Kinetics-400, DAVIS, and YouTube-VOS in all four matched LeVJEPA settings, from ViT-Tiny on 50K videos to ViT-Large on 1M. The fixed-backbone comparison gives a clear video-data-efficiency result: ViT-Small trained with instance supervision on 95K videos exceeds the baseline trained on 190K on both mask-propagation benchmarks. SSv2 accuracy is 29.74% versus 30.34%. These gains use the same student encoder and a fixed pretrained mask generator.

All supervised LeVJEPA pairs come from one frame and view. Even with this local signal, the learned features improve matching across frames as well as action recognition. We also test pooled-object alignment and SIGReg as separate additions to the instance head. Instance separation alone gives the highest measured scores on all four benchmarks among these shared-head variants.

Predictive V-JEPA2 provides an additional test with a different learning objective. It predicts masked features and uses neither SIGReg nor LeVJEPA's global view-alignment scheme (Assran et al., 2025). We apply the same local loss to visible tokens of its trainable encoder and keep the prediction task unchanged. Both reported comparisons improve DAVIS and YouTube-VOS, showing that the correspondence benefit extends beyond alignment and SIGReg.

Our work makes three contributions. First, we establish the value of local object supervision during joint video pretraining under 95% token dropping. Second, matched comparisons across three encoder sizes show gains in action recognition and mask propagation, and the fixed-backbone comparison gives stronger correspondence features from half as many student training videos. Third, an additional test in V-JEPA2 shows that the same local loss improves correspondence under masked-feature prediction.

2 Related Work

Self-supervised video learning. VideoMAE reconstructs masked pixels (Tong et al., 2022), while V-JEPA and V-JEPA2 predict features of masked video regions (Bardes et al., 2024; Assran et al., 2025). LeJEPA combines alignment with SIGReg (Balestriero & LeCun, 2025). LeVJEPA uses this objective for video with random token dropping and block-causal attention (Kuhn et al., 2026). We add local object supervision while preserving each video model's base objective.

Object masks and contrastive supervision. DenseCL learns local correspondences between image views (Wang et al., 2021), and DetCon uses segmentation regions for contrastive learning (Hénaff et al., 2021). SAMFeat uses SAM-derived groups to supervise local descriptors (Wu et al., 2023). MASA contrasts object proposals across augmented views to learn instance associations (Li et al., 2024). Our loss follows supervised contrastive learning (Khosla et al., 2020), with object masks defining positive and negative pairs.

Object-centric LeJEPA combines instance separation with alignment and SIGReg on pooled object features for image pretraining (Geusen & Konukoglu, 2026). It also tests instance separation alone. We test joint video pretraining in two settings: sparse LeVJEPA and predictive V-JEPA2. The main method keeps the video objective and adds only local instance separation. Our shared-head ablations then test the value of adding pooled-object alignment or SIGReg in this video setting.

Dense video features. Label propagation across frames tests the correspondence information in local features (Jabri et al., 2020). V-JEPA 2.1 improves dense features through prediction losses on both visible and masked tokens, together with other training changes (Mur-Labadia et al., 2026). Our additional signal specifies object membership using a pretrained mask generator. We test its effect with a shared mask-propagation procedure on DAVIS and YouTube-VOS, alongside action-recognition probes.

3 METHOD

We add one local instance-separation loss to the original video JEPA objective. Frozen object masks label the patches that reach the encoder. The loss acts on retained tokens in LeVJEPA and visible context tokens in V-JEPA2.

3.1 OBJECT LABELS FROM FROZEN MASKS

We generate masks offline with a pretrained SAM2 model (Ravi et al., 2024). During pretraining, we select the masks for the sampled video frames and apply the same crop and flip to RGB frames and masks. Object IDs stay fixed. Gradients update the video encoder and auxiliary head; SAM2 provides frozen pseudo-labels.

Coverage is the fraction of a patch region covered by an object mask. A patch receives an object label when exactly one mask covers at least half of its region. It receives a background label when no mask reaches this threshold. Patches with unknown coverage or multiple qualifying masks are excluded from the instance loss. The region is one spatial patch in LeVJEPA and a two-frame tubelet in V-JEPA2.

After the encoder selects its tokens, we form local groups. A LeVJEPA group contains retained patches from one frame and one view. A V-JEPA2 group contains visible tokens from one two-frame temporal position and one context mask. We select at most 32 patches per object within each group, keeping those with the highest mask coverage when needed. This selection affects only the auxiliary loss.

3.2 LOCAL INSTANCE SEPARATION

Let h_i be the encoder feature of a visible patch. A small head g maps it to a 128-dimensional vector. The head contains layer normalization, a linear layer, GELU, and a second linear layer. Its hidden size equals the encoder feature size. We normalize its output:

$$z_i = \frac{g(h_i)}{\|g(h_i)\|_2}. \tag{2}$$

Let $A(i)$ contain all other selected, valid patches in the same local group, including background. For a foreground anchor i, $P(i) \subseteq A(i)$ contains patches with the same object label. We use the supervised contrastive loss (Khosla et al., 2020):

$$\ell_i = -\frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i^\top z_p/\tau)}{\sum_{a \in A(i)} \exp(z_i^\top z_a/\tau)}. \tag{3}$$

Patches from the same object are positives. Patches from other objects and background are negatives. An anchor contributes only when it has at least one positive. Background patches appear in the denominator but are never anchors.

We first average anchor losses within each eligible foreground object. We then average objects within each group, groups within each video, and videos with valid supervision. An object is eligible when it has at least two selected patches in the group. Each later average includes entries with at least one eligible contribution. This gives eligible objects equal weight within a group. The main experiments use $\tau = 0.1$ and $\lambda_{\text{inst}} = 0.06$.

3.3 INSTANCE SUPERVISION IN LEVJEPA

LeVJEPA learns through alignment of global view representations and SIGReg regularization (Kuhn et al., 2026). We keep both terms unchanged. The base head maps each view's class token to a global embedding. Alignment brings embeddings from the same video together, while SIGReg controls their distribution. The encoder uses block-causal temporal attention and drops 95% of patch tokens before the transformer. We retain the same uniform token sampling and augmented views.

The instance head processes retained patch features from global and local views. Pairs stay within one frame and one view, so the loss uses the object relationships available after token dropping. The

full objective is

$$\mathcal{L}_{\text{LeVJEPA+inst}} = \mathcal{L}_{\text{LeVJEPA}} + \lambda_{\text{inst}}\mathcal{L}_{\text{inst}}. \tag{4}$$

The base loss acts on global view representations, and the instance loss acts on local patch relationships. Instance separation is the sole auxiliary object-level term in the main method. Separate ablations add alignment or SIGReg on pooled object features from the instance head. Both models in each pair are evaluated with the final frozen online encoder.

3.4 Instance supervision in V-JEPA2

V-JEPA2 predicts features of masked video regions (Assran et al., 2025). A trainable context encoder processes visible tokens. A predictor uses them to estimate features from a target encoder, whose weights follow an exponential moving average of the context encoder.

We attach the instance head to visible features of the trainable context encoder. The context and target masks, prediction loss, predictor, and target update stay unchanged:

$$\mathcal{L}_{\text{V-JEPA2+inst}} = \mathcal{L}_{\text{pred}} + \lambda_{\text{inst}}\mathcal{L}_{\text{inst}}, \tag{5}$$

where $\mathcal{L}_{\text{pred}}$ is the original masked-feature prediction loss. The instance term updates the context encoder and its auxiliary head. The target encoder receives these changes through the moving average.

Each V-JEPA2 token covers two frames and a 16×16 spatial patch. We average each track's mask coverage over this $2 \times 16 \times 16$ region. Both frames must have stored SAM2 masks. Pairs use visible tokens at the same two-frame temporal position and within the same context mask. The supervision thus follows the model's token grid and remains local. We evaluate the frozen target encoder for both the baseline and instance model.

3.5 Training and inference cost

Both versions reuse features from the original encoder pass and keep the encoder architecture and token budget unchanged. Training adds the auxiliary head and loss computation. SAM2 mask generation is an offline cost. After pretraining, we remove the auxiliary head. Evaluation uses the encoder without SAM2 masks or additional encoder computation.

4 Experiments

Our main experiments test local object supervision in LeVJEPA. We compare matched models across four pretraining settings, measure video-data efficiency with a fixed backbone, and test the choice of object-level losses. Additional V-JEPA2 comparisons on 50K and 200K videos test the same local loss with masked-feature prediction.

4.1 Training and evaluation

We evaluate frozen encoders on action recognition and object matching across frames. For SSv2, we train a one-block attentive classifier on the frozen features. For Kinetics-400 (K400), we average the features and train a linear classifier. Both probes use 20 training epochs and an effective batch size of 256. We report final Top-1 accuracy.

For LeVJEPA, we use two K400 feature readouts. Raw trains the classifier directly on mean-pooled features. Std. first standardizes each feature coordinate using the mean and standard deviation estimated from K400 training videos. These statistics stay fixed during probe training. We use Raw at 50K and 100K, both Raw and Std. at 200K, and Std. at 1M. Each matched pair uses the same readout. The V-JEPA2 comparisons use Raw. Appendix A.5 gives the full probe settings. Appendix A.1 describes the pretraining data, Appendices A.3 and A.4 give the training settings, and Appendix A.2 describes mask generation.

For DAVIS and YouTube-VOS, we transfer masks across frames using feature similarity. Each object starts from its first supplied annotation. Every matched pair uses the same propagation and scoring procedure, without a trained segmentation head or SAM2 at evaluation time. DAVIS uses

the official evaluation code and object-level averaging, with 30 validation videos and 61 objects. YouTube-VOS uses our local scoring protocol on the 2019 validation split, with 507 videos and 1,063 objects. Appendix A.6 gives both protocols.

On DAVIS, $\mathcal{J}$ measures mask overlap and $\mathcal{F}$ measures boundary quality. We average each score over evaluated frames within an object, then over objects. The final $\mathcal{J}\&\mathcal{F}$ score is the mean of the two scores. YouTube-VOS reports both scores for seen and unseen categories. Overall is the mean of these four scores.

Within each pair, the baseline and instance model use the same backbone and pretraining data. The LeVJEPA pairs also share the base objective, optimizer, views, token budget, and update count. LeVJEPA and V-JEPA2 keep their own input settings and base objectives. We evaluate the final online encoder for both LeVJEPA models and the final target encoder for both V-JEPA2 models.

LeVJEPA uses ViT-Tiny at 50K, ViT-Small at 100K and 200K, and ViT-Large at 1M. V-JEPA2 uses ViT-Tiny at 50K and ViT-Small at 200K. The 100K and 200K names include pretraining validation splits; the training sets contain 95K and 190K videos. The 50K name refers to 50K training videos. Backbone size, data mixture, and training budget vary across settings. The matched pairs test the benefit of instance supervision within each setting.

Pretraining uses seed 7, with one run per configuration. Scores are on a 0–100 scale; higher is better. Percentage-point differences reported in the text are calculated before rounding. Bold marks the better result within each matched pair, separately for each metric.

4.2 Instance supervision improves every LeVJEPA setting

Instance supervision improves action recognition and mask propagation in all four matched LeVJEPA settings. The gains span ViT-Tiny, ViT-Small, and ViT-Large, with higher scores on every benchmark. Tables 1–4 report each model and data setting separately.

Figure 1 summarizes the SSv2 results across all four pretraining settings.

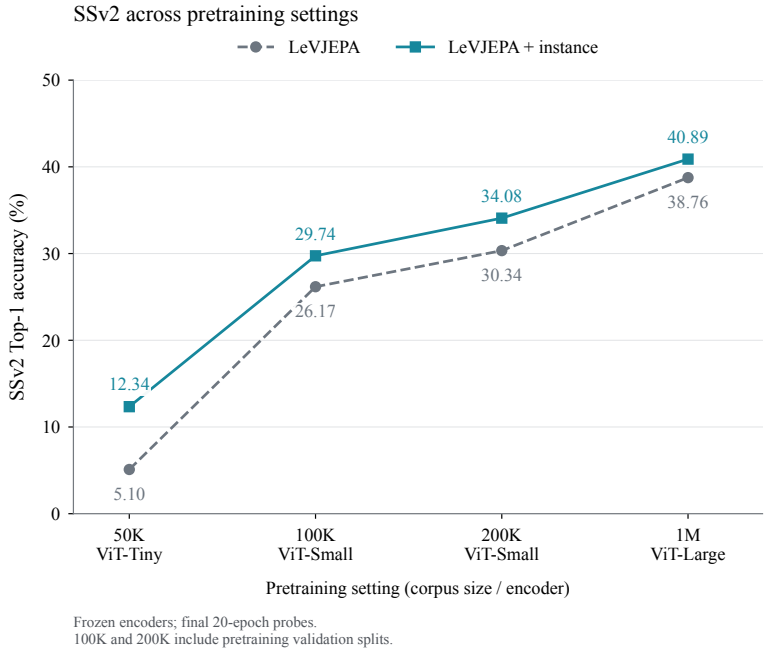

Figure 1: SSv2 Top-1 accuracy across four pretraining settings. LeVJEPA + instance achieves higher accuracy in every matched pair. Each pair uses the same encoder size and pretraining data. Scores come from the final epoch of a 20-epoch probe on frozen features.

50K videos. With the same 50K training videos, LeVJEPA + instance improves SSv2 by 7.24 points and K400 by 5.66 points. DAVIS rises by 13.11 points and YouTube-VOS by 7.70 (Table 1). The gains cover both action recognition and object matching in this small-data setting.

Table 1: LeVJEPA with ViT-Tiny on 50K training videos. K400 uses Raw features. Bold marks the higher score within the matched pair.

Method	SSv2 Top-1	K400 Top-1	DAVIS $\mathcal{J}\&\mathcal{F}$	YouTube-VOS Overall
LeVJEPA	5.10	4.05	31.92	39.31
LeVJEPA + instance	**12.34**	**9.71**	**45.02**	**47.01**

Figure 2 illustrates patch similarity within an object in a selected DAVIS frame. This qualitative example comes from a separate ViT-Tiny / 200K experiment.

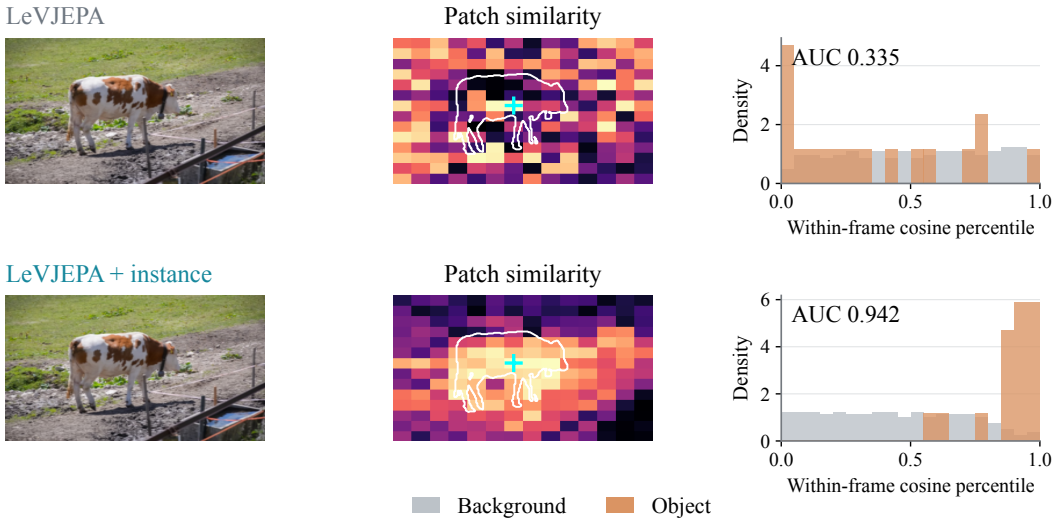

Figure 2: Patch similarity in a selected DAVIS frame from an earlier, separate ViT-Tiny / 200K experiment. The cyan cross marks the query patch, and the white line shows the object boundary. Heatmaps show within-frame cosine-similarity percentiles. Histograms compare object and background patches. The displayed AUC measures their separation for this query.

100K videos. The gains extend to ViT-Small on the 100K set (Table 2). Instance supervision raises DAVIS from 33.25 to 45.68 and YouTube-VOS from 37.69 to 46.94, gains of 12.44 and 9.24 points. Action recognition also improves: SSv2 rises from 26.17 to 29.74 and K400 from 15.99 to 19.81, gains of 3.57 and 3.82 points.

Table 2: Main LeVJEPA comparison on the 100K set: 95K training and 5K pretraining validation videos. K400 uses Raw features. Bold marks the higher score within the matched pair.

Method	SSv2 Top-1	K400 Top-1	DAVIS $\mathcal{J}\&\mathcal{F}$	YouTube-VOS Overall
LeVJEPA	26.17	15.99	33.25	37.69
LeVJEPA + instance	**29.74**	**19.81**	**45.68**	**46.94**

200K videos. At 200K, instance supervision again improves every benchmark (Table 3). The gains are 3.75 points on SSv2, 3.62 on K400 Raw, 11.20 on DAVIS, and 8.98 on YouTube-VOS. K400 also improves after feature standardization: accuracy rises from 24.80 to 26.14, a gain of 1.33 points. Thus, both K400 readouts show a positive effect in this setting.

Table 3: Main LeVJEPA comparison on the 200K set: 190K training and 10K pretraining validation videos. Both models use ViT-Small and the same data. K400 is evaluated with both Raw and train-only standardized (Std.) features.

Method	SSv2 Top-1	K400 Raw Top-1	K400 Std. Top-1	DAVIS $\mathcal{J}\&\mathcal{F}$	YouTube-VOS Overall
LeVJEPA	30.34	18.00	24.80	36.20	39.06
LeVJEPA + instance	**34.08**	**21.61**	**26.14**	**47.41**	**48.04**

1M videos. The advantage remains with ViT-Large on 1M videos (Table 4). Instance supervision raises SSv2 from 38.76 to 40.89, DAVIS from 31.29 to 37.66, and YouTube-VOS from 33.62 to 39.93. Standardized K400 rises from 29.56 to 32.21, a gain of 2.65 points. All four benchmarks therefore improve in the largest tested setting.

Table 4: Main LeVJEPA comparison on the 1M-video set with ViT-Large. K400 uses train-only standardized (Std.) features. Bold marks the higher score within the matched pair.

Method	SSv2 Top-1	K400 Std. Top-1	DAVIS $\mathcal{J}\&\mathcal{F}$	YouTube-VOS Overall
LeVJEPA	38.76	29.56	31.29	33.62
LeVJEPA + instance	**40.89**	**32.21**	**37.66**	**39.93**

4.3 Data efficiency with a fixed backbone

The 100K and 200K experiments use the same ViT-Small backbone. The smaller set is sampled from the larger one while preserving the share of each data source. It contains 95K training videos and 5K pretraining validation videos, compared with 190K and 10K in the larger set.
With half as many student training videos and about half the updates, LeVJEPA + instance exceeds the 190K baseline on both mask-propagation benchmarks (Table 5). It reaches 45.68 on DAVIS and 46.94 on YouTube-VOS, compared with 36.20 and 39.06 for the baseline. K400 Raw is also higher. SSv2 accuracy is 29.74% versus 30.34%, a difference of 0.60 points. This comparison shows better use of student pretraining videos with the same backbone and a fixed pretrained mask generator. Using the full 200K set improves all four instance-model scores further.

Table 5: Data-size comparison with ViT-Small. Training counts exclude validation videos. All runs use a batch size of 1,024 and about 100 passes over the training data. Each block compares the same baseline with one instance model; bold marks the higher score within that block.

Method	Train videos	Updates	SSv2	K400 Raw	DAVIS	YouTube-VOS
LeVJEPA	190,000	18,555	**30.34**	18.00	36.20	39.06
LeVJEPA + instance	95,000	9,278	29.74	**19.81**	**45.68**	**46.94**
LeVJEPA	190,000	18,555	30.34	18.00	36.20	39.06
LeVJEPA + instance	190,000	18,555	**34.08**	**21.61**	**47.41**	**48.04**

4.4 ADDING POOLED-OBJECT LOSSES TO THE INSTANCE HEAD

With ViT-Tiny on 50K videos, we test two additions to the instance head: alignment and SIGReg on pooled object features. Each variant keeps the base LeVJEPA objective and instance loss. The extra term uses the same projection space as instance separation.

Instance separation alone gives the highest measured score on every benchmark among these shared-head variants (Table 6). The best tested combination therefore keeps LeVJEPA's global objective and uses local instance separation as its only auxiliary object-level loss.

The added alignment and SIGReg terms have weights of 1.0 and 0.02, respectively. The instance weight stays at 0.06. Appendix A.7 gives the pooling and loss settings. Paired bootstrap intervals for the dense-score advantage over added object alignment are positive on both benchmarks (Table 13). The intervals for the differences from added object SIGReg include zero.

Table 6: Additional pooled-object losses on the shared instance head, using ViT-Tiny and 50K videos. All models keep the global LeVJEPA loss. Bold marks the highest score in each column.

Auxiliary objective	SSv2 Top-1	K400 Top-1	DAVIS $\mathcal{J}\&\mathcal{F}$	YouTube-VOS Overall
None	5.10	4.05	31.92	39.31
Instance separation	**12.34**	**9.71**	**45.02**	**47.01**
Instance + object alignment	10.38	7.85	42.55	45.78
Instance + object SIGReg	8.57	8.89	43.97	46.67

4.5 ADDITIONAL EXPERIMENT: TRANSFER TO V-JEPA2

We test the same local loss in V-JEPA2 on 50K and 200K videos. Instance supervision acts on visible features of the trainable encoder, while V-JEPA2 keeps its original masked-feature prediction loss.

Both settings show large gains in mask propagation. At 50K, DAVIS improves by 17.45 points and YouTube-VOS by 12.37. At 200K, the gains are 13.83 and 7.72 points. Local instance supervision therefore improves object correspondence under predictive JEPA training as well as under alignment and SIGReg.

Action recognition improves by 12.00 points on SSv2 and 6.92 on K400 at 50K. At 200K, the measured gains are 0.71 and 3.22 points. The largest gains in both predictive-model comparisons are in object correspondence.

Table 7: Additional V-JEPA2 comparison on 50K videos. Bold marks the higher score within the matched pair.

Method	SSv2 Top-1	K400 Top-1	DAVIS $\mathcal{J}\&\mathcal{F}$	YouTube-VOS Overall
V-JEPA2	13.34	4.54	24.21	27.07
V-JEPA2 + instance	**25.34**	**11.46**	**41.66**	**39.43**

Table 8: Additional V-JEPA2 comparison on 200K videos. Bold marks the higher score within the matched pair.

Method	SSv2 Top-1	K400 Top-1	DAVIS $\mathcal{J}\&\mathcal{F}$	YouTube-VOS Overall
V-JEPA2	40.60	22.74	39.23	38.95
V-JEPA2 + instance	**41.30**	**25.95**	**53.07**	**46.67**

Appendix B gives detailed mask scores and SSv2 scores during probe training for each data size. The encoder stays frozen throughout these probe evaluations.

Published reference points. Table 9 places the study alongside published image and video results, each under its source paper's protocol. Object-centric LeJEPA averages DAVIS scores over videos and uses eight memory frames with temperature 0.1. Our protocol averages over objects and uses fifteen memory frames with temperature 0.2. YT VOS-S holds out 20% of YouTube-VOS training videos; our local scores use the 2019 validation split. Our matched pairs measure the gain from instance supervision with the evaluation procedure fixed.

Table 9: Published dense-feature results on a 0–100 scale. Each block uses the cited paper's evaluation protocol. YT VOS-S denotes the split used in the V-JEPA 2.1 paper; a dash means that no score is given in the cited table.

Method	Backbone	Pretraining data	DAVIS	YT VOS-S
Object-centric LeJEPA, Table 4 (Geusen & Konukoglu, 2026)				
Image LeJEPA	ViT-B/16	COCO images	61.3	—
Object-centric LeJEPA	ViT-B/16	COCO images	68.2	—
V-JEPA 2.1, Table 8 (Mur-Labadia et al., 2026)				
DINOv2	ViT-g/14	LVD-142M	63.9	65.6
V-JEPA2	ViT-g/16	VideoMix22M	52.5	53.7
V-JEPA 2.1	ViT-g/16	VisionMix163M	68.1	72.3

5 Discussion

Local object supervision improves both action recognition and mask propagation in every matched LeVJEPA setting, while the encoder keeps its global objective and observes only 5% of patch tokens. The additional V-JEPA2 comparisons show that the correspondence benefit also holds under masked-feature prediction. This gives the method a clear role: adding object membership to video training through the encoder's existing local features.

With ViT-Small, instance supervision gives stronger correspondence features from half as many student training videos and about half the updates. These gains use a fixed pretrained mask generator. Training adds a local loss and auxiliary head; the encoder architecture and inference cost stay unchanged.

6 Conclusion

We show that local instance supervision strengthens sparse video JEPA training. It improves action recognition and mask propagation in all four matched LeVJEPA settings, across three encoder sizes. A fixed-backbone comparison gives stronger correspondence features from half as many student training videos, and the tested shared-head variants perform best with instance separation as the only auxiliary object-level loss. Additional V-JEPA2 comparisons extend the correspondence result to masked-feature prediction. The method preserves each model's base objective and encoder token budget and adds no encoder inference cost.

7 Limitations

Our evaluation focuses on local object supervision in the studied LeVJEPA and V-JEPA2 configurations, using frozen-encoder action recognition and mask propagation. Extending this analysis to longer temporal contexts and additional downstream tasks is a natural direction for future work. We use a fixed pretrained SAM2.1 model throughout, keeping object supervision consistent across experiments; alternative mask generators remain to be explored. The video-data-efficiency results concern student pretraining with this fixed teacher, whose prior training and offline mask generation are separate resources. The auxiliary head and mask generator are unnecessary at encoder inference.

Ethics and Societal Impact Statement

This work aims to improve video representation learning, with potential uses in action recognition and object tracking. Applications that process videos of people should protect privacy and follow the terms of the source datasets. Responsible use also requires checking model accuracy and bias in the intended setting.

Reproducibility Statement

Appendix A describes the data sources and splits, SAM2 checkpoint and mask generation, model and optimizer settings, training budgets, and random seeds. It also specifies the frozen-encoder probes, mask propagation, scoring procedures, and bootstrap analysis. Appendix B reports detailed benchmark scores and probe learning curves. These descriptions make the training and evaluation settings behind the reported comparisons explicit.

AI Use Statement

Generative AI tools assisted with literature search and text polishing. The authors are responsible for the experiments and the final manuscript.

References

Mido Assran et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. URL https://arxiv.org/abs/2511.08544.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024. URL https://arxiv.org/abs/2404.08471.

Jakob Geusen and Ender Konukoglu. Object-centric LeJEPA. *arXiv preprint arXiv:2607.02404*, 2026. URL https://arxiv.org/abs/2607.02404.

Olivier J. Hénaff et al. Efficient visual pretraining with contrastive detection. *arXiv preprint arXiv:2103.10957*, 2021. URL https://arxiv.org/abs/2103.10957.

Allan Jabri, Andrew Owens, and Alexei A. Efros. Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*, 2020. URL https://arxiv.org/abs/2006.14613.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020. URL https://arxiv.org/abs/2004.11362.

Lukas Kuhn, Lucas Maes, Giuseppe Serra, Quentin Le Lidec, Yann LeCun, Randall Balestriero, and Florian Buettner. LeVJEPA: Efficient & scalable video pretraining without the heuristics. *arXiv preprint arXiv:2608.27395*, 2026. URL https://arxiv.org/abs/2608.27395.

Siyuan Li, Lei Ke, Martin Danelljan, Luigi Piccinelli, Mattia Segu, Luc Van Gool, and Fisher Yu. Matching anything by segmenting anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2406.04221.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026. URL https://arxiv.org/abs/2603.14482.

Nikhila Ravi et al. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL https://arxiv.org/abs/2408.00714.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*, 2022. URL https://arxiv.org/abs/2203.12602.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/papers/Wang_Dense_Contrastive_Learning_for_Self-Supervised_Visual_Pre-Training_CVPR_2021_paper.pdf.

Jingqian Wu, Rongtao Xu, Zach Wood-Doughty, Changwei Wang, Shibiao Xu, and Edmund Y. Lam. Segment anything model is a good teacher for local feature learning. *arXiv preprint arXiv:2309.16992*, 2023. URL https://arxiv.org/abs/2309.16992.

A Implementation and Evaluation Details

A.1 Pretraining data and splits

The 50K setting uses 50,000 videos from the official SSv2 training split. We select them with seed 7 after checking that each video contains at least 16 readable frames. Another 2,500 videos from the same official training split are used for pretraining monitoring. Action recognition is evaluated on the official SSv2 validation set.

The larger settings mix Kinetics-400/600/700-2020, PE Video, SSv2, and Walking Tours. We select videos using identifiers, source quotas, and hashes set by a random seed, without action labels. The 1M corpus contains 418,667 Kinetics clips, 525,526 PE Video clips, 52,615 SSv2 videos, and 3,192 Walking Tours episodes. Each Walking Tours episode is a 128-frame segment from one of ten recordings.

We merge repeated Kinetics clip identifiers across releases, keeping the newer release. RGB preparation checks readability and fills each source quota with readable clips before fixing the video list. The selection seed is 20260910.

The 200K corpus is a subset of the 1M corpus with a higher share of SSv2. It keeps all 52,615 SSv2 videos and selects videos from the other sources with seed 20260912. We split it into 190,000 training videos and 10,000 pretraining validation videos, preserving the source proportions. The 100K corpus is a nested subset. Seed 20260920 selects half of each parent split, again preserving source proportions. Its 95,000 training videos come from the 190,000-video training set, and its 5,000 validation videos come from the parent validation set. Table 10 gives the exact training counts.

Table 10: Source counts in the encoder-training splits. The 100K and 200K names also include their pretraining validation splits. Walking Tours is counted in 128-frame episodes.

Setting	Kinetics	PE Video	SSv2	Walking Tours	Total
50K	0	0	50,000	0	50,000
100K	30,938	38,834	24,992	236	95,000
200K	61,876	77,668	49,984	472	190,000
1M	418,667	525,526	52,615	3,192	1,000,000

SSv2 accounts for about 26.3% of the 100K and 200K corpora and 5.3% of the 1M corpus. The nested 95K/190K comparison keeps the mixture and backbone fixed. The two methods within each

matched comparison use the same source videos and sampling settings. Walking Tours uses episode-level pretraining splits. All reported evaluations use the final scheduled checkpoint.

RGB preparation. For Kinetics and PE Video, we prepare clips from the start of each source clip at 8 frames per second. Each clip has up to 48 frames and a short image side of 320 pixels. Selected clips contain at least 16 readable frames. SSv2 uses its released RGB videos. Walking Tours uses the stored 128-frame episodes and frame indices. These frames are resized to a short side of 384 pixels before cropping. RGB availability defines which frames can be sampled; stored masks define which frames can provide object supervision. In the mixed-data comparisons, a shared RGB sampling reference keeps temporal sampling the same for the baseline and instance model.

A.2 Frozen SAM2 mask generation

We use the released SAM2.1 Hiera Base Plus checkpoint, `sam2.1_hiera_base_plus.pt`, with model configuration `sam2.1_hiera_b+.yaml` (Ravi et al., 2024). We use the released weights unchanged and keep the teacher frozen. The SAM2 paper reports SA-V manual, Internal, and SA-1B as the training data for the released model and identifies the revised release as SAM2.1 (Ravi et al., 2024, Appendix D.2.2). The implementation is pinned to SAM2 revision `2b90b9f5ceec907a1c18123530e92e794ad901a4`.

Automatic mask proposals are generated on the first frame of each annotated clip. We use a 32×32 point grid, 64 points per batch, a predicted-IoU threshold of 0.8, and a stability threshold of 0.95 with offset 1.0. Box non-maximum suppression uses an IoU threshold of 0.7. We use no extra crop layers, enable mask-to-mask refinement, and disable the model builder's postprocessing option.

We retain proposals covering between 0.2% and 80% of the image and rank them by predicted IoU times stability score. In this order, we reject a proposal if its mask IoU with an accepted proposal is at least 0.85. We keep at most 16 objects. Each accepted mask starts a SAM2 video track, which is propagated through the annotated clip with a logit threshold of zero. Track IDs stay fixed within a clip; they are not matched across videos. This process uses neither manual prompts nor action labels.

We store masks for each decoded frame in the annotated range: up to 48 frames for SSv2, Kinetics, and PE Video, and 128 frames for a Walking Tours episode. RGB frames and masks receive the same spatial transforms. Track IDs identify the same object across the two frames of a V-JEPA2 tubelet. The assignment rules are given in Section 3. All teacher inference takes place offline, and masks remain fixed pseudo-labels. Table 10 counts student pretraining videos. All instance models use the same fixed pretrained SAM2 teacher.

A.3 LeVJEPA training settings

Table 11 lists the training budgets. One update is one optimizer step after gradient accumulation. ViT-Tiny has 12 blocks, width 192, and 3 attention heads. ViT-Small has 12 blocks, width 384, and 6 heads. ViT-Large has 24 blocks, width 1,024, and 16 heads. All LeVJEPA models use 16×16 spatial patches, one-frame temporal tokens, and 16-frame input clips with stride 1. Block-causal attention lets a token attend to its own frame and earlier frames. Uniform dropping removes 95% of patch tokens before the transformer. Object masks do not affect this sampling.

Table 11: LeVJEPA settings shared by each baseline/instance pair. Batch size is the global number of clips per optimizer update.

Setting	Backbone	Batch	Updates	Warmup updates	Seed
50K	ViT-Tiny	128	12,500	371	7
100K	ViT-Small	1,024	9,278	1,114	7
200K	ViT-Small	1,024	18,555	2,227	7
1M	ViT-Large	3,072	32,553	3,906	7

Each clip produces one global view at 224×224 and ten local views at 96×96. Random crop areas are $[0.8, 1.0]$ for the global view and $[0.02, 0.4]$ for local views; aspect ratios are $[3/4, 4/3]$.

We apply horizontal flipping with probability 0.5, color jitter with probability 0.8, and grayscale conversion with probability 0.2. Jitter strengths for brightness, contrast, saturation, and hue are 0.4, 0.4, 0.2, and 0.1. We normalize RGB values with ImageNet mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$.

We use AdamW with learning rate 4×10^{-4}, weight decay 0.04, $\beta = (0.9, 0.95)$, and $\epsilon = 10^{-8}$. The learning rate rises linearly from 10^{-4} during warmup and then stays constant. Training uses BF16 mixed precision without gradient clipping. The global projection head has hidden width 2,048 and output width 256. SIGReg uses weight 0.02, 1,024 random projections, and 17 quadrature knots. The instance head has hidden width equal to the encoder width and output width 128. Its weight is 0.06, temperature is 0.1, mask-coverage threshold is 0.5, and cap is 32 patches per object within each local group. These instance settings stay fixed across model sizes. The baseline omits the instance loss. Evaluation uses the final online encoder and removes both projection heads.

Positive-pair availability. The 50K ViT-Tiny instance run retains same-object positive pairs in an average of 16.51 frame–view groups per sampled video, despite dropping 95% of tokens. This count comes from training logs over all 12,500 updates. Each video has 176 possible groups: 16 frames in each of 11 views. Positive pairs remain in 9.38% of these groups. The average number of eligible object–frame–view groups is 17.68 per video. These counts describe positive-pair availability after token dropping and mask assignment.

A.4 V-JEPA2 TRAINING SETTINGS

The additional V-JEPA2 comparisons use ViT-Tiny on 50K and ViT-Small on 200K, with the same source cohorts as LeVJEPA. The global batches and update budgets are 128 and 12,500 for Tiny, and 1,024 and 18,555 for Small. The encoder uses 16 frames, 224×224 crops, 16×16 spatial patches, two-frame tubelets, and rotary positional embeddings. Random crop areas are $[0.3, 1.0]$ with aspect ratios $[0.75, 1.35]$.

We keep the two base masking recipes. One uses eight blocks with spatial scale 0.15; the other uses two blocks with spatial scale 0.7. Both use aspect ratios $[0.75, 1.5]$ and span the full temporal input. The predictor has 12 blocks. Its width and head count are 64 and 4 for Tiny, and 384 and 6 for Small. The prediction task uses an L_1 feature loss.

AdamW uses learning rate 5.25×10^{-4}, weight decay 0.04, $\beta = (0.9, 0.999)$, and $\epsilon = 10^{-8}$. Warmup starts at 10^{-4} and lasts 371 updates for Tiny and 2,227 for Small. The learning rate then stays constant. The target encoder uses EMA momentum 0.99925. Training uses BF16 and seed 7. The instance head acts only on visible context-encoder tokens, with the same width, weight, temperature, coverage threshold, and patch cap as in LeVJEPA. The target branch receives no instance loss. We evaluate the final EMA target encoder, following the predictive model's evaluation path.

A.5 ACTION-RECOGNITION PROBES

All classification probes use the frozen encoder's final normalized patch features. We train a fresh classifier for 20 epochs and report the last epoch. The protocol uses seed 0, an effective batch size of 256, and AdamW with learning rate 10^{-3}, $\beta = (0.9, 0.999)$, and weight decay 0.01. The learning rate follows cosine decay to zero without warmup. We use FP16 autocast and gradient scaling. Both methods start with the same learning rate and use the same weight decay.

Inputs contain eight frames with temporal stride 4 and spatial resolution 224×224. Training uses random resized crops with area $[0.08, 1.0]$ and aspect ratio $[3/4, 4/3]$, the `rand-m7-n4-mstd0.5-inc1` augmentation policy, and random erasing with probability 0.25. Horizontal flipping is disabled. The three-view evaluation transform resizes the short side to 224 pixels and takes three spatial crops. All probes use the same RGB normalization as the encoder input above.

SSv2. We use a one-block attentive classifier with 16 attention heads and 174 output classes. Training uses two temporal clips and one spatial view. Evaluation uses two temporal clips and three spatial views. Tokens from the temporal clips are combined within each spatial view; predicted class probabilities are averaged across spatial views. No new positional embeddings are added by the probe.

Kinetics-400. We mean-pool the patch and temporal features and fit an affine 400-class classifier. Training uses one temporal clip and one spatial view; evaluation uses eight temporal clips and three spatial views. We report the readout type explicitly. *Raw* fits the classifier on the pooled features. *Std.* first applies fixed coordinate-wise standardization:

$$\widetilde{h}_d = \frac{\overline{h}_d - \mu_d}{\max(\sigma_d, 10^{-6})}. \tag{6}$$

For each encoder, we estimate μ_d and σ_d once from K400 training videos. Each readable video provides one deterministic temporal clip and one center view. We accumulate moments in FP64 and skip unreadable videos without replacement. Calibration requires at least 95% of the videos in the training list and records the actual count. The statistics stay fixed while a fresh classifier is trained.

Raw and Std. use the same loader settings and 20-epoch probe budget. Evaluation processes 19,872 entries from a 19,881-entry validation list; the incomplete final batch is excluded. The loader allows unreadable entries to be replaced by other videos. Table 3 reports both readouts at 200K.

Late probe epochs. Table 12 reports selected late epochs of the standardized 200K probes. The instance model has higher accuracy at each listed epoch.

Table 12: K400 Std. Top-1 at late epochs of the same completed 200K probes. The encoder stays frozen.

Method	Epoch 15	Epoch 18	Epoch 19	Epoch 20
LeVJEPA	24.26	24.69	24.81	24.80
LeVJEPA + instance	**25.78**	**26.16**	**26.14**	**26.14**

A.6 Mask propagation with frozen features

We use the same propagation procedure for every compared encoder. For each query frame, the encoder receives 16 frames ending at that frame. At the start of a video, we repeat the first frame to fill the window. We resize frames to a short side of 480 pixels while preserving the aspect ratio. We then pad the right and bottom edges by repeating border pixels, so both image dimensions are divisible by 16. Features come from the final normalized transformer block at the last temporal token position. Each patch vector is normalized to unit length. Inference uses BF16 and the full patch grid.

The memory contains the first annotated frame and up to 14 recent frames with predicted masks. For each query patch, we search memory patches within a circular spatial radius of 12 patch positions. We keep the five largest cosine similarities across memory frames. A softmax with temperature 0.2 converts these similarities to weights. The new label probability is the weighted sum of the stored probabilities. Queries are processed in chunks of 128 with the same similarity rule.

The initial annotation is converted to one-hot labels and average-pooled over each 16×16 patch. Predicted patch probabilities are bilinearly upsampled, cropped to remove padding, and resized to the original image size before taking the most likely label. We use no trained segmentation head, SAM2 refinement, or ground-truth corrections after initialization.

DAVIS. We evaluate the DAVIS 2017 validation split with 30 videos and 61 objects. The official evaluation code is fixed at revision `ac7c43fca936f9722837b7fbd337d284ba37004b`. We report the mean of region overlap $\mathcal{J}$ and boundary score $\mathcal{F}$, using the official frame exclusions and object aggregation.

YouTube-VOS. We evaluate 507 videos and 1,063 objects from the YouTube-VOS 2019 validation split, using the annotated-frame sequence at nominally 6 frames per second. We compute scores locally against the annotation PNGs in the public `yinloonga/YouTube_VOS_2019` mirror at revision `7a0eb80c729b0dbbceaea15950e86ed606ab21ed`. The pinned `valid.tar` archive has SHA-256 `3c847cd8f64ec6219db88671681303e84537e5478d53a88534592761ba74eb03`. The archive contains 13,710 RGB frames and 13,694 required annotation frames. Preparation verifies their dimensions and object IDs before local scoring.

The data preparation exports a separate inference manifest and initialization masks. The predictor reads RGB frames and these initial masks; later ground-truth masks are used only for scoring. Each object's declared first annotation defines its initialization time and mask. Objects introduced later have their own group and memory. A zero foreground initialization keeps a zero channel throughout propagation.

We use the J/F functions from `scoring_program_release.zip` in the same mirror, with archive SHA-256 `097bf24c6911bfccede5bd0672d117a61dc0ce1a4530d49ad551e45dd70978b8`. The scorer uses native-resolution masks for $\mathcal{J}$ and a short side of 360 for $\mathcal{F}$.

For each object, scoring excludes its initialization frame and includes its remaining metadata-listed frames. We save the mean J/F scores for every object with its video ID and category. Overall is the mean of the four scores for 883 seen and 180 unseen objects: $(\mathcal{J}_s + \mathcal{F}_s + \mathcal{J}_u + \mathcal{F}_u)/4$.

A.7 Additional object-level objectives

The two pooled-object ablations share the instance projection head with the local contrastive loss. They use its outputs before ℓ_2 normalization. The contrastive loss still uses the normalized features in Equation (2).

The pooled-object ablations use a separate assignment rule. Each retained patch is assigned to the mask with the largest coverage, provided that coverage is at least 0.5. We average its projected features within each object and frame, then across frames in which that object is present. This gives one vector per object and view. We apply no further normalization to this vector.

Object alignment uses objects present in at least two views. It compares each view's vector with that object's mean vector across available views. We average squared differences over feature coordinates, then over views, objects, and eligible videos. Object SIGReg uses each object's mean vector across available views. Objects have equal weight within each video. SIGReg uses 1,024 random projections, 17 quadrature knots, and a reference sample count of 128.

The alignment ablation uses an additional weight of 1.0. The SIGReg ablation uses 0.02. Both keep the instance weight at 0.06 and retain the global LeVJEPA objective. The two additional terms are tested separately. Instance separation alone gives the highest measured scores on all four benchmarks in these shared-head comparisons (Table 6).

A.8 Paired bootstrap analysis

We compute 95% paired bootstrap intervals from 10,000 resamples of evaluation videos with replacement, using seed 41. Each resample uses the same videos for both models, whose weights stay fixed. On YouTube-VOS, a sampled video contributes all its object scores from the local scorer in Appendix A.6. If a video is sampled more than once, its scores are included each time. We recompute the seen/unseen means and Overall for each resample. Table 13 reports the score for instance separation alone minus the score with an added object loss. The intervals describe variation across resampled evaluation videos for these fixed checkpoints.

Table 13: Measured gain from instance separation alone, in percentage points, with 95% paired bootstrap intervals.

Benchmark	Additional loss	Difference	95% CI
DAVIS	Object alignment	+2.47	$[1.07,\ 3.83]$
DAVIS	Object SIGReg	+1.05	$[-1.48,\ 3.78]$
YouTube-VOS	Object alignment	+1.23	$[0.59,\ 1.94]$
YouTube-VOS	Object SIGReg	+0.34	$[-0.60,\ 1.39]$

B Additional Results

Instance supervision improves every reported dense metric in all four matched LeVJEPA settings and both additional V-JEPA2 pairs. The gains cover mask overlap and boundary quality, for both

seen and unseen YouTube-VOS categories. LeVJEPA + instance also leads at every listed SSv2 probe epoch.

The tables give detailed scores for each data size. On YouTube-VOS, s and u denote seen and unseen categories. Bold marks the higher score within each baseline/instance pair. Additional object-level losses appear as separate ablations. The SSv2 rows show accuracy during the same completed 20-epoch probe, with the encoder frozen throughout.

Static mask control. Keeping each object's first mask unchanged gives 23.15 Overall on YouTube-VOS, with $\mathcal{J}_s = 30.00$, $\mathcal{F}_s = 21.86$, $\mathcal{J}_u = 20.78$, and $\mathcal{F}_u = 19.97$. This control uses no learned features.

B.1 50K DETAILED RESULTS

Table 14: Detailed results at 50K. Top: mask overlap and boundary scores. Bottom: SSv2 Top-1 at selected epochs of the completed 20-epoch probes.

	DAVIS			YouTube-VOS				
Method	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$	Overall
LeVJEPA	31.54	32.30	31.92	46.00	39.92	36.65	34.67	39.31
LeVJEPA + instance	**44.11**	**45.93**	**45.02**	**52.23**	**48.31**	**43.50**	**44.00**	**47.01**
V-JEPA2	24.43	23.98	24.21	33.28	25.69	26.48	22.81	27.07
V-JEPA2 + instance	**42.26**	**41.06**	**41.66**	**46.55**	**39.13**	**37.48**	**34.56**	**39.43**
LeVJEPA + instance + alignment	41.97	43.14	42.55	52.06	47.43	42.11	41.52	45.78
LeVJEPA + instance + object SIGReg	43.58	44.37	43.97	52.21	48.44	42.82	43.21	46.67

SSv2 probe training

Method	Epoch 1	Epoch 5	Epoch 10	Epoch 20
LeVJEPA	1.80	3.77	4.87	5.10
LeVJEPA + instance	**5.29**	**9.81**	**11.31**	**12.34**
V-JEPA2	6.42	10.46	12.33	13.34
V-JEPA2 + instance	**13.88**	**21.18**	**23.73**	**25.34**
LeVJEPA + instance + alignment	3.97	7.86	9.38	10.38
LeVJEPA + instance + object SIGReg	4.01	6.48	7.81	8.57

B.2 100K DETAILED RESULTS

Table 15: Detailed results at 100K. Top: mask overlap and boundary scores. Bottom: SSv2 Top-1 at selected epochs of the completed 20-epoch probes.

	DAVIS			YouTube-VOS				
Method	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$	Overall
LeVJEPA	32.83	33.66	33.25	45.49	38.97	34.42	31.90	37.69
LeVJEPA + instance	**44.64**	**46.73**	**45.68**	**53.13**	**49.01**	**42.53**	**43.09**	**46.94**

SSv2 probe training

Method	Epoch 1	Epoch 5	Epoch 10	Epoch 20
LeVJEPA	11.82	21.48	24.28	26.17
LeVJEPA + instance	**16.32**	**25.24**	**28.14**	**29.74**

B.3 200K DETAILED RESULTS

Table 16: Detailed results at 200K. Top: mask overlap and boundary scores. Bottom: SSv2 Top-1 at selected epochs of the completed 20-epoch probes.

	DAVIS			YouTube-VOS				
Method	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$	Overall
LeVJEPA	36.13	36.28	36.20	47.11	40.44	35.66	33.01	39.06
LeVJEPA + instance	**45.97**	**48.85**	**47.41**	**54.33**	**50.48**	**43.24**	**44.10**	**48.04**
V-JEPA2	39.28	39.19	39.23	46.47	38.72	36.72	33.89	38.95
V-JEPA2 + instance	**52.80**	**53.33**	**53.07**	**53.54**	**47.04**	**43.88**	**42.24**	**46.67**

SSv2 probe training

Method	Epoch 1	Epoch 5	Epoch 10	Epoch 20
LeVJEPA	16.01	25.49	28.75	30.34
LeVJEPA + instance	**21.59**	**30.42**	**32.38**	**34.08**
V-JEPA2	**28.63**	36.58	39.16	40.60
V-JEPA2 + instance	28.54	**37.00**	**39.81**	**41.30**

B.4 1M DETAILED RESULTS

Table 17: Detailed LeVJEPA results at 1M. Top: mask overlap and boundary scores. Bottom: SSv2 Top-1 at selected epochs of the completed 20-epoch probes.

	DAVIS			YouTube-VOS				
Method	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}_s$	$\mathcal{F}_s$	$\mathcal{J}_u$	$\mathcal{F}_u$	Overall
LeVJEPA	31.58	31.00	31.29	42.28	35.37	29.95	26.89	33.62
LeVJEPA + instance	**36.84**	**38.48**	**37.66**	**46.56**	**41.40**	**36.80**	**34.96**	**39.93**

SSv2 probe training

Method	Epoch 1	Epoch 5	Epoch 10	Epoch 20
LeVJEPA	15.82	30.10	35.59	38.76
LeVJEPA + instance	**23.79**	**33.65**	**38.19**	**40.89**