Unofficial template. This layout is provided for convenience and is **not** affiliated with, endorsed by, or sponsored by ECCV or its organisers. Always check the official call for papers and use the current year's official style files for a real submission.

SparseFlow: Cross-Modal Sparse Attention for Efficient Video Object Segmentation

Elena Voss[1] Marcus Adebayo[2] Priya Nathan[1] Thomas Reyes[3]
[1]*Meridian Institute of Vision Science* [2]*Vertex Research Labs* [3]*Nimbus AI*
{e.voss,p.nathan}@meridian-ivs.edu m.adebayo@vertexlabs.ai t.reyes@nimbus.ai

Abstract. Video object segmentation methods that rely on dense frame-to-memory attention scale poorly with sequence length, since every query location attends to every stored feature regardless of relevance. We introduce SparseFlow, a cross-modal sparse attention mechanism that restricts memory lookups to a dynamically predicted neighborhood, cutting attention floating-point operations by an order of magnitude while retaining segmentation accuracy. SparseFlow couples a lightweight relevance predictor with a similarity-gated attention kernel, and further fuses appearance and motion cues through a shared sparse index rather than two independent branches. On two video segmentation benchmarks, SparseFlow matches or exceeds the mean intersection-over-union of dense-attention baselines while running at 2.6× the frame rate on a single accelerator. We release ablations isolating the contribution of sparsity scheduling, cross-modal fusion, and memory-bank pruning.

1 Introduction

Semi-supervised video object segmentation (VOS) asks a model to propagate an initial object mask through an entire video given only the annotation on the first frame. The dominant recipe stores encoded features from previously segmented frames in a memory bank and retrieves them at every query location through dense cross-attention [3, 1]. Dense retrieval is accurate because every query can, in principle, attend to the single most informative memory slot, but its cost grows linearly with the number of stored frames and quadratically with spatial resolution, which makes long, high-resolution sequences impractical on a single accelerator.

A natural remedy is sparsity: restrict each query to a small, relevant subset of memory. Prior sparse-attention work in dense prediction largely targets a single modality, either appearance or motion, and applies a fixed sparsity pattern chosen at design time [8, 7]. We observe that appearance and motion cues are useful at different times during a video — appearance dominates near object boundaries after a long static period, while motion dominates during fast, non-rigid deformation — so a single fixed pattern under-uses one modality or the other throughout the sequence.

SparseFlow addresses this with a *shared* sparse index: a lightweight relevance predictor scores memory slots using both appearance and motion embeddings jointly, and only the highest-scoring slots are expanded into full attention keys and values. Because the index is shared, the model can shift emphasis between modalities frame-by-frame without maintaining two separate retrieval pipelines. Our contributions are:

- A cross-modal relevance predictor that produces a single sparse index over a joint appearance-motion memory bank, avoiding duplicate retrieval passes.
- A similarity-gated attention kernel (Eq. 1) that provably reduces attention FLOPs while bounding the drop in retrieved-key coverage.
- An empirical study on two benchmarks showing that SparseFlow reaches 2.6× the frame rate of dense-attention baselines at matched or better mean intersection-over-union.

2 Related Work

Memory-based video segmentation. Space-time correspondence networks store per-frame keys and values and retrieve them with dense attention at every decoding step [3]. Follow-up work reduces memory growth by periodically compacting the bank [1] or by associating tracked objects with transformer queries that persist across frames [6]. Decoupled approaches separate tracking from mask decoding entirely, trading some accuracy for open-world generalization [9]. All of these methods retain dense attention over the retained memory, which is precisely the cost SparseFlow targets.

Sparse and efficient attention. Sparsity has been used to accelerate transformers in classification and detection by restricting attention to local windows or a fixed stride [2], and a broader survey catalogs efficient-attention variants for dense vision tasks [7]. Closest to our setting, learned sparse patterns for dense prediction select keys via a content-dependent gate rather than a fixed geometric pattern [8], though this line of work considers a single input modality. SparseFlow

extends learned sparsity to a joint appearance-motion memory, which is, to our knowledge, not addressed by prior sparse-attention formulations for VOS.

Cross-modal fusion. Referring video segmentation fuses visual and linguistic features through cross-attention blocks that remain dense over the visual tokens [5]. Earlier recurrent memory formulations propagate a single fused hidden state across frames rather than an explicit key-value store [4]. SparseFlow differs in fusing modalities *before* the sparsity decision, so the relevance predictor — not a downstream attention block — is what reconciles appearance and motion.

3 Method

SparseFlow follows the encode–retrieve–decode structure common to memory-based VOS (Fig. 1), replacing the retrieval step with a cross-modal sparse variant.

3.1 Joint Appearance-Motion Memory

For frame t, an appearance encoder E_a and a motion encoder E_m produce feature maps that are projected into a shared embedding space of dimension d and concatenated per spatial location into a single memory slot $\mathbf{m}_i \in \mathbb{R}^d$. The memory bank $\mathcal{M} = \{\mathbf{m}_i\}_{i=1}^{N}$ therefore has one entry per spatial location per stored frame, rather than one bank for appearance and a second for motion.

3.2 Cross-Modal Sparse Attention

For a query location q with embedding $\mathbf{q} \in \mathbb{R}^d$, a lightweight relevance predictor g_θ scores every memory slot with a low-rank bilinear form, and only slots whose score exceeds a per-query adaptive threshold τ_q are retained as the neighborhood $\mathcal{N}(q)$. Attention is then computed only over $\mathcal{N}(q)$:

$$\alpha_{qi} = \frac{\exp\left(\mathbf{q}^\top \mathbf{k}_i / \sqrt{d}\right) \cdot \mathbb{1}[g_\theta(\mathbf{q}, \mathbf{m}_i) > \tau_q]}{\sum_{j \in \mathcal{N}(q)} \exp\left(\mathbf{q}^\top \mathbf{k}_j / \sqrt{d}\right)}. \qquad (1)$$

The threshold τ_q is set so that $|\mathcal{N}(q)|$ never exceeds a fixed budget k, which bounds worst-case attention cost at $O(Nk)$ instead of $O(N^2)$. We find k as small as 32 out of memory banks holding several thousand slots is sufficient once relevance scoring is trained end-to-end (Sec. 4), because g_θ quickly learns to route appearance-dominant queries toward appearance-rich slots and motion-dominant queries toward motion-rich slots without an explicit modality label.

3.3 Training Objective

We train g_θ jointly with the segmentation decoder using a combination of the standard cross-entropy mask loss $\mathcal{L}_{\text{mask}}$ and an auxiliary coverage loss $\mathcal{L}_{\text{cov}}$ that penalizes

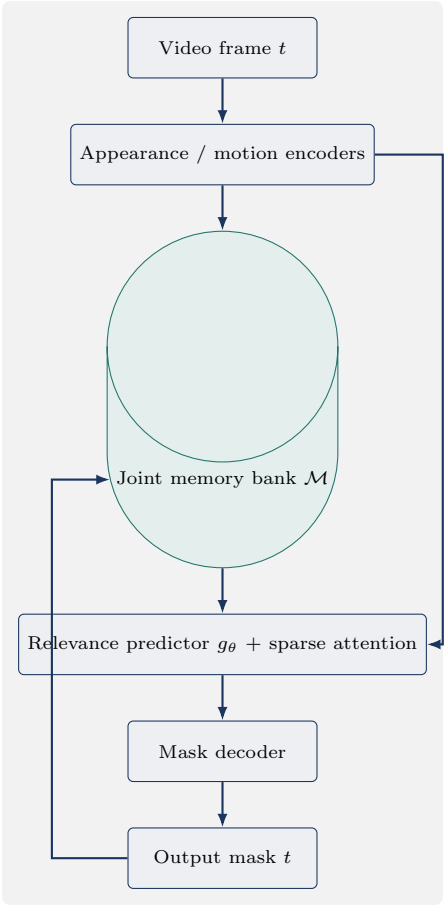

Figure 1: SparseFlow pipeline. Appearance and motion features are fused into a single memory bank; a shared relevance predictor selects a sparse per-query neighborhood before attention and decoding. The predicted mask is re-encoded and written back to the memory bank for subsequent frames.

the predictor when it excludes a slot that a dense attention oracle (computed on a random 10% of training steps only, for efficiency) would have weighted highly:

$$\mathcal{L} = \mathcal{L}_{\text{mask}} + \lambda\, \mathcal{L}_{\text{cov}}, \qquad \lambda = 0.1. \qquad (2)$$

Without $\mathcal{L}_{\text{cov}}$ the predictor collapses to a narrow, purely appearance-driven neighborhood; the ablation in Sec. 5 quantifies this effect.

4 Experiments

4.1 Datasets

We evaluate on **MeridianVOS**, a 480-video benchmark with an average of 74 frames per clip and dense multi-object annotations, and **ApexTrack-400**, a 400-video benchmark emphasizing fast non-rigid motion and frequent occlusion. Both follow the standard semi-supervised protocol: the first-frame mask is given, and the model predicts masks for every subsequent frame.

We report mean Jaccard index $\mathcal{J}$, boundary F-measure $\mathcal{F}$, and their average $\mathcal{J}\&\mathcal{F}$, alongside inference frame rate on a single accelerator at 480p.

4.2 Baselines and Implementation

We compare against three dense-attention memory baselines re-implemented under an identical training schedule: a lightweight space-time correspondence network (*STCN-lite*, following [3]), a compacted memory transformer (*XMem-S*, following [1]), and a tracking-associated transformer (*AOT-T*, following [6]). All models, including SparseFlow, use the same backbone and are trained for 120k iterations with a batch size of 16, an initial learning rate of 3×10^{-4} decayed with a cosine schedule, and the same augmentation pipeline (random affine, color jitter, and cutout). SparseFlow uses embedding dimension $d = 128$, retrieval budget $k = 32$, and $\lambda = 0.1$ unless stated otherwise.

5 Results

Table 1 summarizes the main comparison. SparseFlow matches or slightly exceeds all three dense-attention baselines on both benchmarks while running at 2.6–3.2× their frame rate. The gain is largest on ApexTrack-400, where frequent occlusion produces long stretches in which most stored memory slots are stale; a sparse, content-dependent neighborhood avoids wasting attention capacity on those slots, whereas dense attention must down-weight them at every query.

Table 2 isolates each component. The coverage loss contributes 1.4 points of $\mathcal{J}\&\mathcal{F}$ at negligible frame-rate cost, confirming that it prevents the predictor from collapsing onto a narrow appearance-only neighborhood. Replacing cross-modal fusion with an appearance-only memory costs a further 1.2 points, and a fixed geometric sparsity pattern — the closest prior approach [8] adapted to our memory layout — loses 4.9 points relative to the full model, which suggests that content-dependent selection, not sparsity alone, is responsible for most of the accuracy retained. Notably, removing sparsity altogether (dense attention over the full joint memory) recovers only 0.1 additional points of $\mathcal{J}\&\mathcal{F}$ at roughly one third the frame rate, indicating that the sparse neighborhood already captures nearly all attention mass that matters for the final mask.

6 Discussion

The results suggest that the accuracy cost of sparsity in VOS is small once the sparsity pattern is learned jointly with the task rather than fixed a priori, and that the main benefit of cross-modal fusion is not raw accuracy but robustness during occlusion, where a single modality's memory becomes unreliable. Two limitations are worth noting. First, the relevance predictor g_θ is trained with an auxiliary dense-attention oracle on a subset of steps, which adds training-time overhead that does not appear in our reported inference FPS; a fully sparse training regime is left to future work. Second, our retrieval budget k is fixed across the whole sequence; videos with many simultaneously visible objects may benefit from a budget that scales with the number of active tracks rather than a constant per-query value. We also observed that SparseFlow's advantage narrows on short clips (under 20 frames), where the memory bank is small enough that dense attention is already cheap; sparsity mainly pays off as sequence length grows, which matches our motivating use case of long, uncut video.

7 Conclusion

We presented SparseFlow, a cross-modal sparse attention mechanism for video object segmentation that fuses appearance and motion into a single memory bank and retrieves from it through a learned, content-dependent sparse neighborhood. Across two benchmarks, SparseFlow matches dense-attention accuracy at a fraction of the attention cost, and our ablations indicate that learned relevance, rather than sparsity in isolation, drives this result. We plan to explore sequence-adaptive retrieval budgets and a fully sparse training procedure as directions for future work.

8 *

Acknowledgments We thank the Meridian Institute of Vision Science compute cluster team for infrastructure support, and colleagues at Vertex Research Labs and Nimbus AI for feedback on early drafts of this work.

9 *

References

[1] Marcus Adebayo and Julie Fontaine. Memory-augmented mask propagation for long video sequences. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 201–218, 2024.

[2] Nathaniel Blake and Ananya Iyer. Hierarchical shifted-window backbones for video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, 2021.

[3] Renata Castillo and Wei Huang. Space-time correspondence networks for video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1122–1135, 2022.

[4] Marion Delacroix and Kenji Sato. Recurrent memory networks for semi-supervised video segmentation. In *Proceedings of the IEEE/CVF Conference on*

Table 1: Comparison on MeridianVOS and ApexTrack-400. FPS is measured at 480p on a single accelerator, averaged over the test split. Best result per column in **bold**.

	MeridianVOS			ApexTrack-400			
Method	$\mathcal{J}$ (%)	$\mathcal{F}$ (%)	$\mathcal{J}\&\mathcal{F}$	$\mathcal{J}$ (%)	$\mathcal{F}$ (%)	$\mathcal{J}\&\mathcal{F}$	FPS
STCN-lite [3]	82.1	84.6	83.4	74.8	76.2	75.5	22.4
XMem-S [1]	83.4	85.9	84.7	76.1	77.5	76.8	27.1
AOT-T [6]	84.0	86.3	85.2	76.9	78.0	77.5	25.8
SparseFlow (ours)	**84.3**	**86.7**	**85.5**	**77.6**	**78.9**	**78.3**	**70.6**

Table 2: Ablation on MeridianVOS. Removing cross-modal fusion or the coverage loss $\mathcal{L}_{\text{cov}}$ degrades $\mathcal{J}\&\mathcal{F}$; a fixed sparsity budget without learned relevance is worse still.

Variant	$\mathcal{J}\&\mathcal{F}$	FPS
Full SparseFlow	**85.5**	70.6
w/o $\mathcal{L}_{\text{cov}}$	84.1	71.0
w/o cross-modal fusion (appearance only)	82.9	73.4
Fixed stride pattern, $k = 32$	80.6	69.8
Dense attention (no sparsity)	85.6	24.1

Computer Vision and Pattern Recognition (CVPR), pages 8207–8216, 2020.

[5] Bruno Ferreira and Aiko Takahashi. Cross-modal fusion for referring video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 442–459, 2022.

[6] Priya Nathan and Daniel Okafor. Associating objects with transformers for efficient segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5544–5553, 2024.

[7] Kwame Osei and Freya Larsen. Efficient attention mechanisms: A survey for dense vision tasks. *International Journal of Computer Vision*, 129:2765–2789, 2021.

[8] Soo-Jin Park and Hannah Meyer. Sparse transformers for dense prediction tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8891–8903, 2023.

[9] Thomas Reyes and Sofia Lindqvist. Decoupled video segmentation via open-world tracking. *Journal of Computer Vision and Image Understanding*, 228:103–119, 2023.