FastSAM2: Fully Convolutional Networks for Fast Segment Anything in Images and Videos

Anonymous authors
Paper under double-blind review

Abstract

Segment Anything Model 2 (SAM 2) provides a unified interface for promptable image and video segmentation, but repeated high-resolution encoding and global memory attention remain costly. We investigate whether a deliberately local, content-conditioned memory operator can preserve quality when cross-frame correspondence is predominantly local, while trading flexible long-range retrieval for lower latency. We present FastSAM2, a unified fully convolutional image-video model whose Cross-frame Convolutional Aggregation (CCA) module combines temporal mixing, similarity gating, large-kernel depthwise propagation, memory-entry weighting, and gated fusion without forming a global query-key matrix. The model is trained only on public SA-1B and SA-V annotations, without pseudo-labels or distillation. Under a matched 1024×1024, batch-one, memory-write-inclusive protocol, FastSAM2 reaches 51.7 AP on LVIS and 132.5 FPS on H20, versus 41.6 FPS for SAM 2 B+; its 3.18–$3.20\times$ speedup transfers across three GPUs. FastSAM2 remains within 0.2–0.3 points of SAM 2 B+ on four short/medium-duration video benchmarks, but trails SAM 2.1 B+ on SA-V (74.9 vs. 78.2) and long-term LVOS v2 (72.9 vs. 78.2). The resulting system offers a favorable measured efficiency–quality operating point, while its degradation under long absence and reappearance is consistent with the limitations of local spatial interaction.

1 Introduction

Promptable segmentation replaces task-specific image and video pipelines with a common interface conditioned on points, boxes, or masks. SAM (Kirillov et al., 2023) established this interface for images, and SAM 2 (Ravi et al., 2025) extended it to video with a streaming memory that propagates a prompted object across frames. This unification, however, repeats two expensive operations at every frame: high-resolution visual encoding and content-dependent retrieval from an accumulated memory bank. The deployment question is therefore not simply whether SAM 2 can reach real-time throughput on a high-end GPU, but whether similar segmentation quality can be obtained with substantially lower latency and computation across hardware.

Image-only variants such as FastSAM (Zhao et al., 2023), MobileSAM (Zhang et al., 2023), EdgeSAM (Zhou et al., 2025), TinySAM (Shu et al., 2025), NanoSAM (NVIDIA AI-IOT, 2023), and EfficientSAM (Xiong et al., 2024) reduce encoder cost. For video, EfficientTAM (Xiong et al., 2025) pools memory keys and values, whereas Efficient-SAM2 (Zhang et al., 2026) sparsifies SAM 2.1 visual encoding and memory retrieval. Retaining attention is not itself a weakness: global, content-dependent retrieval is valuable for large motion and distant reappearance. It does, however, preserve a dynamic token-interaction path even when most useful correspondence lies near the current location. This motivates our central question: can an explicit local correspondence prior provide a better efficiency–quality operating point for the common regime of adjacent-frame motion?

Our hypothesis is conditional rather than universal: in many short- and medium-duration sequences, useful cross-frame evidence is spatially concentrated, so local interaction can be sufficient; after long occlusion, abrupt displacement, or distant reappearance, flexible global retrieval may remain advantageous. This hypothesis leads to **Cross-frame Convolutional Aggregation (CCA)**. Large-kernel depthwise propagation covers a bounded displacement range, similarity gates suppress locally inconsistent memory features, learned temporal weights select useful memory entries, and gated

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.

residual fusion preserves current-frame evidence. A recurrent memory carries history across frames, but CCA never constructs the full query-key matrix. Its cost is linear in spatial tokens and memory length for fixed width and kernel size. The design is thus an explicit exchange of long-range flexibility for a compact, hardware-friendly interaction path, not a claim that local convolution universally dominates attention.

We integrate CCA into **FastSAM2**, a fully convolutional system with a shared encoder and mask decoder: images bypass temporal memory, while videos activate CCA and recurrent memory. The model is jointly trained on public SA-1B (Kirillov et al., 2023) and SA-V (Ravi et al., 2025) ground truth without pseudo-labels or distillation. At 1024×1024, FastSAM2 reaches 51.7 LVIS AP and is about $3.2\times$ faster than SAM 2 B+ across three GPUs, while remaining within 0.2–0.3 points on four short/medium-duration video benchmarks. Against the stronger SAM 2.1 B+, it is 3.3–$3.5\times$ faster but less accurate on SA-V and LVOS v2. These results identify both the useful regime and the boundary of local interaction. Our contributions are:

- **A unified fully convolutional system.** FastSAM2 removes attention from both the per-frame encoder and temporal memory path while sharing the encoder and decoder across promptable image and video segmentation.
- **Content-gated local memory interaction.** CCA combines temporal mixing, similarity-gated large-kernel propagation, memory-entry weighting, and gated residual fusion, yielding linear scaling in spatial tokens and memory length without global pairwise attention.
- **A scoped empirical characterization.** Under a matched protocol, we evaluate image quality, five video benchmarks, and three GPUs against SAM 2, SAM 2.1, EfficientTAM, and Efficient-SAM2, showing where local interaction preserves quality and where long-range re-identification remains a limitation.

2 Related Work

Promptable Visual Segmentation. Image segmentation has progressed from fully convolutional networks (Long et al., 2015) and region-based detectors (Bolya et al., 2019; He et al., 2017; Redmon et al., 2016) through scene parsing (Chen et al., 2017; Zhao et al., 2017), transformer-based encoders (Touvron et al., 2021; Xie et al., 2021), and universal frameworks (Cheng et al., 2022; Jain et al., 2023). SAM (Kirillov et al., 2023) reframed the task around promptability, and SAM 2 (Ravi et al., 2025) extended it to video with streaming memory attention. The resulting ecosystem spans high-quality variants (Ke et al., 2023; Ding et al., 2025), open-vocabulary segmentation (Ren et al., 2024; Zou et al., 2023; Liu et al., 2024), tracking (Cheng et al., 2023; Yang et al., 2023), and memory-based VOS (Oh et al., 2019; Cheng & Schwing, 2022; Cheng et al., 2024). FastSAM2 preserves the unified promptable interface while replacing temporal attention with a fully convolutional interaction operator.

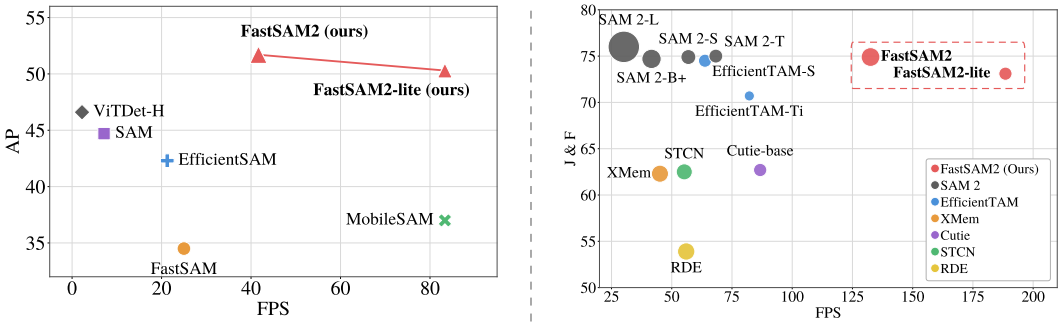

Figure 1: Efficiency–quality comparison at 1024×1024 on H20. *Left:* LVIS image AP versus FPS for the evaluated image baselines. *Right:* SA-V $\mathcal{J}\&\mathcal{F}$ versus FPS, with bubble area proportional to parameters. FastSAM2 occupies a favorable operating point among the methods shown; broader video baselines and long-term results appear in Tables 2 and 3.

Efficient Segment Anything Models. The cost of SAM and SAM 2 has motivated efficient variants. FastSAM (Zhao et al., 2023) uses a YOLOv8-based CNN, while MobileSAM (Zhang et al., 2023), EdgeSAM (Zhou et al., 2025), TinySAM (Shu et al., 2025), NanoSAM (NVIDIA AI-IOT, 2023), and EfficientSAM (Xiong et al., 2024) compress or pretrain lightweight image encoders. For video, EfficientTAM (Xiong et al., 2025) pools memory keys and values, and Efficient-SAM2 (Zhang et al., 2026) adds object-aware sparse window routing and memory retrieval to SAM 2.1. Both preserve attention, as do general low-rank, kernelized, and windowed alternatives (Wang et al., 2020; Beltagy et al., 2020; Katharopoulos et al., 2020). FastSAM2 instead replaces the cross-frame interaction module, yielding a complementary efficiency–quality operating point rather than claiming dominance over every efficient-attention design.

Large-Kernel CNNs and CNN-Transformer Hybrids. Several lines of work have revitalized convolution as a competitive primitive. Attention can express convolutional layers; trained attention and effective receptive fields are often spatially concentrated; and CNN/ViT representations share substantial structure (Cordonnier et al., 2020; Luo et al., 2016; Raghu et al., 2021; Yu et al., 2022). Deep CNNs have a formal universality result (Zhou, 2020), while positional encodings are central to universality arguments for sequence models (Yun et al., 2020). Large-kernel CNNs compete strongly with vision transformers (Ding et al., 2022; 2024; Liu et al., 2022; Woo et al., 2023), and DECO (Chen et al., 2025) substitutes depthwise interaction modules for attention in detection. CCA adapts these ideas to promptable video segmentation with similarity-gated temporal aggregation.

3 Approach

3.1 Preliminaries

Promptable Visual Segmentation. SAM (Kirillov et al., 2023) introduced the Segment Anything task for images, in which a prompt such as a point, box, or mask drives the model to output a segmentation mask of the object of interest. SAM 2 (Ravi et al., 2025) generalizes this to Promptable Visual Segmentation, which extends promptable segmentation to the video domain, where prompts on any frame define a spatio-temporal mask, denoted a masklet, to be predicted across all frames. SAM 2 treats an image as a single-frame video and, in the absence of temporal context, reduces to SAM.

Cross-frame interaction via memory attention. SAM 2 realizes cross-frame interaction with a stack of L memory-attention blocks. Each block performs self-attention on the current-frame feature sequence $\mathbf{F}_t \in \mathbb{R}^{N\times d}$ and then cross-attends to a memory bank $\mathcal{M} = \{\mathbf{M}_i\}_{i=1}^{T}$, $\mathbf{M}_i \in \mathbb{R}^{n_i \times d}$, that stacks features and object pointers of past frames. Writing the operation per query position $p \in \{1, \dots, N\}$ and per memory entry i, the cross-attention output is

$$\mathrm{Attn}(\mathbf{F}_t, \mathcal{M})_p \;=\; \sum_{i=1}^{T} \sum_{q\in\Omega_i} \alpha_{p,q}^{(i)}\, \mathbf{V}_q^{(i)}, \qquad \alpha_{p,q}^{(i)} \;=\; \mathrm{softmax}_{(i,q)}\left(\frac{\langle \mathbf{Q}_p, \mathbf{K}_q^{(i)}\rangle}{\sqrt{d}}\right), \tag{1}$$

where Ω_i indexes the spatial positions of memory entry $\mathbf{M}_i$, and $\mathbf{Q}, \mathbf{K}^{(i)}, \mathbf{V}^{(i)}$ are the query, key and value projections. The total cost across L blocks is $\mathcal{O}\big(L\, N(n_1 + \cdots + n_T)\, d\big)$; on long videos the memory size $\sum_i n_i$ exceeds $30k$ tokens, so cross-frame attention dominates inference. Crucially, Equation (1) treats every memory position (i, q) uniformly and leaves it to the data to learn that, between adjacent frames, the truly informative positions live in a small spatial neighborhood of p.

3.2 FastSAM2 Architecture

FastSAM2 has two design goals. First, it replaces the attention-based cross-frame interaction of Equation (1) with a fully convolutional operator whose cost is linear in the number of tokens and whose receptive field is matched to the spatiotemporal support that actually carries inter-frame correspondence. Second, it unifies image and video segmentation under a single CNN backbone so that adding video capability does not sacrifice image performance. The resulting pipeline in Figure 2 uses only convolutional primitives end-to-end.

Shared CNN encoder for image and video. FastSAM2 uses a CSPDarknet (Jocher et al., 2023) plus PAN-FPN encoder shared across image and video tasks, initialized from pretrained FastSAM-x (Zhao

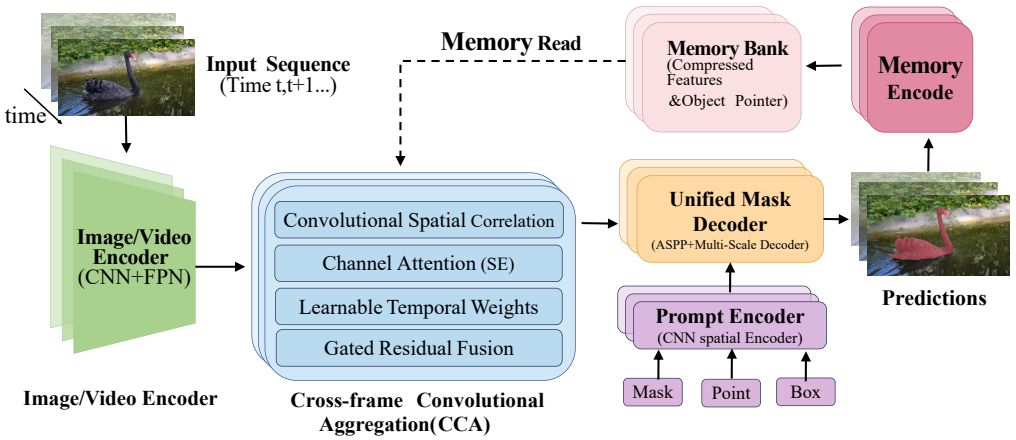

Figure 2: Overview of the FastSAM2 pipeline. The *image/video encoder* is a CNN backbone with PAN-FPN that extracts multi-scale features from an input sequence. The *CCA module* reads memory features from the memory bank and fuses them with current-frame features through four convolutional stages: 3D temporal depthwise convolution, similarity-gated large-kernel spatial correlation, learnable temporal weighting, and gated residual fusion. The *prompt encoder* encodes user prompts of mask, points, or box. The *unified mask decoder* uses ASPP and multi-scale decoding to produce segmentation masks conditioned on CCA-enhanced features and prompt embeddings. A *memory encoder* compresses predicted masks and features into the memory bank as compressed features and an object pointer, forming a recurrent memory read and write cycle for the video branch.

et al., 2023). For a frame $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, it produces features $\{P_2, P_3, P_4, P_5\}$ at strides $\{4, 8, 16, 32\}$. Image mode passes them directly to the shared mask decoder; video mode first enhances P_4 with CCA and temporal memory. Both tasks therefore update the shared encoder and mask head. The reported gains characterize this combined architecture, initialization, and joint-training recipe; without a matched training-by-operator factorial experiment, we do not attribute them to CCA alone.

3.3 Cross-frame Convolutional Aggregation

The design goal of CCA is to construct a lower-cost cross-frame operator with the same input-output type as memory cross-attention in Equation (1). We first state the operator, then describe its four-stage realization, and finally separate family-level expressivity motivation from the cost and empirical behavior of the deployed finite model.

The CCA operator. Let $\mathbf{F}_t \in \mathbb{R}^{C\times H\times W}$ be the stride-16 current-frame feature map, let $\mathcal{M} = \{\mathbf{M}_i\}_{i=1}^{T}$ be T memory entries with $\mathbf{M}_i \in \mathbb{R}^{C\times H\times W}$, and let $N = HW$. Denote by $\mathcal{N}_k(p)$ the $k\times k$ spatial neighborhood centered at position p. The Cross-frame Convolutional Aggregation operator at position p is

$$\mathrm{CCA}(\mathbf{F}_t, \mathcal{M})_p \;=\; \sum_{i=1}^{T} w_i \sum_{q\in\mathcal{N}_k(p)} \beta_{p,q}^{(i)}\, \mathbf{W}_k^{\mathrm{dw}}(p-q) \odot \mathbf{V}_q^{(i)}, \qquad \beta_{p,q}^{(i)} \;=\; \sigma\big(\mathbf{Q}_p \odot \mathbf{K}_q^{(i)}\big), \tag{2}$$

where $\mathbf{Q}_p$, $\mathbf{K}_q^{(i)}$, and $\mathbf{V}_q^{(i)}$ are the 1×1-projected query, key, and value features at the indicated positions, σ is the sigmoid, $\odot$ denotes element-wise multiplication, $\mathbf{W}_k^{\mathrm{dw}}(p-q)$ is the entry of a learnable depthwise kernel of support $k\times k$ at offset $p-q$, and $\{w_i\}_{i=1}^{T}$ are learned scalar temporal weights produced by a small MLP applied to globally pooled descriptors of $\mathbf{F}_t$ and each $\mathbf{M}_i$. The dynamic per-token gate $\beta_{p,q}^{(i)}$ replaces the data-dependent attention coefficient $\alpha_{p,q}^{(i)}$ in Equation (1), the static depthwise kernel $\mathbf{W}_k^{\mathrm{dw}}$ replaces the global summation over Ω_i, and the scalar weights w_i replace the cross-entry component of the softmax. Stacking L blocks gives the full video branch.

Convolutional realization. A CCA block implements Equation (2) through four convolutional stages. *Stage 1* stacks the current frame and the T memory entries along a temporal axis and applies

a separable temporal-then-spatial depthwise convolution with kernels of size $T\times1\times1$ and $1\times3\times3$ followed by a residual connection, producing a temporally aware query map from which the projection $\mathbf{Q}$ is read. *Stage 2* computes, for each memory entry, the inner sum of Equation (2): a sigmoid gate $\sigma(\mathbf{Q}\odot\mathbf{K}^{(i)})$ selects memory positions that agree with the query, the gated values are propagated by a $k\times k$ depthwise kernel with default $k=9$, and a channel squeeze-and-excitation module recalibrates the channels. *Stage 3* produces the scalar weights w_i by feeding globally pooled descriptors of $\mathbf{Q}$ and $\mathbf{M}_i$ through a two-layer MLP and applying a softmax across memory entries. *Stage 4* merges the aggregated correlation feature with the current-frame feature through a learned spatial gate computed by a 1×1 convolution on their concatenation. The detailed block diagram, the explicit equation for each stage, and the symbol list are given in Section D.

Expressivity scope. The family-level result in Theorem 1 states that a sufficiently wide convolutional spatiotemporal operator can approximate continuous memory attention on a fixed bounded grid when its stacked receptive field covers the full domain (Zhou, 2020; Yun et al., 2020). We move the statement and proof sketch to Section A because it motivates the operator family rather than guarantees the finite model. The deployed $L=3, k=9$ CCA covers only a 25×25 neighborhood on the 64×64 feature grid; its accuracy must therefore be established empirically, and Table 3 shows the resulting weakness on long-term LVOS v2.

Computational footprint. Because each query position only accesses a fixed-size local window, the cost of one CCA block grows linearly with both spatial tokens and memory length. Summing the costs of the four stages,

$$\mathcal{C}_{\mathrm{CCA}} \;=\; \mathcal{O}(NTd) \;+\; \mathcal{O}(NTdk^2) \;+\; \mathcal{O}(Td^2) \;+\; \mathcal{O}(Nd) \;=\; \mathcal{O}\big(NTdk^2\big), \tag{3}$$

where $N=HW$, T is memory length, d is width, and k is kernel size. Sub-quadratic cost is not itself novel: efficient attention already uses low-rank, kernel-feature, or windowed approximations (Xiong et al., 2025; Wang et al., 2020; Beltagy et al., 2020; Katharopoulos et al., 2020). CCA instead imposes a local convolutional interaction and never forms the full $\mathbf{QK}^\top$ matrix. The resulting latency, full cost comparison, and receptive-field trade-off are evaluated in Sections B and 4.2.

3.4 Prompt Encoder, Mask Decoder, and Memory System

Prompt encoder. The prompt encoder converts three prompt types into embeddings for the mask decoder. Mask prompts, which include the first-frame ground-truth mask or a mask, are processed by lightweight convolutions, while point and box prompts are encoded via positional embeddings. Users can initialize tracking with a single click, a bounding box, or a full mask.

Unified mask decoder. The mask decoder takes CCA-enhanced features $\hat{\mathbf{F}}_t$ together with prompt embeddings and produces the segmentation mask $\hat{M}_t\in\mathbb{R}^{H\times W}$. An ASPP (Chen et al., 2017) module provides multi-scale context, and skip connections from $\{P_2, P_3, P_4\}$ recover fine-grained boundaries. The same decoder is used for both image and video tasks.

Memory system. A recurrent memory system maintains a compact history of past predictions. After each frame is decoded, a memory encoder compresses the predicted mask and decoder features into a memory entry that is pushed into the memory bank, and an object pointer tracks object-level state through an exponential moving average,

$$\mathbf{p}_t \;=\; \alpha\,\mathbf{p}_{t-1} \;+\; (1-\alpha)\,\mathbf{f}_t, \tag{4}$$

where α is a momentum coefficient and $\mathbf{f}_t$ is the current object feature. At each new frame, CCA reads $\mathcal{M}=\{\mathbf{M}_i\}$ from the memory bank and the encoder writes back the new entry, forming a recurrent cycle that supports long-range tracking at constant per-frame cost.

Unified image-video inference. At inference time, an image is processed by feeding $\{P_2, P_3, P_4, P_5\}$ directly into the Segment Anything branch of the mask decoder, bypassing CCA and the memory system. A video is processed by initializing the first frame with a prompt and then unrolling the CCA and memory recurrence for subsequent frames. A single set of weights therefore serves both tasks.

Table 1: Image segmentation results on LVIS. FastSAM2 achieves the best overall and size-specific AP among the evaluated methods. FastSAM2-lite retains 50.3 AP at 83.4 FPS; its AP_L of 66.2 is within 0.1 of ViTDet-H. FPS is measured on one H20 at 1024×1024. Best in **bold**.

Model	AP	AP_S	AP_M	AP_L	FPS
ViTDet-H (Li et al., 2022)	46.6	35.0	58.0	66.3	2.2
SAM (Kirillov et al., 2023)	44.7	32.5	57.6	65.5	7.1
FastSAM (Zhao et al., 2023)	34.5	24.6	46.2	50.8	21.3
EfficientSAM (Xiong et al., 2024)	42.3	30.8	54.0	62.3	25.0
MobileSAM (Zhang et al., 2023)	37.0	–	–	–	83.3
FastSAM2-lite (ours)	50.3	40.1	60.5	66.2	**83.4**
FastSAM2 (ours)	**51.7**	**41.9**	**62.1**	**67.8**	41.7

4 Experiments

4.1 Experimental Setting

Model configurations. We evaluate both instantiations defined in Section C. The default **FastSAM2** has 76M parameters and 198.2 GFLOPs, while the compressed **FastSAM2-lite** has 34.5M parameters and 81.7 GFLOPs.

Joint image-video training on SA-1B and SA-V. We train only on public ground-truth SA-1B (Kirillov et al., 2023) and SA-V (Ravi et al., 2025): no SAM 2/SAM 2.1 pseudo-labels, teacher distillation, private data, or generated data are used. SA-1B contributes 1,031,178 indexed images from 100 public shards with at most eight masks per image; SA-V contributes 50,583 training videos, sampling one masklet and one four-frame, stride-four clip per video iteration. Optimization alternates image and video batches at a $1:1$ ratio. The shared CSPDarknet is initialized from public FastSAM-x; all other modules are random. We train for 50 epochs with FP16 AMP, AdamW (Loshchilov & Hutter, 2019), learning rate 10^{-4}, weight decay 0.01, gradient clipping 1.0, three-epoch warmup, cosine decay to 10^{-6}, and batch size 16 across eight H20 GPUs. No test-time augmentation or post-processing is used. Full sampling and loss details appear in Section P.

Evaluation. We evaluate first-frame-prompted VOS on SA-V test (Ravi et al., 2025), MOSE val (Ding et al., 2023), DAVIS 2017 val (Pont-Tuset et al., 2017), YouTube-VOS 2019 val (Xu et al., 2018), and long-term LVOS v2 (Hong et al., 2026). Image segmentation uses LVIS (Gupta et al., 2019). All timing is FP32 at 1024×1024, batch size one, with 20 warm-up iterations and 100 synchronized timed frames. It includes encoder, temporal interaction, decoder, and memory write, but excludes I/O, preprocessing, and mask post-processing; no autocast or `torch.compile` is used. We report H20, A800, and RTX 4090 to separate operator complexity from hardware-specific latency. Full software and timing details appear in Section P.

4.2 Main Results

We evaluate FastSAM2 and FastSAM2-lite on both image and video segmentation, following the evaluation protocols of SAM 2 (Ravi et al., 2025) and EfficientTAM (Xiong et al., 2025).

4.2.1 Image Segmentation.

Table 1 reports LVIS results. FastSAM2 reaches 51.7 AP at 41.7 FPS, improving over the evaluated FastSAM, EfficientSAM, MobileSAM, SAM, and ViTDet-H baselines; FastSAM2-lite reaches 50.3 AP at 83.4 FPS. Because the system changes architecture, initialization, and training jointly, these numbers support the combined recipe rather than a causal claim that video supervision or CCA alone produces the gain. A complete joint-training-by-operator factorial retraining was not performed. Qualitative comparisons appear in Section O.

Table 2: Comparison with video segmentation baselines on MOSE val, DAVIS 2017 val, YouTube-VOS 2019 val, and SA-V test. Quality is $\mathcal{J}\&\mathcal{F}$ except YouTube-VOS ($\mathcal{G}$); *Avg.* averages the four quality columns. FPS is measured on one H20 at 1024×1024. Best in **bold**, second best underlined.

Model	Size (M)	FLOPs (G)	FPS	MOSE val	DAVIS 17 val	YTVOS 19 val ($\mathcal{G}$)	SA-V test	Avg.
STCN (Cheng et al., 2021)	54	342.5	55.2	52.5	85.4	82.7	62.5	70.8
XMem (Cheng & Schwing, 2022)	62	151.7	45.1	59.6	86.0	85.6	62.3	73.4
DEVA (Cheng et al., 2023)	–	–	–	66.0	87.0	85.4	56.2	73.7
Cutie-base (Cheng et al., 2024)	35	99.4	86.7	69.9	87.9	87.0	62.7	76.9
EfficientTAM-Ti/2 (Xiong et al., 2025)	18	–	127.3	68.4	88.4	87.1	70.8	78.7
EfficientTAM-Ti (Xiong et al., 2025)	18	–	82.2	69.3	89.1	86.7	70.7	79.0
EfficientTAM-S/2 (Xiong et al., 2025)	34	–	86.9	70.8	88.6	87.2	74.0	80.2
EfficientTAM-S (Xiong et al., 2025)	34	141.4	63.8	71.4	89.2	87.2	74.5	80.6
SAM 2 B+ (Ravi et al., 2025)	80.8	261.7	41.6	72.8	90.2	88.6	74.7	81.6
SAM 2.1 B+ (Meta AI, 2024)	80.8	–	39.7	**73.7**	**90.7**	–	**78.2**	–
Efficient-SAM2-B+ (Zhang et al., 2026)	–	–	63.5	70.9	89.3	–	75.5	–
FastSAM2-lite (ours)	34.5	**81.7**	**188.5**	70.2	88.7	86.7	73.1	79.7
FastSAM2 (ours)	76.0	198.2	132.5	73.0	90.4	**88.9**	74.9	**81.8**

Table 3: Matched speed and video quality. FPS uses FP32, batch one, 1024×1024, and includes memory write. Quality is $\mathcal{J}\&\mathcal{F}$ except YouTube-VOS ($\mathcal{G}$). FastSAM2's speedup transfers across GPUs, while LVOS v2 exposes its long-range limitation.

	FPS			Segmentation quality				
Model	H20	A800	4090	MOSE	DAVIS	YTVOS	SA-V	LVOS v2
EfficientTAM-S (Xiong et al., 2025)	63.8	45.6	25.0	71.4	89.2	87.2	74.5	73.4
SAM 2 B+ (Ravi et al., 2025)	41.6	29.7	16.4	72.8	90.2	88.6	74.7	75.8
SAM 2.1 B+ (Meta AI, 2024)	39.7	27.3	15.1	73.7	90.7	–	78.2	78.2
Efficient-SAM2-B+ (Zhang et al., 2026)	63.5	43.7	24.1	70.9	89.3	–	75.5	–
FastSAM2-lite (ours)	**188.5**	**134.3**	**73.9**	70.2	88.7	86.7	73.1	70.8
FastSAM2 (ours)	132.5	94.9	52.2	73.0	90.4	88.9	74.9	72.9

4.2.2 Video Segmentation.

Table 2 compares FastSAM2 with a broad set of VOS and efficient-memory baselines on four short/medium-duration benchmarks, including SAM 2.1 and Efficient-SAM2 where results are available. Table 3 complements this comparison with multi-GPU timing and long-term LVOS v2 evaluation. All FPS values follow the protocol in Section 4; dashes denote unavailable completed measurements.

Among methods with all four quality measurements, FastSAM2 reaches the highest average of 81.8, compared with 81.6 for SAM 2 B+ and 80.6 for EfficientTAM-S, while running at 132.5 FPS. SAM 2.1 B+ is more accurate on the three reported datasets, while Efficient-SAM2-B+ exceeds FastSAM2 by 0.6 on SA-V but trails it on MOSE and DAVIS; their averages are omitted because YouTube-VOS results are unavailable. FastSAM2-lite provides the fastest operating point at 188.5 FPS and 79.7 average quality. Table 3 further tests hardware transfer and long-term re-identification.

FastSAM2 matches or slightly improves over SAM 2 on MOSE, DAVIS, YouTube-VOS, and SA-V while running 3.18–3.20× faster across the three GPUs. Relative to SAM 2.1 it is 3.34–3.48× faster, but trails by 3.3 on SA-V and 5.3 on LVOS v2. Against Efficient-SAM2-B+ it is about 2.1× faster, with $+2.1$ on MOSE, $+1.1$ on DAVIS, and -0.6 on SA-V. FastSAM2-lite is the fastest operating point but drops further on LVOS. Thus, the evidence supports a favorable efficiency–quality trade-off, not hardware-independent latency or uniform accuracy superiority.

Qualitative Results. Figure 3 compares FastSAM2 with SAM 2 (Ravi et al., 2025) on the DAVIS 2017 *blackswan* and *camel* sequences using the first-frame ground-truth mask as the only

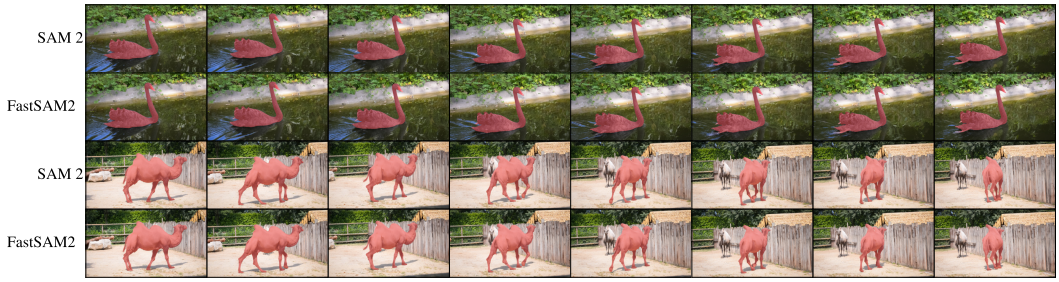

Figure 3: Qualitative comparison with SAM 2 on DAVIS 2017 val. For *blackswan* (top two rows) and *camel* (bottom two rows), the upper row shows SAM 2 and the lower row shows FastSAM2. Eight uniformly sampled frames are shown with predicted masks overlaid in red. FastSAM2 maintains comparable masks using fully convolutional cross-frame interaction.

Table 4: Ablation on DWConv kernel size.

Kernel	ERF ($L=3$)	$\mathcal{J}\&\mathcal{F}$
3×3	7×7	0.65
5×5	13×13	0.70
7×7	19×19	0.72
9×9	25×25	**0.75**
11×11	31×31	0.74
13×13	37×37	0.72

Table 5: CCA stage component ablation.

Configuration	$\mathcal{J}\&\mathcal{F}$	Δ
Full CCA	**0.75**	–
w/o gated fusion	0.55	-0.20
w/o 3D temp. conv	0.60	-0.15
w/o channel attn.	0.70	-0.05
w/o spatial corr.	0.70	-0.05
w/o temp. weights	0.72	-0.03

prompt. Both methods preserve the target through deformation, scale change, and partial occlusion; additional SA-V sequences appear in Section N.

4.3 Ablation Studies

We ablate key design choices of CCA on SA-V val. Unless stated otherwise, all ablations are run under the FastSAM2 configuration.

Table 4 shows an empirical optimum at $k=9$: smaller kernels under-cover common inter-frame displacement, whereas larger kernels admit more background and distractor evidence without improving correspondence. Table 5 shows that gated fusion and temporal convolution are the most consequential CCA stages. These component removals validate the internal design, but they do not isolate CCA from joint training or initialization. Additional depth, memory-bank, and local-attention ablations appear in Sections L and M; deformable convolution, state-space alternatives, and a complete matched operator factorial remain future comparisons.

5 Conclusion

We presented **FastSAM2**, a unified fully convolutional image-video segmentation system built around CCA. Under matched timing, its approximately $3.2\times$ speedup over SAM 2 transfers from H20 to A800 and RTX 4090, and it remains about $2.1\times$ faster than Efficient-SAM2-B+. On short/medium benchmarks it matches SAM 2, but SAM 2.1 is more accurate on SA-V and LVOS v2, and the long-term gap is largest after disappearance and distant reappearance. The theorem in Theorem 1 characterizes a sufficiently wide, full-receptive-field convolutional family; it does not guarantee the deployed local CCA. Accordingly, our supported claim is a favorable measured efficiency–quality trade-off from the combined architecture and training recipe, not universal replacement of attention or attribution of the full gain to CCA alone. Extended limitations and reproducibility details appear in Sections P and Q.

AI USE STATEMENT

Generative AI tools were used to assist with adapting the LaTeX source to the ICLR 2027 template, editing presentation and citation formatting, and checking bibliographic metadata against authoritative publication records. The authors reviewed the AI-assisted edits and the cited records and take responsibility for the final text, claims, experiments, and references.

REPRODUCIBILITY STATEMENT

The model architecture, training protocol, datasets, evaluation settings, and hardware used in the experiments are described in Sections 3 and 4. Additional implementation details, complexity analysis, proof sketches, ablations, and qualitative results are provided in the appendix.

REFERENCES

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9157–9166, 2019.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. doi: 10.48550/arXiv.1706.05587.

Xinghao Chen, Siwei Li, Yijing Yang, and Yunhe Wang. DECO: Unleashing the potential of ConvNets for query-based detection and segmentation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1290–1299, 2022.

Ho Kei Cheng and Alexander G Schwing. XMem: Long-term video object segmentation with an Atkinson-Shiffrin memory model. In *European Conference on Computer Vision*, pp. 640–658, 2022.

Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *Advances in Neural Information Processing Systems*, 2021.

Ho Kei Cheng, Seoung Wug Oh, Brian Price, Alexander Schwing, and Joon-Young Lee. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1316–1326, 2023.

Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3151–3161, 2024.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations*, 2020.

Henghui Ding, Chang Liu, Shuting He, Xudong Jiang, Philip H. S. Torr, and Song Bai. MOSE: A new dataset for video object segmentation in complex scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20224–20234, 2023.

Shuangrui Ding, Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Yuwei Guo, Dahua Lin, and Jiaqi Wang. SAM2Long: Enhancing SAM 2 for long video segmentation with a training-free memory tree. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13614–13624, 2025.

Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in CNNs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11963–11975, 2022.

Xiaohan Ding, Yiyuan Zhang, Yixiao Ge, Sijie Zhao, Lin Song, Xiangyu Yue, and Ying Shan. UniRepLKNet: A universal perception large-kernel ConvNet for audio, video, point cloud, time-series and image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5513–5524, 2024.

Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5356–5364, 2019.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Lingyi Hong, Zhongying Liu, Wenchao Chen, Chenzhi Tan, Yuang Feng, Xinyu Zhou, Pinxue Guo, Jinglun Li, Zhaoyu Chen, Shuyong Gao, Wei Zhang, and Wenqiang Zhang. LVOS: A benchmark for large-scale long-term video object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(1):946–961, 2026. doi: 10.1109/TPAMI.2025.3611020.

Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2989–2998, 2023.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLOv8. GitHub repository, 2023.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pp. 5156–5165, 2020.

Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. *Advances in Neural Information Processing Systems*, 36: 29914–29934, 2023. doi: 10.52202/075280-1303.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pp. 280–296, 2022.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Computer Vision – ECCV 2024*, pp. 38–55, 2024. doi: 10.1007/978-3-031-72970-6_3.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2016.

Meta AI. SAM 2.1 Developer Suite. GitHub repository, 2024. URL https://github.com/facebookresearch/sam2.

NVIDIA AI-IOT. NanoSAM. GitHub repository, 2023.

Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9226–9235, 2019.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Advances in Neural Information Processing Systems*, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations*, 2025.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded SAM: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

Han Shu, Wenshuo Li, Yehui Tang, Yiman Zhang, Yihao Chen, Houqiang Li, Yunhe Wang, and Xinghao Chen. TinySAM: Pushing the envelope for efficient segment anything model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(19):20470–20478, 2025. doi: 10.1609/aaai.v39i19.34255.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, volume 139, pp. 10347–10357. PMLR, 2021.

Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16133–16142, 2023.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, Raghuraman Krishnamoorthi, and Vikas Chandra. EfficientSAM: Leveraged masked image pretraining for efficient segment anything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16111–16121, 2024. doi: 10.1109/CVPR52733.2024.01525.

Yunyang Xiong, Chong Zhou, Xiaoyu Xiang, Lemeng Wu, Chenchen Zhu, Zechun Liu, Saksham Suri, Balakrishnan Varadarajan, Ramya Akula, Forrest Iandola, Raghuraman Krishnamoorthi, Bilge Soran, and Vikas Chandra. Efficient track anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11513–11524, 2025.

Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. YouTube-VOS: Sequence-to-sequence video object segmentation. In *European Conference on Computer Vision*, pp. 585–601, 2018.

Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023.

Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. MetaFormer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10819–10829, 2022.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr.

Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight SAM for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

Jing Zhang, Zhikai Li, Xuewen Liu, and Qingyi Gu. Efficient-SAM2: Accelerating SAM2 with object-aware visual encoding and memory retrieval. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2602.08224.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.

Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

Chong Zhou, Xiangtai Li, Chen Change Loy, and Bo Dai. EdgeSAM: Prompt-in-the-loop distillation for SAM. *International Journal of Computer Vision*, 133(12):8452–8468, 2025. doi: 10.1007/s11263-025-02562-9.

Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, 2020. doi: 10.1016/j.acha.2019.06.004.

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in Neural Information Processing Systems*, 36:19769–19782, 2023. doi: 10.52202/075280-0868.

A Family-Level Expressivity and Proof Sketch

This result concerns an ideal convolutional family, not the deployed finite CCA.

Theorem 1 (Universal approximation of memory attention by convolutional operators)**.** *Fix a spatial resolution $H\times W$, memory length T, and bounded feature domain $\mathcal{K}\subset\mathbb{R}^{(T+1)\times H\times W\times d}$. The memory-attention block of Equation (1) defines a continuous map $\mathcal{A}\colon \mathcal{K}\to\mathbb{R}^{C\times H\times W}$. For every $\varepsilon>0$, there exists a fully convolutional spatiotemporal operator Φ with sufficient width, pointwise nonlinearities, positional information, and a stacked receptive field covering the complete spatiotemporal domain such that*

$$\sup_{(\mathbf{F}_t,\mathcal{M})\in\mathcal{K}} \|\Phi(\mathbf{F}_t,\mathcal{M}) - \mathcal{A}(\mathbf{F}_t,\mathcal{M})\| < \varepsilon. \tag{5}$$

We sketch the argument behind Equation (5); the result is an expressivity statement without a finite-width or finite-receptive-field error bound.

Step 1: Memory cross-attention is continuous on bounded inputs. The map $\mathcal{A}:\ (\mathbf{F}_t,\mathcal{M})\mapsto \mathrm{Attn}(\mathbf{F}_t,\mathcal{M})$ in Equation (1) is a finite composition of 1×1 linear projections to query, key and value, inner products $\mathbf{Q}\mathbf{K}^\top/\sqrt{d}$, softmax over the finite index set $\bigcup_i \Omega_i$, and convex combinations of value features. Each primitive is continuous on $\mathbb{R}^{(T+1)\times H\times W\times d}$, and softmax is in addition Lipschitz on any bounded subset of its argument. Restricting to a bounded feature domain $\mathcal{K}$, we conclude that $\mathcal{A}\colon \mathcal{K}\to\mathbb{R}^{C\times H\times W}$ is continuous and the image $\mathcal{A}(\mathcal{K})$ is compact.

Step 2: Convolutional spatiotemporal operators are universal approximators on the fixed grid. On the fixed grid $\{1,\dots,T+1\}\times\{1,\dots,H\}\times\{1,\dots,W\}$, deep CNNs with sufficient width/depth and full-domain receptive-field coverage admit a universal approximation property (Zhou, 2020). Coordinate channels remove the translation-equivariance restriction when absolute position matters, paralleling the role of positional encodings in sequence-to-sequence universality (Yun et al., 2020). Pointwise projections provide channel mixing and convolution supplies spatial-temporal mixing. This applies to the broad family Φ; it does not state that depthwise convolution with any fixed width or local support is universal.

Step 3: Combine. Combining Step 1 and Step 2, $\mathcal{A}$ is a continuous map of the bounded input set $\mathcal{K}$, and the convolutional family $\{\Phi\}$ contains universal approximators on this set. For every $\varepsilon>0$, there exists a parameter setting of Φ such that

$$\sup_{(\mathbf{F}_t,\mathcal{M})\in\mathcal{K}} \big\|\Phi(\mathbf{F}_t,\mathcal{M}) - \mathcal{A}(\mathbf{F}_t,\mathcal{M})\big\| < \varepsilon, \tag{6}$$

which is exactly Equation (5). CCA in Equations (2) and (8) to (11) is a finite, efficient member inspired by this family, but the theorem supplies no approximation guarantee for the deployed $L=3, k=9$ instance. Its adequacy is empirical: kernel and depth ablations support this operating point on short/medium clips, whereas LVOS v2 in Table 3 exposes a remaining long-range gap.

Remark on positional information. If memory attention uses absolute positional encodings, translation equivariance can restrict the maps a plain convolution represents. Appending coordinate channels (x, y, τ) resolves this at no asymptotic cost (Yun et al., 2020); FastSAM2 supplies positional features through PAN-FPN. This construction is part of the family-level argument and does not close the deployed model's finite-capacity gap.

B Receptive Field, Complexity, and Inductive Bias of CCA

This appendix expands on the receptive field, computational footprint, and inductive bias of the CCA operator in Equation (2).

B.1 Stacked Receptive Field of CCA

Theorem 1 requires that the convolutional operator Φ has a stacked receptive field large enough to cover the spatiotemporal memory domain. We compute the receptive field of CCA at the default settings to characterize where on the approximation-cost frontier of Theorem 1 the practical instantiation sits. Because Stage 2 is a depthwise convolution of support $k\times k$, stacking L CCA blocks yields

a feature-map receptive field of radius $L(k-1)/2$ at stride 16, equivalent in input-image coordinates to

$$R_{\mathrm{CCA}} \;=\; 16 \cdot L\,(k-1)/2, \tag{7}$$

where L is the number of stacked CCA blocks and k is the depthwise kernel size of Stage 2. With $L=3$ and $k=9$, $R_{\mathrm{CCA}}=192$ pixels, or a 25×25 neighborhood on the 64×64 stride-16 grid. The gap to full-grid coverage separates deployed CCA from the family in Theorem 1. Increasing depth or kernel size can reduce that gap at higher cost, but the kernel ablation shows that naive enlargement also admits distractors. The locality analyses in Sections E to J explain why the local field works on typical short/medium sequences; the LVOS v2 result in Table 3 shows that it is not sufficient for all long-range re-identification cases.

B.2 Per-Block Complexity Comparison

Table 6 instantiates the per-block complexity in Equation (3) together with that of memory cross-attention and a local-window attention reference at the default resolution.

Table 6: Per-block complexity at 1024×1024, with $N=4096$, $d=256$, $k=9$, and $T=6$. CCA is linear in N and T but uses local rather than global interaction.

Operator	Complexity	FLOPs per frame (stride 16)
Memory cross-attention (Equation (1))	$\mathcal{O}(N^2Td)$	$\sim$25.8G
Local window attention ($w=9$)	$\mathcal{O}(Nw^2Td)$	$\sim$0.51G
CCA (Equation (2))	$\mathcal{O}(NTdk^2)$	$\sim$0.51G

B.3 Additional Inductive-Bias Advantages

Beyond receptive field coverage and complexity, CCA offers two further inductive-bias advantages over attention-based memory modules with comparable receptive fields.

- *Translation equivariance.* Objects of interest can appear at arbitrary spatial positions. A depthwise convolution is translation-equivariant by construction through weight sharing across space, whereas attention must learn positional information through positional encodings.
- *Compact spatial mixing primitive.* Luo et al. (2016) showed that the effective receptive field of deep networks follows a Gaussian distribution with energy decaying as $\exp(-r^2/2\sigma^2)$ from the center, so even attention layers with a global theoretical receptive field allocate most of their effective capacity to a compact spatial neighborhood around each query. CCA realizes this compact spatial mixing pattern directly through the large-kernel depthwise support in Equation (7), which removes the need to learn it from data. Empirically, our ablation in Section E shows that letting attention learn this pattern from data is difficult and often counterproductive at the resolutions used here.

C Model Instantiations: FastSAM2 and FastSAM2-lite

We release two instantiations of the architecture that share the same prompt encoder, mask decoder, memory encoder, and loss, and differ only in encoder width and the number L of stacked CCA blocks.

FastSAM2 is the default model. Its encoder is a wide CSPDarknet plus PAN-FPN initialized from a FastSAM-x checkpoint, and its video branch stacks $L=3$ CCA blocks with $k=9$ depthwise kernels, embedding dimension 256, and a memory bank of 6 entries. The full network is jointly fine-tuned on SA-1B and SA-V.

FastSAM2-lite is a compressed variant designed for real-time inference. Three design choices produce the compression while preserving accuracy. First, the encoder is narrowed by scaling the CSPDarknet channel counts by a factor of 0.5, which yields $\sim 2.0\times$ fewer encoder parameters and reduces the memory footprint. Second, the CCA stack is shallower with $L=2$ blocks and

the CCA embedding dimension halved from 256 to 128, reducing the cross-frame interaction cost in Equation (3) by $\sim 4\times$ per block. Third, the memory bank is reduced from 6 to 4 compressed entries with the 160-dim compression kept unchanged, removing redundant memory tokens while maintaining long-range tracking via the object-pointer EMA in Equation (4). FastSAM2-lite is jointly trained with FastSAM2 from scratch to avoid student-teacher gaps.

D Detailed CCA Block Architecture

Figure 4 illustrates the full architecture of a single Cross-frame Convolutional Aggregation block, which is stacked L times in the video branch. Each block takes as input the current frame features $\mathbf{F}_t$ and memory features $\{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_T\}$ from the memory bank, and outputs the temporally enhanced features $\hat{\mathbf{F}}_t$. The block consists of four stages, each summarized below by the tensor it produces and the explicit equation it computes.

Stage 1: 3D Depthwise Temporal Convolution. The current frame features and memory features are stacked along a temporal axis into a tensor $\mathbf{Z} \in \mathbb{R}^{C\times(T+1)\times H\times W}$. A separable temporal-then-spatial depthwise convolution with a residual connection produces a temporally aware query map

$$\mathbf{F}_t^{\text{3D}} \;=\; \big[\,\text{DWConv}_{1\times3\times3} \circ \text{DWConv}_{T\times1\times1}(\mathbf{Z})\big]_{t=0} \;+\; \mathbf{F}_t, \tag{8}$$

where the temporal kernel has size $T\times1\times1$ with groups equal to C, the spatial kernel has size $1\times3\times3$ with groups equal to C, and a pointwise 1×1 convolution with BN and SiLU activation is applied between them. The $t=0$ slice corresponds to the current frame.

Stage 2: Cross-InterConv. The top branch projects $\mathbf{F}_t^{\text{3D}}$ into a query $\mathbf{Q}$ via $\text{Conv}_{1\times1}$ + BN + SiLU. The bottom branch projects each memory $\mathbf{M}_i$ into key $\mathbf{K}^{(i)} = \text{PW}_K(\mathbf{M}_i)$ and value $\mathbf{V}^{(i)} = \text{PW}_V(\mathbf{M}_i)$. An element-wise similarity map $\sigma(\mathbf{Q} \odot \mathbf{K}^{(i)})$ gates the values, a $k\times k$ depthwise convolution with default $k = 9$ propagates the gated evidence over a local neighborhood, and a channel squeeze-and-excitation module recalibrates the channels, yielding the per-entry correlation map

$$\text{Corr}_i \;=\; \text{SE}\Big(\mathbf{W}_k^{\text{dw}} \;*\; \big[\sigma(\mathbf{Q} \odot \mathbf{K}^{(i)}) \odot \mathbf{V}^{(i)}\big]\Big), \tag{9}$$

where $*$ denotes depthwise convolution and the SE module follows the standard $\text{GAP} \to \text{Conv} \to \text{SiLU} \to \text{Conv} \to \sigma$ structure.

Stage 3: Learnable Temporal Weights. Global average pooling is applied to both $\mathbf{Q}$ and each $\mathbf{M}_i$, and the concatenated descriptors pass through a two-layer MLP φ that scores the relevance of each memory entry to the current frame. A softmax normalizes these scores into weights and aggregates the per-entry correlation maps,

$$w_i \;=\; \frac{\exp\big(\varphi(\text{GAP}(\mathbf{F}_t^{\text{3D}}), \text{GAP}(\mathbf{M}_i))\big)}{\sum_{j=1}^{T} \exp\big(\varphi(\text{GAP}(\mathbf{F}_t^{\text{3D}}), \text{GAP}(\mathbf{M}_j))\big)}, \qquad \mathbf{F}_{\text{agg}} \;=\; \sum_{i=1}^{T} w_i \cdot \text{Corr}_i\,. \tag{10}$$

Stage 4: Gated Fusion. An output projection $\text{Conv}_{1\times1} \circ \text{DWConv}_{3\times3}$ transforms $\mathbf{F}_{\text{agg}}$, and a learned spatial gate G_t controls how much temporal evidence is fused into the current-frame feature,

$$\begin{aligned} \hat{\mathbf{F}}_t &= G_t \odot \big[\,\text{Conv}_{1\times1} \circ \text{DWConv}_{3\times3}(\mathbf{F}_{\text{agg}})\big] + (1 - G_t) \odot \mathbf{F}_t, \\ G_t &= \sigma\big(\,\text{Conv}_{1\times1}([\mathbf{F}_t; \mathbf{F}_{\text{agg}}])\big), \end{aligned} \tag{11}$$

where $[\cdot;\cdot]$ denotes channel-wise concatenation. The gate preserves reliable current-frame features when the temporal context is noisy or redundant.

E Empirical Spatial Concentration of SAM 2 Memory Attention

Theorem 1 concerns a sufficiently wide, full-domain convolutional family. Deployed CCA instead has the finite receptive field in Equation (7). We use the following analyses only to motivate the default $L = 3, k = 9$ operating point on typical sequences: trained SAM 2-style attention is often spatially concentrated, and the CCA stack covers much of that local contribution region. These observations are consistent with the short/medium benchmark results and the kernel/depth ablations, but they neither measure an approximation error to attention nor contradict the long-range failure shown on LVOS v2.

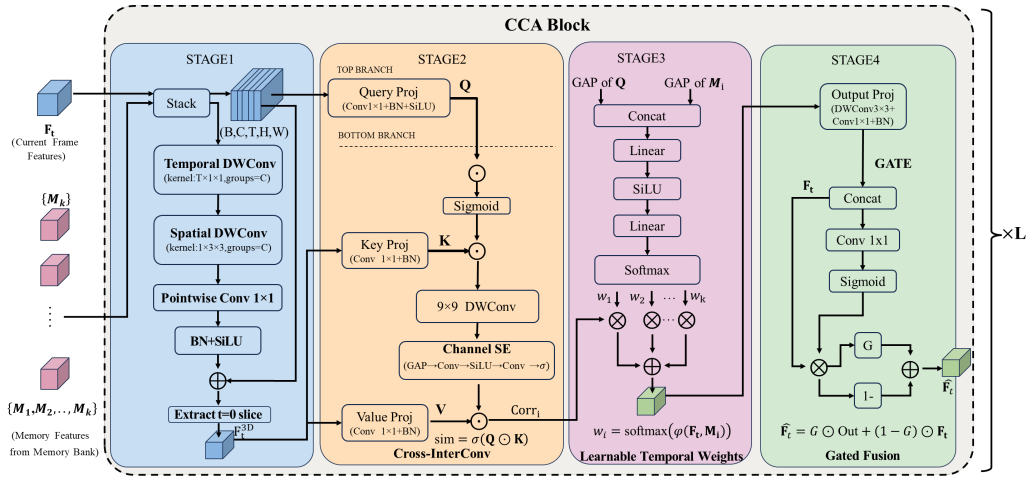

Figure 4: Detailed architecture of a CCA block, stacked L times. **Stage 1** performs 3D depthwise temporal convolution to capture short-range temporal patterns. **Stage 2** computes the spatial correlation between the current frame and each memory entry via similarity-gated 9×9 depthwise convolution. **Stage 3** aggregates multi-memory correlation features via softmax-normalized scalar weights. **Stage 4** integrates the aggregated temporal features with the original frame features through a learned gate.

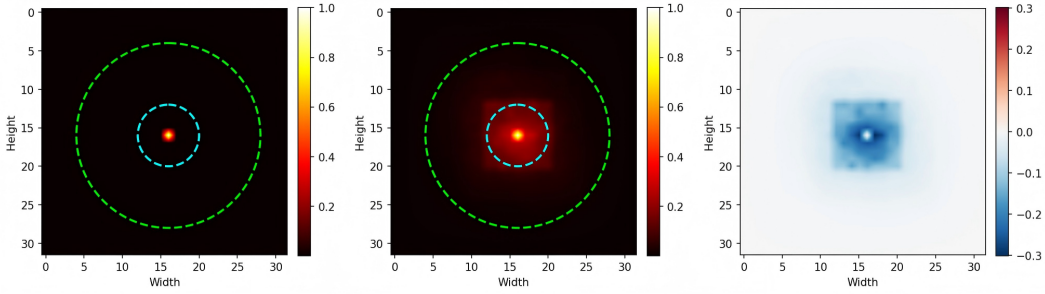

Figure 5: Effective receptive field (ERF) heatmaps for the transformer (*left*), CNN (*middle*), and their difference (*right*). Both exhibit Gaussian distributions as predicted by theory (Luo et al., 2016). Dashed circles mark the single 9×9 DWConv RF (*cyan*) and the 3-layer DECO RF at $r=12$ (*green*). Despite a global theoretical RF, the transformer's ERF concentrates within the CNN's designed RF boundary.

E.1 Effective Receptive Field Analysis

Luo et al. (2016) proved that the ERF of deep networks follows a Gaussian with $\sigma = O(k\sqrt{n})$, occupying only a small local fraction of the theoretical RF, so a transformer's effective contribution region is concentrated in a local neighborhood even when its theoretical receptive field is global. We measure gradient-based ERFs for a transformer and a CNN trained on video segmentation at 32×32 resolution, as shown in Figure 5 and Table 7. The transformer's ERF is Gaussian with $\sigma = 7.46$ and only 63.0% energy within $r \leq 12$, while our CNN with 3 layers of 9×9 DWConv and total RF $= 25\times25$ achieves $\sigma = 5.35$ with 86.9% energy in the same radius, so the CNN's designed RF already covers the transformer's effective contribution region. Additional cross-section visualizations are in Section F.

E.2 Attention Distance Analysis

We directly analyze the attention weights of the transformer by measuring the spatial distance between each query and its attended keys across 4 layers. The distributions are heavily skewed toward short distances: the mean attention distance per head is only 14–17 pixels out of a maximum of 43.8 (ratio

Table 7: ERF statistics. Both models exhibit Gaussian ERFs. The CNN's 3-layer 9×9 DWConv RF ($r \leq 12$) covers the transformer's effective utilization region, despite the transformer's global theoretical RF.

Model	Theoretical RF	Gaussian σ	Eff. Radius (90%)	Energy $r\leq12$	Energy $r\leq15$
Transformer	global	7.46	17.0	63.0%	80.0%
CNN ($3\times(9\times9)$)	25×25	5.35	14.0	86.9%	94.7%

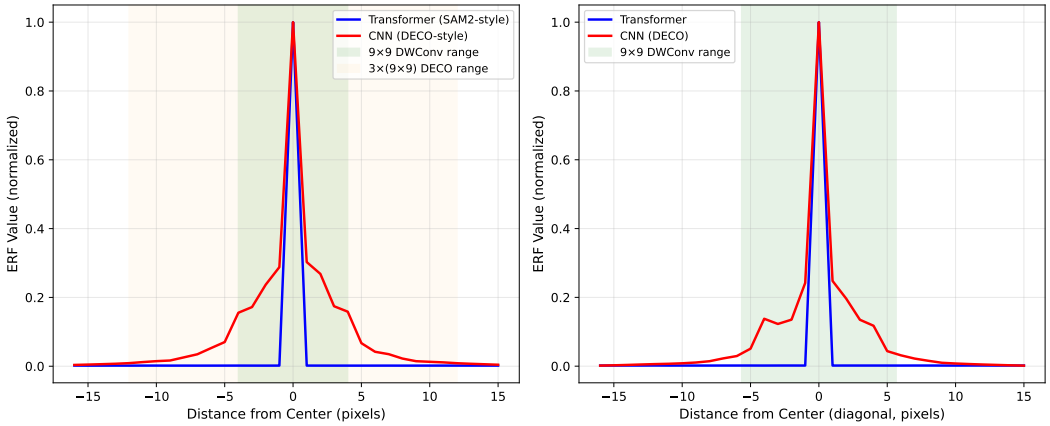

Figure 6: ERF cross-sections (horizontal and diagonal) through the center. Both ERFs exhibit Gaussian decay; the CNN (*red*, $\sigma=5.35$) is more concentrated than the transformer (*blue*, $\sigma=7.46$). Shaded regions mark the 9×9 DWConv (*green*) and 3-layer DECO (*yellow*) ranges.

0.34–0.44), and the 3-layer DECO RF ($r=12$) already captures $\sim$37.5% of attention mass while the remainder spreads thinly over a much larger area with diminishing returns. Detailed visualizations are provided in Sections G to J.

E.3 Local Attention Ablation

We conduct a controlled ablation comparing global attention, local windowed attention at various window sizes, and a pure CNN baseline under identical architecture and training. Smaller windows consistently outperform larger ones, with the window-3 variant reaching IoU 0.736 versus IoU 0.121 for global attention; global attention performs worst as distant positions inject noise; and the pure CNN with 9×9 DWConv attains the best IoU of 0.764, surpassing the best local-attention variant. The built-in locality prior of convolutions is therefore more effective than learned attention patterns. Full results are reported in Section M.

F ERF Cross-Section Visualization

Figure 6 provides horizontal and diagonal cross-sections through the ERF center, directly visualizing the Gaussian decay predicted by Luo et al. (2016). Both models peak sharply at the center, but the CNN (*red*) has a tighter, more concentrated profile ($\sigma=5.35$) than the transformer (*blue*, $\sigma=7.46$). The shaded regions mark the DWConv and DECO receptive field ranges, which cover the dominant energy for both architectures. The transformer's tail beyond the DECO boundary carries negligible energy, consistent with the $\exp(-r^2/2\sigma^2)$ Gaussian decay.

G Attention Distance Distribution

Figure 7 shows the attention distance distribution for each head across 4 transformer layers trained on video segmentation. The distributions are heavily skewed toward short distances, with the majority

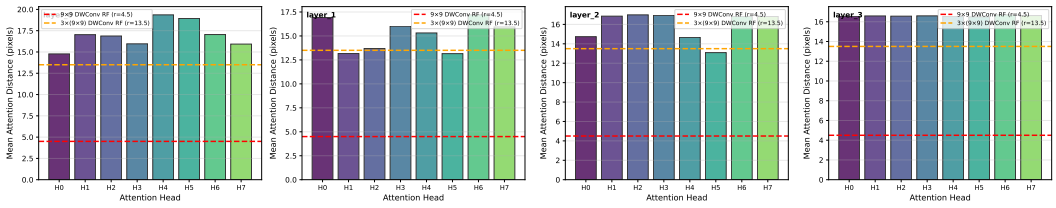

Figure 7: Attention distance distribution per head across 4 transformer layers trained on video segmentation. Most attention mass concentrates near the query position, indicating local behavior. Vertical dashed lines mark the 9×9 DWConv ($r=4$, *red*) and $3\times(9\times9)$ DECO ($r=12$, *green*) receptive field boundaries.

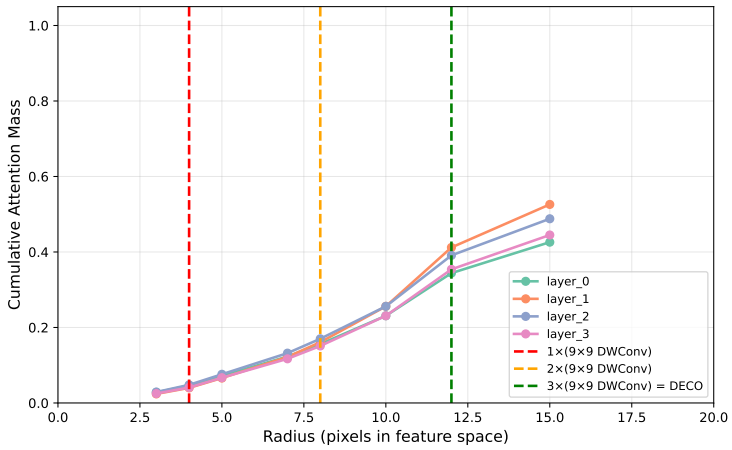

Figure 8: Cumulative attention mass within radius r for each transformer layer (*solid lines*), compared against CNN receptive field boundaries (*dashed lines*). The 3-layer DECO receptive field ($r=12$, *green dashed*) captures a substantial fraction of the attention's effective range.

of attention mass concentrated near the query position. The red vertical dashed line at $r=4$ marks the receptive field of a single 9×9 DWConv, and the green vertical dashed line at $r=12$ marks the 3-layer CCA receptive field. The mean attention distance per head ranges from 14 to 17 pixels, with a maximum possible distance of 43.8, yielding a distance ratio of 0.34 to 0.44 across layers.

H Cumulative Attention Mass

Figure 8 directly quantifies the overlap between attention and CNN receptive fields. At $r=12$, the 3-layer CCA boundary, approximately 37.5% of average attention mass is captured. While this fraction may seem modest, the 9×9 DWConv efficiently captures the most critical local correspondences while the remaining long-range attention contributes diminishing returns, as confirmed by the local attention ablation in Table 4.

I Attention Entropy Analysis

Figure 9 shows the normalized attention entropy per head per layer. Lower entropy indicates more spatially concentrated attention patterns. While most heads exhibit relatively high entropy in the range 0.7 to 1.0, some heads in early layers show notably lower entropy near 0.5, suggesting a mix of local and diffuse attention strategies. The average entropy across heads in the right panel of Figure 9 increases with depth, indicating that deeper layers tend toward more distributed attention, though still far from uniform.

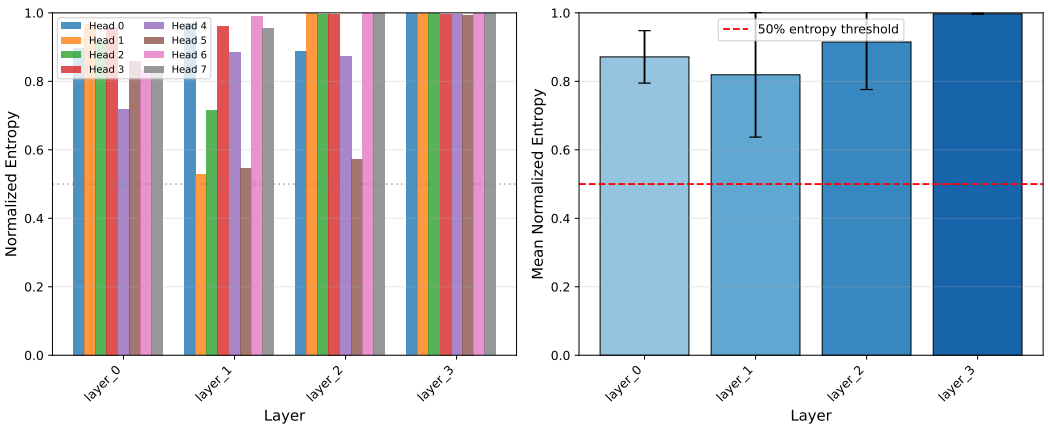

Figure 9: Attention entropy analysis. *Left*: Normalized entropy per head per layer; lower values indicate more concentrated (local) patterns. *Right*: Average entropy per layer with error bars. The red dashed line marks the 50% entropy threshold. Most heads operate above this threshold, but the overall pattern shows attention is not uniformly distributed.

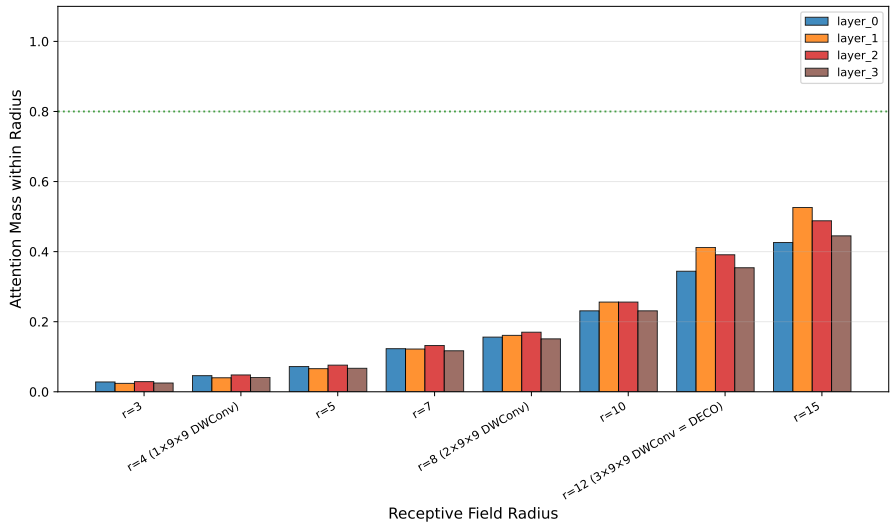

Figure 10: Attention locality score: fraction of attention mass within CNN receptive fields of radius r. The $3\times(9\times9)$ DECO receptive field ($r=12$) captures 35–40% of attention mass. The 80% threshold (*green dashed line*) requires $r>25$, but the ablation shows that attention beyond $r=12$ contributes negatively to segmentation performance.

J Attention Locality Score

Figure 10 directly measures how much attention mass falls within CNN receptive fields of increasing size. For the 3-layer CCA receptive field at $r=12$, approximately 35% to 40% of attention mass is captured across all layers. While this fraction is modest compared to the 80% threshold marked by the green dashed line, it should be interpreted in context: the remaining attention mass is spread thinly over a much larger area and contributes diminishing returns, as demonstrated by the local attention ablation in Section M.

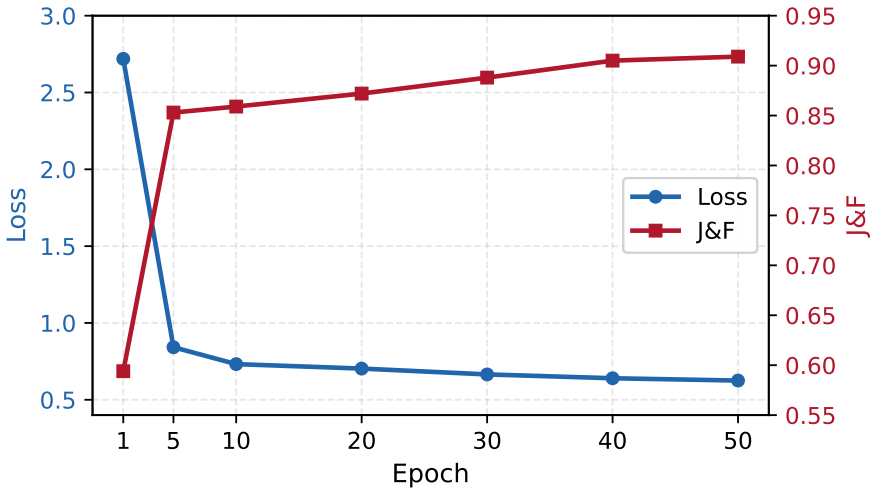

Figure 11: Training dynamics of FastSAM2-lite on the SA-V dataset. The loss (*blue*, left axis) drops rapidly in the first 5 epochs and then decreases gradually, while $\mathcal{J}\&\mathcal{F}$ (*red*, right axis) rises quickly to 0.853 at epoch 5 and converges to 0.909 at epoch 50.

Table 8: Ablation on memory bank size.

Bank Size	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}\&\mathcal{F}$
1 (anchor)	0.680	0.740	0.710
2	0.676	0.744	0.710
4	0.735	0.761	0.748
6	**0.740**	**0.758**	**0.749**
8	0.736	0.758	0.747
12	0.734	0.760	0.747

Table 9: Ablation on CCA layers L.

L	Params (M)	GFLOPs	$\mathcal{J}\&\mathcal{F}$
1	30.1	72.4	0.69
2	32.3	77.1	0.72
3	34.5	81.7	**0.75**
4	36.7	86.3	0.75

K Training Dynamics

Figure 11 shows the training dynamics of FastSAM2-lite on the SA-V dataset. The model converges quickly, achieving $\mathcal{J}\&\mathcal{F} = 0.853$ within 5 epochs and reaching 0.909 at convergence at epoch 50. The smooth, monotonic curves indicate that the pure CNN architecture is stable and does not suffer from the optimization difficulties sometimes observed with attention mechanisms.

L Detailed Ablation Analysis

This section provides additional ablation studies and extended analysis. All experiments are conducted on the SA-V val split (Ravi et al., 2025) under identical training settings: 8-frame sequences, 1024×1024 resolution, and SA-V train as the training source. Kernel size and CCA stage ablations are presented in the main paper in Tables 4 and 5.

Impact of Kernel Size. As shown in Table 4, increasing the kernel from 3×3 to 9×9 yields consistent improvements with a $+0.10$ gain in $\mathcal{J}\&\mathcal{F}$. Beyond 9×9, performance degrades, with 11×11 at 0.74 and 13×13 at 0.72, aligning with our ERF analysis: receptive fields larger than 25×25 extend beyond the task-relevant local region.

Number of CCA Layers. Table 9 shows that increasing L from 1 to 3 brings consistent gains of $+0.03$ per layer. Beyond $L=3$, additional layers provide no further improvement while increasing parameters from 34.5M to 36.7M and latency.

Memory Bank Size. As shown in Table 8, a single anchor frame at size 1 achieves only $\mathcal{J}\&\mathcal{F}=0.710$, and adding one recent frame at size 2 yields no improvement, indicating that a single recent memory is insufficient. A clear jump occurs at size 4 with 0.748, and performance peaks at size 6 with 0.749.

Table 10: Local attention ablation on synthetic video segmentation. Restricting attention to smaller local windows *improves* performance, and pure CNN achieves the best result.

Configuration	Mean IoU	Std IoU	Mean F1	Std F1
CNN (Conv 9×9)	**0.764**	0.160	**0.853**	0.143
Window = 3	0.736	0.173	0.833	0.158
Window = 5	0.713	0.201	0.810	0.192
Window = 7	0.619	0.244	0.729	0.244
Window = 9	0.501	0.277	0.613	0.300
Global Attention	0.121	0.054	0.212	0.087

Beyond 6, adding more entries at sizes 8 or 12 provides no further gain at 0.747, while increasing memory and computation cost. Six entries yield the best trade-off.

CCA Stage Component Ablation. As described in Section D, the CCA block consists of four stages: 3D temporal convolution, spatial correlation via large-kernel DWConv with SE, learnable temporal weights, and gated residual fusion. Table 5 shows that the gated fusion of Stage 4 is the most critical component, and removing it causes the largest drop of -0.20 in $\mathcal{J}\&\mathcal{F}$, indicating that the learned gate is essential for balancing temporal and current-frame evidence. The 3D temporal convolution of Stage 1 ranks second with -0.15, confirming the importance of inter-frame interaction before cross-attention. Channel attention at -0.05 and spatial correlation at -0.05 provide moderate contributions, while temporal weights at -0.03 contribute the least but still offer meaningful gains. All four stages are necessary for optimal performance.

M Local Attention Ablation

To isolate the effect of attention locality, we conduct a controlled ablation comparing global attention, local windowed attention at various window sizes, and a pure CNN baseline. All models share an identical architecture, differing only in the choice of attention or convolution mechanism, and are trained with the same data and hyperparameters.

Table 10 and Fig. 12 present the results, revealing three key findings.

Locality is beneficial. Smaller attention windows consistently outperform larger ones. Window-3 attention achieves IoU 0.736, dramatically outperforming global attention at IoU 0.121 by 0.615 points. Attending to all positions introduces noise from irrelevant features.

Global attention performs worst. With full global attention, the model achieves only IoU 0.121, substantially below all local variants. In visual segmentation, useful correspondence information is predominantly local, and global attention wastes model capacity learning to ignore irrelevant tokens.

CNN outperforms local attention. The CNN baseline with 9×9 depthwise convolutions achieves the best IoU of 0.764, surpassing even the best local attention variant at IoU 0.736. The built-in locality prior of convolutions is more effective than learned local attention patterns.

N Additional Qualitative Results on SA-V

In addition to the DAVIS 2017 val comparison shown in Figure 3 of the main paper, Figure 13 reports qualitative video segmentation results of FastSAM2 on the SA-V test set. The figure spans both single-object and multi-object tracking scenarios drawn from the long-video, in-the-wild setting that SA-V is designed to evaluate. Despite using a purely convolutional cross-frame interaction operator (CCA), FastSAM2 maintains accurate segmentation across frames under deformation, viewpoint change, background clutter, and multi-object occlusion.

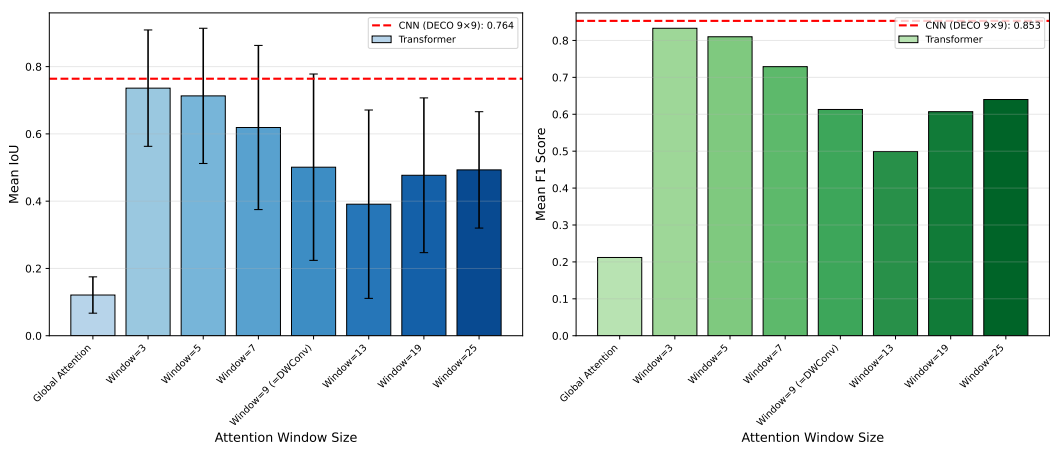

Figure 12: Video segmentation performance vs. attention window size. *Left*: Mean IoU; *Right*: Mean F1 score. Blue/green bars show transformer variants with different window sizes; the red dashed line marks the pure CNN (Conv 9×9) baseline. Performance improves monotonically as the window shrinks, and the CNN achieves the best result (IoU 0.764), demonstrating that the locality inductive bias is an advantage for visual segmentation.



Figure 13: Qualitative video segmentation results of FastSAM2 on SA-V. Each row shows a sequence with the predicted mask overlay across 5 frames. Rows 1 to 4: single-object tracking under deformation and viewpoint change. Rows 5 and 6: multi-object tracking with distinct mask colors per object. FastSAM2 maintains accurate segmentation across frames using purely convolutional temporal fusion.

O Qualitative Comparison on Promptable Image Segmentation

To complement the LVIS quantitative results in Table 1, Figure 14 compares FastSAM2 with representative Segment Anything baselines under three canonical promptable image segmentation

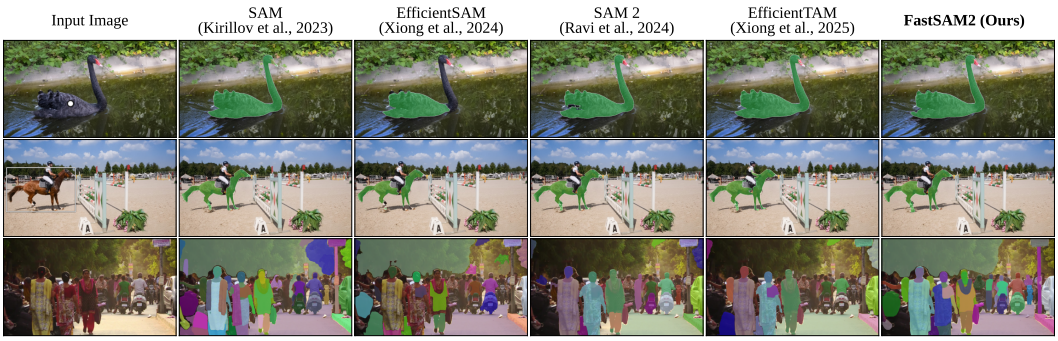

Figure 14: Qualitative comparison of promptable image segmentation. Top group: point prompt. Middle group: box prompt. Bottom group: segment-anything everything task. Frames are sampled from DAVIS 2017 and shared across the three prompt types. FastSAM2 yields tighter boundaries and recovers thin or small parts that the baselines miss while operating in real time.

protocols: a single foreground point, a tight bounding box, and the segment-anything everything task that requires the model to enumerate all objects in the image. We use frames sampled from DAVIS 2017 as the underlying images so that the same scene can be visualized across all three prompt types. Across point, box, and segment-anything prompts, FastSAM2 produces tighter mask boundaries and recovers thin or small parts that the baselines miss, while operating in real time at the FPS reported in Table 1. This indicates that the unified joint SA-1B and SA-V training does not trade prompt-conditioned image quality for video tracking ability.

P Reproducibility and Additional Validation

Supervision and initialization. The reported FastSAM2 and FastSAM2-lite checkpoints use no SAM 2/SAM 2.1 pseudo-labels, teacher-student distillation, private data, generated data, or SAM-derived filtering. The shared CSPDarknet encoder is initialized from the public FastSAM-x checkpoint; PAN-FPN, CCA, prompt encoder, mask decoder, and memory encoder are randomly initialized. A separate exploratory SAM 2.1-teacher branch exists in the broader codebase, but none of the results in this paper use it.

Data and optimization. We index $1{,}031{,}178$ SA-1B images from 100 public shards and sample at most eight masks per image. From $50{,}583$ public SA-V training videos, each video iteration samples one masklet and one four-frame clip with temporal stride four. Training alternates image and video batches, uses FP16 AMP, and applies the optimizer schedule in Section 4. The objective weights binary cross-entropy, Dice, and IoU losses by 1.0, 2.0, and 1.0, respectively. Evaluation uses neither test-time augmentation nor CRF, multi-scale ensembling, or other mask post-processing. This public training scale is smaller than the complete internal data-engine schedule reported for SAM 2.

Matched timing protocol. All new timing uses PyTorch 2.8.0+cu128, CUDA 12.8, cuDNN 91002, FP32 inference, batch size one, and 1024×1024 inputs. We run 20 warm-up iterations and time 100 frames with `torch.cuda.synchronize()` around the measured region. The region includes image encoder, temporal interaction, mask decoder, and memory write for every method, with memory banks filled before steady-state measurement; it excludes disk I/O, resizing/normalization, and mask post-processing. Autocast and `torch.compile` are disabled. SAM 2 uses the official Hiera-B+ configuration/checkpoint and PyTorch SDPA with flash, memory-efficient, and math backends enabled. These controls support comparison at the reported operating point, not an implementation-independent speed guarantee.

Scope of additional experiments. The LVOS v2 results in Table 3 test long-duration disappearance and reappearance. The memory-bank ablation in Table 8 uses eight-frame sequences, not eight-second clips, so its saturation near six entries does not establish sufficiency for long videos. We did not complete matched SA-1B-only versus joint training, attention-versus-CCA retraining, ordered-versus-shuffled clips, a sample-count-matched frame control, or full-benchmark Linformer/linear/local-

attention variants. Consequently, we describe FastSAM2 as a combined architecture-and-training contribution and do not assign the entire image or video gain to temporal consistency or CCA alone.

Q Discussion

A unified multi-task fully convolutional architecture. FastSAM2 shares its CNN encoder and mask head across image and video modes; CCA and recurrent memory are active only for video. Joint SA-1B/SA-V optimization exposes the shared representation to both data sources, but our completed experiments do not disentangle data diversity, repeated observations, temporal consistency, and the architecture change. The supported result is therefore the performance of the combined system, not a causal claim that temporal supervision alone explains the LVIS gain.

Why does the practical receptive field of CCA suffice? Enlarging CCA beyond $k = 9$ hurts SA-V validation, and smaller attention windows improve the controlled synthetic reference in Tables 4 and 10. These observations support a local prior for typical adjacent-frame correspondence, not universal sufficiency. LVOS v2 provides the counterpoint: after long absence and distant reappearance, the same local field trails SAM 2 and SAM 2.1. Adaptive global retrieval or a hybrid local/global memory is a promising extension.

Connections to efficient memory attention. EfficientTAM (Xiong et al., 2025) pools memory keys/values, while Efficient-SAM2 (Zhang et al., 2026) sparsifies visual encoding and memory retrieval; low-rank, kernelized, and windowed attention offer other trade-offs (Wang et al., 2020; Beltagy et al., 2020; Katharopoulos et al., 2020). CCA's distinction is architectural: it never forms the full $\mathbf{QK}^\top$ matrix. Under matched timing FastSAM2 is about $2.1\times$ faster than Efficient-SAM2-B+, but their relative quality depends on the benchmark. We therefore view the methods as complementary operating points rather than a strict ranking of operator families.

Limitations. The 25×25 deployed receptive field can miss distant re-identification, as quantified by 72.9 $\mathcal{J}\&\mathcal{F}$ on LVOS v2 versus 75.8 for SAM 2 and 78.2 for SAM 2.1. The object-pointer EMA only partially compensates. Speed is established at one resolution, batch size one, and three desktop/data-center GPUs; edge devices, resolution scaling, and throughput-oriented batching remain open. Finally, the family theorem gives no finite-model error bound, and the missing factorial controls prevent causal attribution of the total gain to CCA or temporal supervision alone.