

Fast SAM2 with Text-Driven Token Pruning

Avilasha Mandal^{a,b}, Chaoning Zhang^{a,*}, Fachrina Dewi Puspitasari^a,
Xudong Wang^a, Jiaquan Zhang^a, Caiyan Qin^c, Guoqing Wang^a, Yang
Yang^a, Heng Tao Shen^a

^a*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, Sichuan, China*

^b*Department of Computer Science and Engineering, Indian Institute of Technology, Delhi, New Delhi, 110016, Delhi, India*

^c*School of Robotics and Advanced Manufacture, Harbin Institute of Technology, Shenzhen, 518055, Guangdong, China*

Abstract

Segment Anything Model 2 (SAM2), a vision foundation model has significantly advanced in prompt-driven video object segmentation, yet their practical deployment remains limited by the high computational and memory cost of processing dense visual tokens across time. The SAM2 pipelines typically propagate all visual tokens produced by the image encoder through downstream temporal reasoning modules, regardless of their relevance to the target object, resulting in reduced scalability due to quadratic memory–attention overhead. In this work, we introduce a **text-guided token pruning** framework that improves inference efficiency by selectively reducing token density prior to temporal propagation, without modifying the underlying segmentation architecture. Operating after visual encoding and before memory-based propagation, our method ranks tokens using a lightweight routing mechanism that integrates local visual context, semantic relevance derived from object-centric textual descriptions (either user-provided or automatically generated), and uncertainty cues that help preserve ambiguous or boundary-critical regions. By retaining only the most informative tokens for downstream processing, the proposed approach reduces redundant computation while maintaining segmentation fidelity. Extensive experiments across multiple challenging video segmentation benchmarks demonstrate that post-

*Corresponding author

Email address: chaoningzhang1990@gmail.com (Chaoning Zhang)

encoder token pruning provides a practical and effective pathway to efficient, prompt-aware video segmentation, achieving up to **42.50%** faster inference and **37.41%** lower GPU memory usage compared to the unpruned baseline SAM2, while preserving competitive \mathcal{J} & \mathcal{F} performance. These results highlight the potential of early token selection to improve the scalability of transformer-based video segmentation systems for real-time and resource-constrained applications.

Keywords: Interactive video object segmentation, Token pruning, Segment Anything Model 2, Vision transformers, Text-guided segmentation

1. Introduction

Over the past few years, video object segmentation (VOS) has undergone a transformative shift [1, 2], driven by foundation models that combine large-scale pretraining, flexible generalization, and prompt-based control. Among these, **Segment Anything Model 2 (SAM 2)** [2] has emerged as a strong engine for interactive video object segmentation (iVOS). SAM 2 supports sparse user inputs such as clicks, boxes, or masks and propagates object masks across long video sequences with high temporal consistency. Architecturally, it builds upon Vision Transformers (ViTs) [3, 4] and a prompt-guided mask decoder inherited from SAM [5]. Unlike its image-only predecessor, SAM 2 introduces a memory mechanism that stores and reuses visual information across frames, enabling prompt-sensitive segmentation suitable for robotics, medical workflows, and real-time video editing.

However, SAM 2 and other memory-based VOS systems [7, 8, 9, 10, 11, 12] suffer from a fundamental inefficiency: they treat all visual tokens with equal importance. ViT-based encoders tokenize each frame into dense grids of local patches, which are processed by multi-headed self-attention [13] and deep transformer stacks. In SAM 2, these tokens are further stored and repeatedly accessed by the memory engine across time. Regardless of whether a token corresponds to static background, ambiguous boundaries, or semantically relevant foreground, it is carried through the same attention-heavy propagation pipeline. As a result, memory usage and inference latency increase substantially with the number of stored tokens, creating a bottleneck for latency-sensitive and resource-constrained settings.

A straightforward way to address this issue would be to sparsify [14] the image encoder itself. However, the SAM 2 encoder is typically used in a

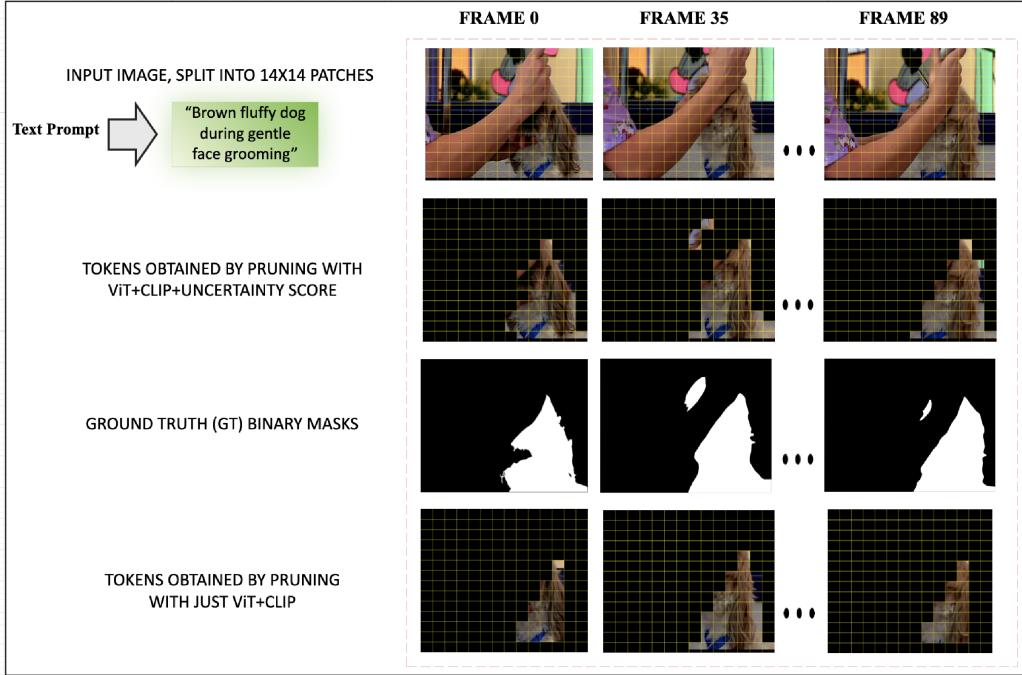


Figure 1: Qualitative visualisation of retained tokens with the text-driven token pruning approach (on UV dataset) [6] before passing into SAM2 decoder

frozen, pretrained form, and modifying its internal token flow would require retraining or architectural changes. In this work, we therefore take a different approach: rather than altering the encoder, we propose a **post-encoder, text-driven token pruning framework** (as illustrated in Figure. 2) that operates *between* the image encoder and the memory/mask-decoder stack. Our method selectively filters the encoder’s output tokens before they enter SAM 2’s memory and temporal attention modules, yielding substantial computational and memory savings where they matter most, while remaining non-invasive to the backbone.

To decide which tokens should be retained, we incorporate two additional sources of information alongside the visual embeddings: **semantic cues derived from text** [31] and **model uncertainty over ambiguous regions** [15, 16, 17]. Intuitively, as illustrated in Figure. 1 only a subset of tokens is relevant to a given segmentation intent (e.g., “segment the brown fluffy dog during gentle face grooming”), while other regions may require preservation due to visual ambiguity even if they are weakly aligned semantically. We

therefore evaluate tokens using three complementary signals—semantic relevance, predictive uncertainty, and visual context—which are fused to retain only the most informative subset per frame. This strategy reduces redundant memory accumulation, improves inference speed, and preserves segmentation accuracy over long temporal horizons.

Semantic relevance is obtained by aligning each visual token with a compact text embedding that reflects the intended object or region of interest. This text may be provided directly by a user, or automatically generated when no explicit instruction is available. In the latter case, we employ a lightweight two-stage procedure in which a vision–language model proposes an object-centric description from a coarse region of interest, followed by a refinement step that distills it into a concise phrase. We emphasize that automatically generated text does not guarantee perfect alignment with user intent; rather, it serves as a semantic prior. Accordingly, we study robustness to vague, partial, or overly verbose prompts and show that the pruning mechanism remains stable under such noise (shown in table 6).

Predictive uncertainty is estimated via Monte Carlo Dropout [16, 15, 17] applied to intermediate transformer layers. Tokens that exhibit high variability across stochastic forward passes typically correspond to edges, occlusions, motion blur, or visually confusing regions—precisely those that should not be pruned aggressively. We empirically select intermediate layers that balance local detail and contextual information [39, 40, 15], and we include ablations that isolate the contribution of uncertainty (table 2) and vary the number of Monte Carlo passes to quantify both accuracy gains and efficiency trade-offs (table 5).

The resulting token scores are fused using a lightweight MLP [42, 44] that ranks tokens by importance. Only the top- k tokens per frame are retained and passed to the SAM 2 memory and mask decoder. Importantly, SAM 2 already operates on a variable-length set of memory tokens, so reducing this sequence length requires no architectural modification. For initialization, we follow the standard iVOS protocol and use a single positive click (point) in the first frame containing the target object. In practical scenarios, this click may be provided by a user or computed automatically; for objects with complex shapes or holes, we adopt a distance-transform-based representative click (point) rather than a naive geometric centroid, which better reflects typical human interactions.

Compared to prior token pruning methods [18, 19], our approach is the first to integrate semantic alignment and uncertainty modeling into token se-

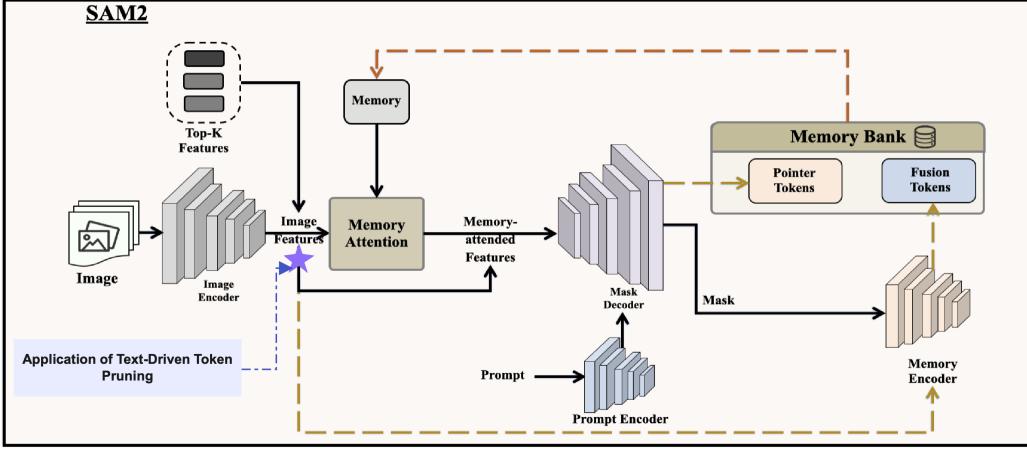


Figure 2: Qualitative visualization steps of segmentation with our text-driven token pruning approach atop SAM2

lection for a memory-based iVOS pipeline. Earlier approaches typically rely on heuristic saliency measures, operate on single images, or require retraining the backbone. In contrast, our method is *training-free at segmentation time*: the projection operators used for semantic and uncertainty alignment are obtained via closed-form fitting on encoder features. The only gradient-based optimization occurs in the lightweight MLP, which routes the semantically aligned tokens and prunes irrelevant ones, after the image encoder and before entering the memory stack for segmentation. This makes the proposed framework lightweight, modular, and easy to integrate with existing SAM 2-based systems.

We evaluate our approach on five challenging benchmarks—UVOS [6], PUMAVOS [21, 20], EndoVis [22, 23], VOST [24], and LVOSv2 [25, 26]—and compare against strong VOS baselines including STM, AOT, DeAOT, XMem, Cutie, and SAM 2. Across datasets, we observe up to 37.41% **reduction in GPU memory usage** and 42.50% **faster inference** relative to SAM 2, while preserving competitive \mathcal{J} & \mathcal{F} accuracy. Additional analyses examine the trade-off between token retention rate and performance, sensitivity to uncertainty estimation, and robustness to imperfect prompts and out-of-distribution imagery.

Our key contributions are summarized as follows:

- Our primary goal is not to introduce a new segmentation architecture,

but to improve the deployability, efficiency, and robustness of foundation segmentation models in **real-world video settings**.

- We propose a text-guided token pruning framework for interactive video object segmentation that integrates **semantic alignment, uncertainty estimation, and visual context** to rank and select task-relevant tokens prior to temporal propagation.
- Our approach is a modular, post image encoder pruning design that can be seamlessly integrated into SAM 2 **reducing memory and attention overhead** in the downstream propagation and decoding stages.

2. Related Works

In this section, we review prior work in interactive video object segmentation, foundation models for image and video segmentation, and token-level pruning for efficient Vision Transformers. We aim to situate our method within these lines of research, clarifying both architectural context and practical constraints that motivate token pruning between the encoder and propagation stages of SAM 2.

2.1. Interactive Video Object Segmentation (*iVOS*)

Interactive video object segmentation (*iVOS*) refers to identifying and propagating object masks across video frames with sparse human input such as clicks, boxes, or partial masks. Classical approaches such as STM [10], AOT [11], and XMem [7] use spatio-temporal memory mechanisms to match query frames against stored object embeddings. These methods achieve strong performance but often require repeated interaction or complex memory updates.

Foundation models have reshaped this landscape. SAM [5] introduced prompt-driven segmentation for images, while SAM 2 [2] extended this paradigm to videos through a long-range memory engine. SAM 2 supports both interactive refinement and semi-automatic propagation, but it inherits a computational bottleneck: its memory stack stores and attends to *dense* token grids from every processed frame, regardless of their informativeness. Recent on-device variants such as EdgeTAM [9] and lightweight propagation frameworks focus on architectural compression or approximated attention mechanisms, yet they preserve dense token flows.

2.2. Foundation Models for Efficient VOS

A parallel research thread aims to make VOS systems more computationally efficient. Variants such as DeAOT [12], Cutie [8], and efficient long-term trackers reduce propagation overhead using compact memories or hierarchical matching. While these models reduce computation in the propagation module, they still rely on dense spatial token grids at the feature-encoding stage. None explicitly address the redundancy in token sets *before* memory insertion, which becomes significant in long videos where memory grows linearly with time.

Our method is complementary to these approaches: instead of redesigning SAM 2’s [2] architecture, we intervene at the token level by filtering encoder outputs before they reach the memory bank. Thus, our pruning strategy can be combined with many of these efficient propagation methods.

2.3. Token Pruning in Vision Transformers

Vision Transformers tokenize inputs into spatial patches and apply quadratic-cost attention over all tokens [3]. As a result, pruning or compressing tokens has become a popular strategy for accelerating both classification and dense prediction models. Early methods such as DynamicViT [27] and Token-Learner [28] learn to identify salient tokens, while later approaches focus on task-specific regimes such as detection or instance segmentation [19, 29]. These works demonstrate that informative tokens often occupy a small subset of the spatial grid.

However, few gaps remain. First, most pruning methods require *retraining the backbone*, which is incompatible with SAM 2 where frozen encoder weights are typically used. Second, existing pruning strategies operate primarily on visual saliency or attention magnitude; they do not incorporate semantic intent from prompts or predictive uncertainty to preserve ambiguous regions.

Recent multimodal pruning methods [30, 32] investigate text-guided filtering for image segmentation, but they do not address video memory, do not integrate uncertainty, and are not directly compatible with SAM 2’s inference interfaces.

iVOS foundations (SAM 2 and its predecessors) provide strong segmentation quality but propagate dense tokens through costly memory operations; efficient-VOS methods reduce propagation cost but do not address token redundancy; and token pruning literature introduces token sparsity but assumes retraining or lacks semantic and uncertainty conditioning. Our work

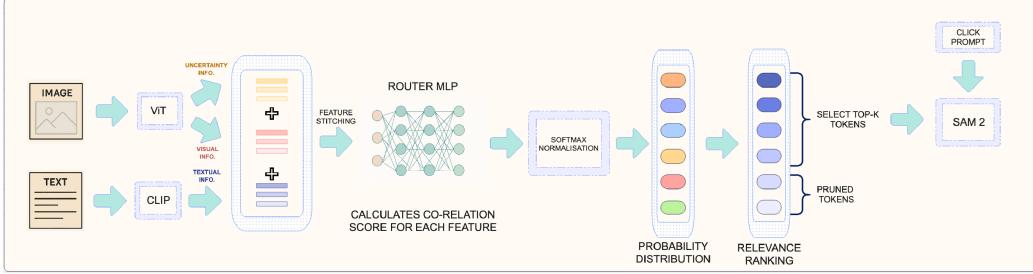


Figure 3: End-to-end pipeline of our text-driven token pruning framework atop SAM2. Semantic cues from text prompts, visual context from ViT tokens, along with token uncertainty are fused by a lightweight router to retain only task-relevant tokens before SAM2 decoding.

bridges these lines by proposing a *post-encoder, text-guided and uncertainty-aware* token pruning module compatible with SAM 2’s architecture, intended to reduce the computational footprint of its memory and decoding stack without modifying the encoder or requiring fine-tuning.

3. Method

3.1. Overview

Our goal is to accelerate SAM 2 by reducing the number of visual tokens processed in its *memory* and *mask-decoder* pathways, which constitute the dominant inference-time bottlenecks. Importantly, our method does *not* prune tokens inside the ViT-Hiera encoder—which remains dense and frozen for accurate multi-headed self attention—but instead intervenes *after* the encoder and *before* tokens are inserted into the memory engine. As illustrated in Fig. 3, the proposed module computes semantic relevance, predictive uncertainty, and visual-context features for all tokens, ranks them using a lightweight scoring network, and selects the top- k tokens per frame:

$$X_{\text{pruned}} = \text{TopK}(f_\theta(X_{\text{ViT}}, e_{\text{text}}, U)). \quad (1)$$

The pruned tokens are used throughout SAM 2’s propagation and decoding stack without requiring architectural modification, since the memory engine already supports variable-length token sequences.

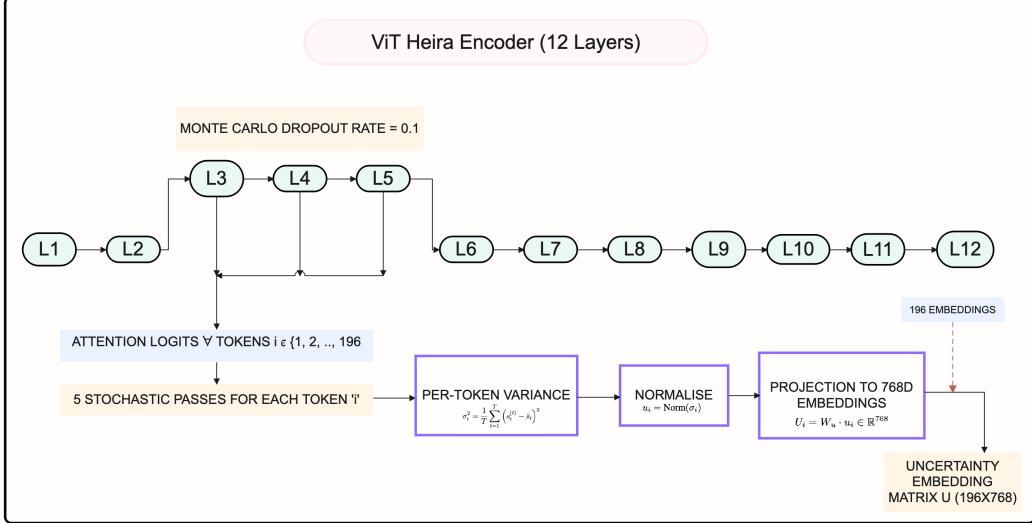


Figure 4: Monte Carlo (MC) Dropout applied to ViT-Hiera layers L3–L5 for uncertainty prediction per token

3.2. Visual Token Extraction from SAM 2 Encoder

Each frame I_t is processed by SAM 2’s frozen ViT-Hiera encoder, producing a dense token grid:

$$X_{\text{ViT}} = \text{ViT-Hiera}(I_t) \in \mathbb{R}^{N \times d_v}, \quad N = 14 \times 14 = 196. \quad (2)$$

Here, $X_{\text{ViT}} \in \mathbb{R}^{N \times d_v}$ denotes the visual token embeddings produced by the SAM 2 image encoder, with $N = 196$ and $d_v = 768$, the typical dimension of ViT visual tokens. These token embeddings are the only feature maps available to all downstream SAM 2 modules; we therefore prune strictly *after* this stage.

3.3. Semantic Prompt Extraction Without Ground Truth

In realistic deployment, ground-truth masks are unavailable to specify segmentation intent. Thus, we eliminate ground-truth dependency in the prompt-generation step. A user may optionally provide a short textual instruction. When no user-provided text is available, we obtain a coarse region of interest using a class-agnostic object proposal mechanism. Specifically, we deploy a Region Proposal Network (RPN), Faster R-CNN [33], to generate a set of candidate bounding boxes, each associated with an objectness

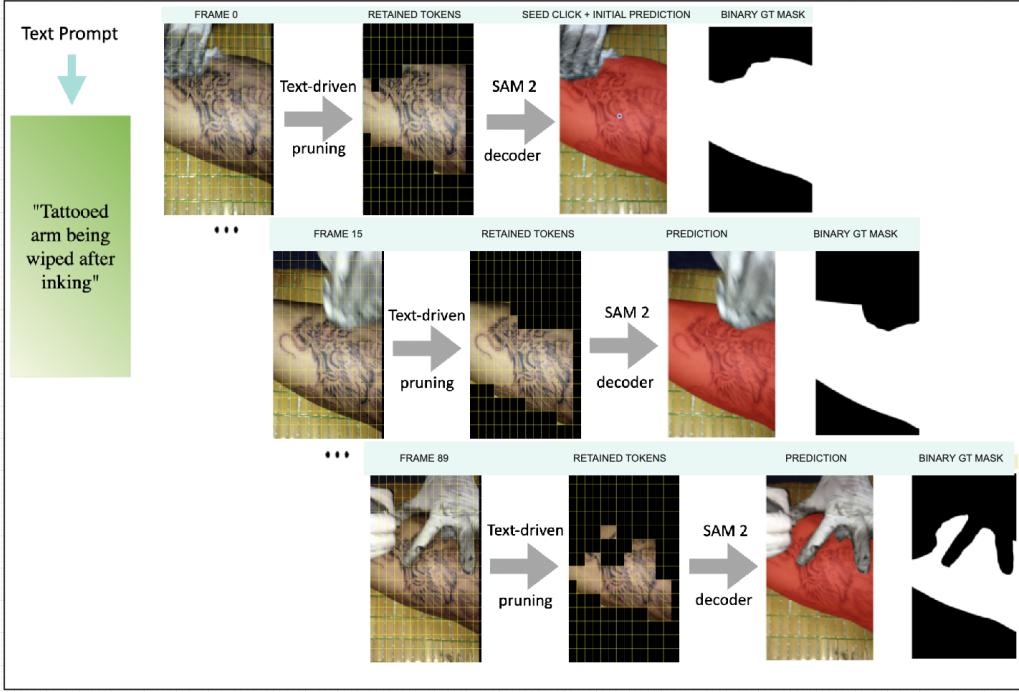


Figure 5: Qualitative segmentation results on a UVO dataset [6] sequence with our text-driven token pruning approach atop SAM2. We achieve mean $\mathcal{J}\&\mathcal{F}$ of 97.95% with no other refinement clicks required other than seed click to serve as input to SAM2’s prompt encoder. The image shows the dense tokens in the input frame, followed by semantically relevant tokens at 30% retention, and the segmentation mask by SAM2, with the ground truth also shown for comparison.

score. We select the proposal with the highest objectness score as the foreground region of interest in the first frame. A vision–language model such as, Large Language and Vision Assistant (LLaVA) [34, 35, 36] generates a coarse caption for the region, as in case of fig 1 we obtained description as, "A small fluffy dog with brown fur is being gently held by a person’s hand holding grooming tools around it’s face during grooming". This caption is then refined by a small language model, Bidirectional Encoder Representations from Transformers (BERT) [37, 38] into a concise phrase, "Brown fluffy dog during gentle face grooming" (prompt used in fig 1). This provides a lightweight semantic prior, with minimal +0.5s overhead, when no user intent is available in a realistic deployment (practically feasible with any underlying segmentation pipeline).

We encode the prompt using a frozen CLIP [31] text encoder:

$$e_{\text{text}} = f_{\text{CLIP}}(\mathcal{P}) \in \mathbb{R}^{d_t}. \quad (3)$$

where, \mathcal{P} denotes the input text prompt; $f_{\text{CLIP}}(\cdot)$ is a frozen CLIP text encoder; $e_{\text{text}} \in \mathbb{R}^{d_t}$ is the resulting text embedding with $d_t = 512$. To match SAM 2's visual token dimensionality ($d_v = 768$), we compute a *training-free* least-squares projection:

$$W_t = \arg \min_{W \in \mathbb{R}^{d_t \times d_v}} \|X_{\text{ViT}} W - e_{\text{text}}\|_2^2, \quad (4)$$

where, W_t is training-free least-squares projection matrix; e_{text} is the text vector broadcast across tokens. This projection is computed once per video, requires no labels, and preserves the training-free nature of the method. The aligned semantic embedding is:

$$e'_{\text{text}} = W_t^\top e_{\text{text}} \in \mathbb{R}^{768}. \quad (5)$$

3.4. Token-Level Uncertainty Estimation

Ambiguous regions such as occlusions, motion blur, or thin structures are precisely estimated where aggressive pruning is harmful. To preserve such regions, we estimate predictive uncertainty via Monte Carlo Dropout (MCD) [15] applied to intermediate encoder layers, as shown in Figure. 4. Following empirical studies on ViTs, we apply dropout to layers 3–5 [39, 40, 15], which offer strong spatial detail while maintaining contextual awareness.

For each of T stochastic forward passes, we extract the pre-softmax attention logits $s_i^{(t)}$ for token i :

$$s_i^{(t)} = f_{\text{attn}}^{(t)}(X_{\text{ViT}}), \quad t = 1, \dots, T. \quad (6)$$

Uncertainty is quantified as variance:

$$\sigma_i^2 = \frac{1}{T} \sum_{t=1}^T \left(s_i^{(t)} - \bar{s}_i \right)^2, \quad \bar{s}_i = \frac{1}{T} \sum_{t=1}^T s_i^{(t)}. \quad (7)$$

where, σ_i^2 denotes the variance-based uncertainty estimate for token i , \bar{s}_i denotes the mean attention logit for token i over all T stochastic passes.

The standard deviation σ_i values are normalized across tokens using min–max normalization:

$$\tilde{\sigma}_i = \frac{\sigma_i - \min_j \sigma_j}{\max_j \sigma_j - \min_j \sigma_j}. \quad (8)$$

where, $\tilde{\sigma}_i$ is its normalized form

After normalization, uncertainty values are projected to token space using a second least-squares projection:

$$W_u = \arg \min_W \|X_{ViT} W - \tilde{\sigma}_i\|_2^2, \quad U_i = W_u^\top \tilde{\sigma}_i. \quad (9)$$

where, W_u is training-free least-squares projection matrix; U_i is the uncertainty feature aligned to the visual token space. This projection also requires no training signals. We further benchmark $T \in \{4, 5, 6\}$ to quantify the efficiency–accuracy trade-off.

3.5. Fused Token Representation

For token i , we fuse visual features, aligned semantic embedding, and uncertainty features:

$$h_i = [X_{ViT,i}; e'_{text}; U_i] \in \mathbb{R}^{3d_v}. \quad (10)$$

where, h_i denotes the fused token representation

3.6. Token Scoring and Pruning

Each fused descriptor is passed through a lightweight two-layer MLP:

$$s_i = \text{MLP}(h_i), \quad \text{MLP} : 2304 \rightarrow 256 \rightarrow 1. \quad (11)$$

[Note that the input size to the MLP is 2304 vectors, as we have 3 signals (visual, semantic, and uncertainty) each aligned to 768 dimensions now.] Scores are softmax-normalized:

$$\alpha_i = \frac{\exp(s_i)}{\sum_{j=1}^N \exp(s_j)}. \quad (12)$$

where, s_i and α_i are the raw and normalized token importance scores, respectively. We retain the top- k tokens:

$$X_{\text{pruned}} = \{X_{ViT,i} \mid i \in \text{TopK}(\alpha, k)\}. \quad (13)$$

SAM 2’s memory accepts variable-length sequences, so no architectural modification is required. The pruned tokens simply replace the full set of 196 tokens for memory writing and mask decoding.

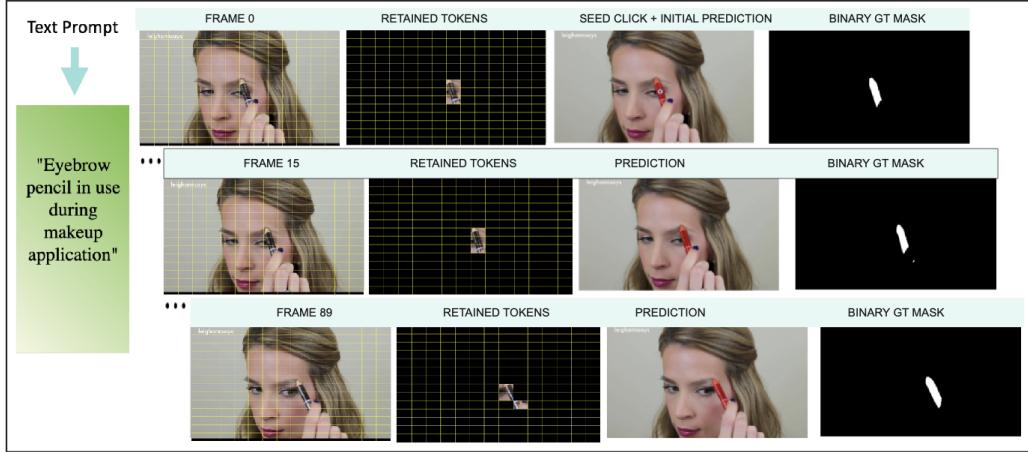


Figure 6: Qualitative segmentation results on a UV dataset [6] sequence with our text-driven token pruning approach atop SAM2. On this video sequence, we achieve mean $\mathcal{J}\&\mathcal{F}$ of 91.84% and minimum $\mathcal{J}\&\mathcal{F}$ of 73.85% with just 4 refinement clicks required including the seed click to serve as input to SAM2’s prompt encoder. The image shows the dense tokens in the input frame, followed by semantically relevant tokens at 30% retention, and the segmentation mask by SAM2, with the ground truth binary mask also shown for qualitative comparison.

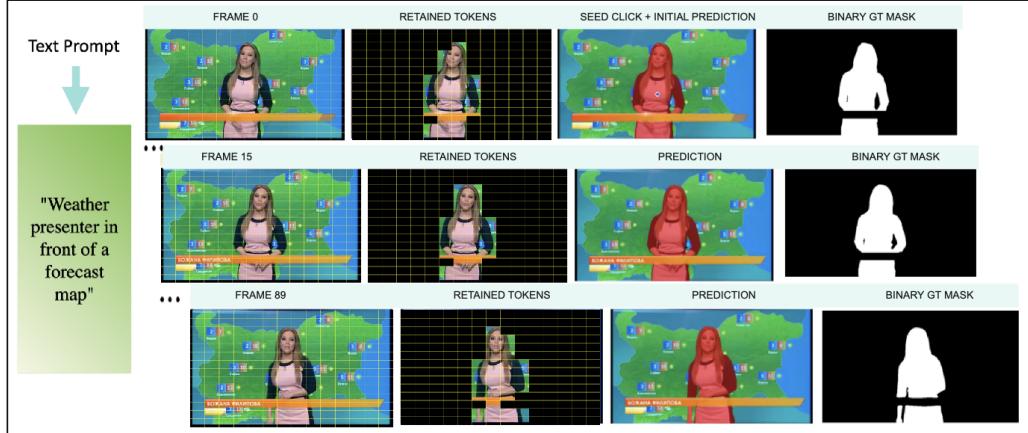


Figure 7: Qualitative segmentation results on a UV dataset [6] sequence with our text-driven token pruning approach atop SAM2. On this video sequence, we achieve mean $\mathcal{J}\&\mathcal{F}$ of 96.95% and minimum $\mathcal{J}\&\mathcal{F}$ of 96.01% with no other refinement clicks required other than the seed click to serve as input to SAM2’s prompt encoder. The image shows the dense tokens in the input frame, followed by semantically relevant tokens at 30% retention, and the segmentation mask by SAM2, with the ground truth binary mask also shown for qualitative comparison.

3.7. Integration With SAM 2 Memory and Prompt Encoder

A single positive click is required as input for the prompt encoder to initialize SAM 2. We propose an initialization click prompt algorithm either as (i) a user click or (ii) a distance-transform–based representative point for irregular shapes (e.g., torus-like objects). This avoids incorrect geometric centroid selection.

Let \mathbf{p} denote the initial click, which is embedded via SAM 2’s prompt encoder:

$$z_{\text{prompt}} = f_{\text{prompt}}(\mathbf{p}). \quad (14)$$

At each frame, the mask is produced by SAM 2’s decoder:

$$M_t = f_{\text{decoder}}(X_{\text{pruned}}, z_{\text{prompt}}, \mathcal{M}_{1:t-1}), \quad (15)$$

where $\mathcal{M}_{1:t-1}$ denotes the memory bank.

Because the memory engine accumulates only pruned tokens rather than full dense grids, its computational and memory cost reduce proportionally.

3.8. Optional Interactive Refinement

Although our method does not require high human-in-the-loop support, SAM 2 inherently allows optional manual refinement. If $\mathcal{J} \& \mathcal{F}$ drops below a threshold, a synthetic or user-provided click can be added, leading to semi-automatic propagation, core in interactive video object segmentation. We observe empirically that pruning reduces drift and therefore reduces refinement calls.

4. Experiments

All experiments are conducted using an NVIDIA GeForce RTX 3090 GPU. SAM2 is initialized using official ViT-Hiera (tiny) checkpoint [4]. Experiments are conducted using images resized to an input resolution of 224×224 . The visual encoder (ViT-Hiera [4]) produces 14x14 spatial tokens, each with a feature dimension of 768. Textual prompts distilled from LLaVA [34] and BERT [37] are projected into a 512-dimensional embedding space. To estimate uncertainty, we perform $T = 5$ stochastic forward passes using Monte Carlo dropout applied in layers 3–5 of SAM2’s visual encoder. The router MLP [41, 42, 43, 44] used for token scoring consists of two linear layers separated by a GELU [45, 46] activation. Following scoring, we retain the top

Algorithm 1: Fast SAM2 with Text-Driven Token Pruning

Input: Video frames $\{I_t\}_{t=1}^L$; optional text prompt \mathcal{P} ; token budget k ;
MC passes $T = 5$

Output: Segmentation masks $\{M_t\}_{t=1}^L$
 Encode text prompt using frozen CLIP:;
 $e_{\text{text}} \leftarrow f_{\text{CLIP}}(\mathcal{P})$;
 Compute semantic alignment:;
 $W_t \leftarrow \arg \min_W \|X_{\text{ViT}} W - e_{\text{text}}\|_2^2$;
 $e'_{\text{text}} \leftarrow W_t^\top e_{\text{text}}$;
for $t = 1$ **to** L **do**
 | $X_{\text{ViT}} \leftarrow \text{ViT-Hiera}(I_t)$;
 | **for** $\tau = 1$ **to** T **do**
 | | $s_i^{(\tau)} \leftarrow f_{\text{attn}}^{(\tau)}(X_{\text{ViT}})$;
 | | $\sigma_i^2 \leftarrow \frac{1}{T} \sum_{\tau} (s_i^{(\tau)} - \bar{s}_i)^2$;
 | | $\tilde{s}_i \leftarrow \frac{\sigma_i - \min_j \sigma_j}{\max_j \sigma_j - \min_j \sigma_j}$;
 | | $W_u \leftarrow \arg \min_W \|X_{\text{ViT}} W - \tilde{s}\|_2^2$;
 | | $U_i \leftarrow W_u^\top \tilde{s}_i$;
 | | $h_i \leftarrow [X_{\text{ViT},i}; e'_{\text{text}}; U_i]$;
 | | $s_i \leftarrow \text{MLP}(h_i)$;
 | | $\alpha_i \leftarrow \frac{\exp(s_i)}{\sum_j \exp(s_j)}$;
 | | $X_{\text{pruned}} \leftarrow \text{TopK}(\alpha, k)$;
 | | $M_t \leftarrow f_{\text{decoder}}(X_{\text{pruned}}, z_{\text{prompt}}, \mathcal{M}_{1:t-1})$;
return $\{M_t\}_{t=1}^L$

30% of tokens for downstream propagation. For interactive segmentation, we initialize SAM2 with a single positive point prompt (user initialized or automated) located at the centroid of the object of interest in the first frame that contains this object. This serves as the initial condition for the memory encoder. In our experiments, we allow a maximum of 10 refinements rounds per sequence with 90 frames (including starter click), with up to 3 clicks allowed in each refinement. Figures. 5, 6, 7 illustrate the **qualitative visualisation** of results on several videos by leveraging our text-driven token pruning to accelerate SAM2.

Best and second-best results are highlighted in **green** and **yellow** shades respectively, in all of tables 1-6.

4.1. Evaluation Metrics

We adopt the standard $\mathcal{J}\&\mathcal{F}$ metric [47, 48], which averages region similarity using the Jaccard index (\mathcal{J}) and contour accuracy using the F-measure (\mathcal{F}). Unless otherwise specified, scores are averaged over all annotated frames and all object IDs and reported per dataset.

$$\mathcal{J}\&\mathcal{F} = \frac{1}{2TO_t} \sum_{t=1}^T \sum_{o=1}^{O_t} \left(\frac{|S_{t,o} \cap G_{t,o}|}{|S_{t,o} \cup G_{t,o}|} + \frac{2P_{t,o}^c R_{t,o}^c}{P_{t,o}^c + R_{t,o}^c} \right) \quad (16)$$

where S , G , P^c , R^c , T , and O_t refer to the output mask, ground-truth mask, precision between output and ground-truth contours, recall between output and ground-truth contours, number of frames, and number of objects in each frame, respectively.

4.2. Main Results

To evaluate the performance of our method, we conduct experiments on several benchmark datasets with dense annotation (all sets of PUMaVOS [20] and EndoVis2018 [23], as well as validation sets of VOST [24], UVO [6], and LVOsV2 [26]). We also include the evaluation of the baseline SAM2 and five prior SOTA methods in VOS task, STM [10], AOT [11], DeAOT [12], XMem [7], and Cutie [8]. For simplicity, we only perform segmentation on objects that appear in the first frame. Table 1 and figure 8 show that in terms of inference speed and memory consumption, our method yield competitive results against prior SOTA models (averaged) by an average frames per second (FPS) increase of **31.7%** and average GPU memory usage reduction of **46.3%**. Our acceleration method only causes a minor drop or consistent results in the segmentation performance.

The reduced token count leads to a **42.50%** inference speedup and lowers GPU memory usage by **37.41%** over SAM2. Averaging over all sequences from all datasets used in Table 1, our method requires **2.6** simulated clicks per sequence compared to 4.2 clicks required in SAM2 baseline showing minimal human-in-the-loop requirement. Hence pruning the noisy information helps keep the $\mathcal{J}\&\mathcal{F}$ scores consistently above a threshold of 80%, falling below which click-prompts are leveraged.

4.3. Ablation Studies

4.3.1. Semantic and Uncertainty Contributions.

In table 2, The observed performance gains on the UVO dataset [6] (averaged over all sequences) demonstrates that incorporating both semantic

Table 1: Comparison with SOTA methods across datasets.

Dataset	Method	$\mathcal{J} \uparrow$	$\mathcal{F} \uparrow$	$\mathcal{J} \& \mathcal{F} \uparrow$	FPS \uparrow	VRAM (GB) \downarrow	Refinements
UVO [6]	SAM2 [2]	0.614	0.987	0.805	17.9	2.35	4
	STM [10]	0.706	0.988	0.847	18.4	1.39	—
	AOT [11]	0.768	0.837	0.802	23.0	1.67	—
	DeAOT [12]	0.761	0.832	0.797	22.2	1.87	—
	XMem [7]	0.706	0.988	0.847	24.1	1.70	—
	Cutie [8]	0.701	0.981	0.841	21.3	1.86	—
	Ours	0.813	0.903	0.858	30.5	1.26	2
PUMaVOS [20]	SAM2 [2]	0.891	0.985	0.938	17.6	2.27	4
	STM [10]	0.448	0.981	0.714	5.1	1.81	—
	AOT [11]	0.878	0.914	0.896	6.9	1.93	—
	DeAOT [12]	0.880	0.914	0.897	6.3	2.05	—
	XMem [7]	0.873	0.983	0.928	24.5	1.64	—
	Cutie [8]	0.878	0.980	0.929	21.9	1.78	—
	Ours	0.838	0.987	0.912	24.6	1.22	3
EndoVis [23]	SAM2 [2]	0.894	0.981	0.938	16.2	8.90	5
	STM [10]	0.809	0.984	0.897	11.5	9.21	—
	AOT [11]	0.752	0.832	0.792	13.8	8.18	—
	DeAOT [12]	0.742	0.825	0.783	13.2	8.58	—
	XMem [7]	0.879	0.979	0.929	20.5	7.82	—
	Cutie [8]	0.879	0.907	0.893	18.7	8.34	—
	Ours	0.834	0.989	0.911	21.6	7.63	3
VOST [24]	SAM2 [2]	0.582	0.967	0.775	14.6	1.84	4
	STM [10]	0.327	0.976	0.651	9.25	7.41	—
	AOT [11]	0.822	0.983	0.902	15.71	9.31	—
	DeAOT [12]	0.942	0.959	0.951	16.03	10.42	—
	XMem [7]	0.674	0.970	0.822	18.5	1.52	—
	Cutie [8]	0.686	0.977	0.831	18.7	8.34	—
	Ours	0.719	0.981	0.850	22.3	1.18	3
LVOSv2 [26]	SAM2 [2]	0.625	0.969	0.797	12.8	2.01	4
	STM [10]	0.489	0.981	0.735	10.4	8.37	—
	AOT [11]	0.822	0.983	0.902	15.8	8.78	—
	DeAOT [12]	0.913	0.939	0.926	13.55	8.81	—
	XMem [7]	0.701	0.974	0.838	16.7	1.63	—
	Cutie [8]	0.886	0.957	0.922	18.7	8.34	—
	Ours	0.829	0.981	0.905	19.6	1.12	2

relevance and uncertainty estimation into the token pruning process leads to consistent improvements over using semantic cues alone, as visually ambiguous or occluded regions are no longer prematurely pruned.

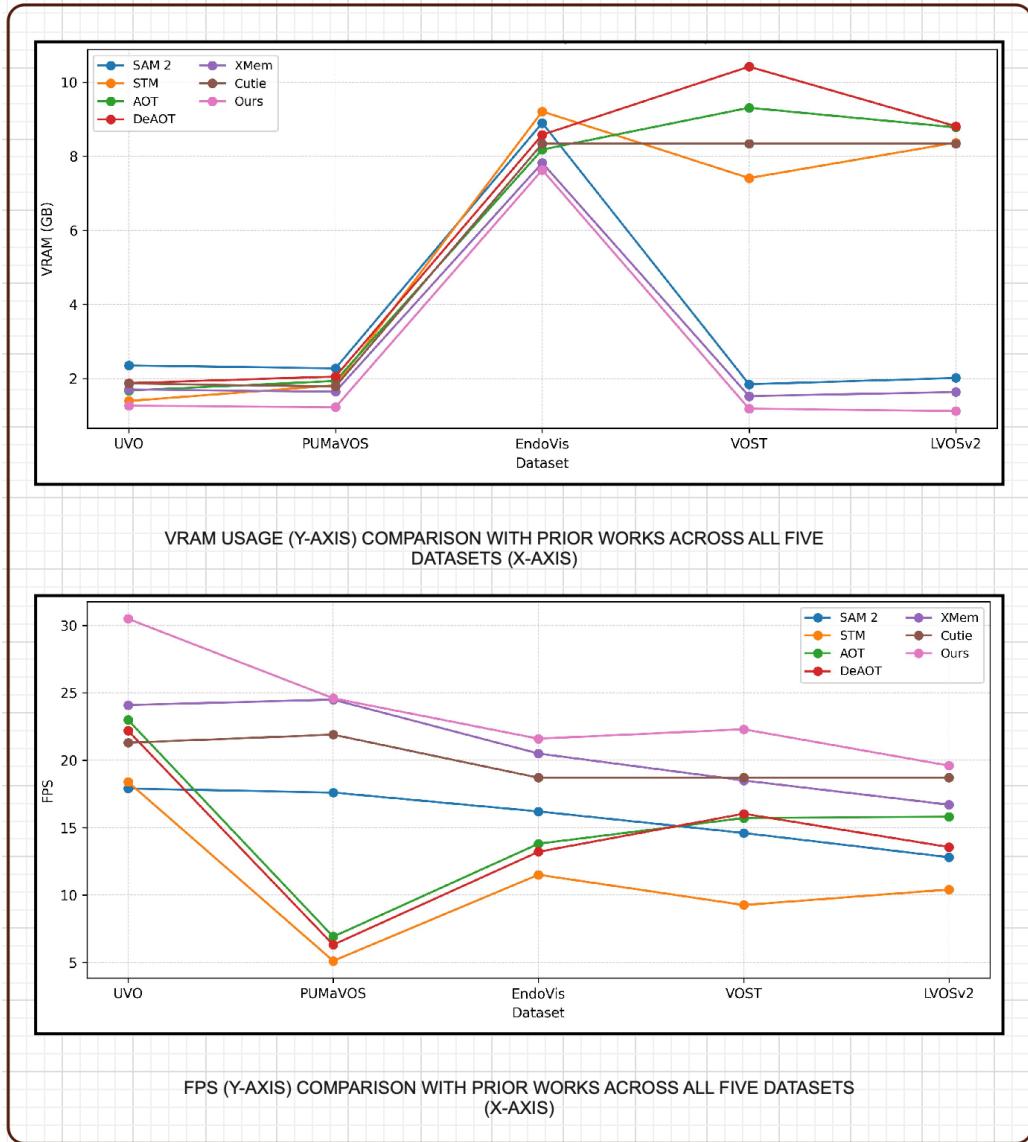


Figure 8: Efficiency comparison across five video object segmentation benchmarks. Top: GPU memory consumption (VRAM). Bottom: inference speed (FPS). Our text-guided token pruning consistently reduces memory usage and improves runtime efficiency compared to prior methods, including SAM2, without altering the underlying segmentation architecture. The results demonstrate that selectively propagating only semantically and visually relevant tokens substantially improves scalability across diverse datasets.

Table 2: Signal-wise ablation on UVO.

Signal Used	$\mathcal{J} \& \mathcal{F} \uparrow$	Retained Tokens
Text Prompt Only	83.9	57
Text Prompt + Uncertainty	85.8	59

4.3.2. ConvNeXt for Local Precision

We append a frozen ConvNeXt [49] backbone alongside the ViT encoder, as shown in Fig 9 to capture finer local spatial patterns, which might be ignored by ViT’s which are more inclined on capturing the macroscopic features and their embeddings lack the finer parts below a certain scale, even with a hierarchical scale facility, as in ViT-Hiera. Table 3 shows minor yet consistent improvements, on UVO [6] (averaged over all sequences).

Table 3: Effect of ConvNeXt integration (on UVO)

Visual Encoder	$\mathcal{J} \& \mathcal{F} \uparrow$	Overhead (ms)
ViT only	85.8	–
ViT + ConvNeXt	86.0	+18.2

4.4. Token Retention Sensitivity

We evaluate pruning aggressiveness experiment on UVO [6] (averaging over all sequences) by varying token retention rates. As shown in Table 4, optimal balance is found near 30%, which suppresses distractors while maintaining semantic coverage.

Table 4: Varying token retention and its effect.

Retention (%)	$\mathcal{J} \& \mathcal{F} \uparrow$	FPS \uparrow
100 (No pruning)	85.6	17.9
50	85.7	23.2
30	85.8	30.5
10	84.9	42.8

4.5. Monte Carlo Pass Sensitivity

We analyze the effect of the number of Monte Carlo Dropout passes T used for uncertainty estimation on UVO [6], averaging results over all sequences. As shown in Table 5, increasing T improves uncertainty stability and segmentation accuracy up to a point, after which additional passes introduce diminishing returns while incurring higher computational cost. We observe that $T = 5$ offers the best trade-off between segmentation quality and inference efficiency, and therefore adopt it as the default setting.

Table 5: Effect of Monte Carlo passes T on uncertainty estimation.

MC Passes (T)	$\mathcal{J} \& \mathcal{F} \uparrow$	FPS \uparrow
4	85.5	32.1
5	85.8	30.5
6	85.8	27.6

4.6. Robustness to Automated and Human Prompts

We study the sensitivity of our pruning mechanism to different sources and qualities of textual prompts. In practical deployment, text prompts may be provided directly by a user or automatically generated (hence found vague at some random generation). Importantly, the text signal in our framework is not intended to perfectly encode user intent, but rather to act as a semantic prior that guides token selection. We therefore evaluate three representative prompt settings: (i) an accurate automated prompt generated from an object-centric region description, (ii) a vague automated prompt with some semantic specificity, and (iii) a human-provided prompt. As shown in Table 6, all three settings yield similar segmentation accuracy for the video sequence shown in fig. 1, indicating that the pruning strategy remains stable even under noisy or under-specified semantic guidance.

Table 6: Effect of prompt specificity and source on segmentation accuracy and efficiency (UVO dataset, averaged over all sequences).

Prompt Type	$\mathcal{J} \& \mathcal{F} \uparrow$
(i) brown fluffy dog during gentle face grooming	85.80
(ii) puppy being handled	85.78
(iii) dog being groomed	85.80

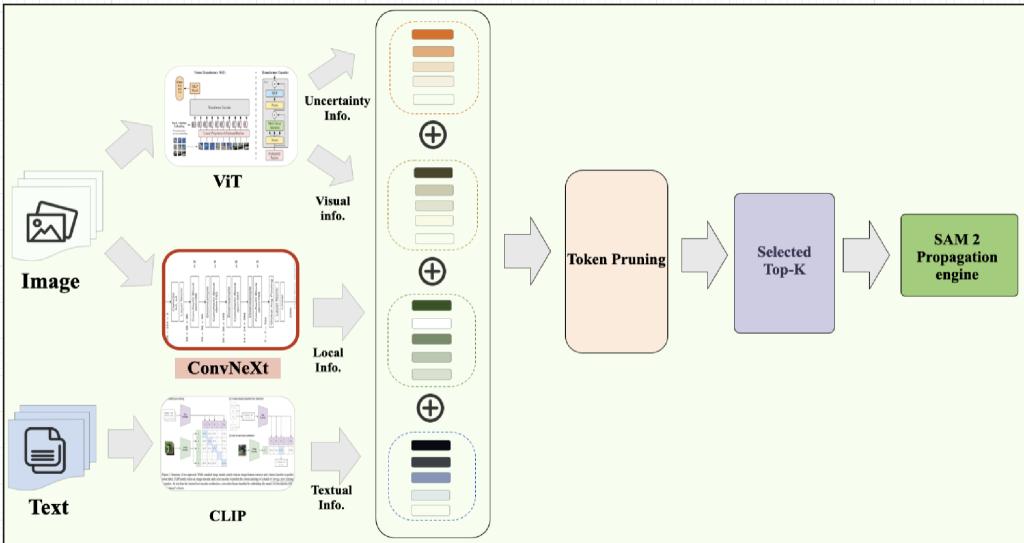


Figure 9: A frozen ConvNeXt [49] backbone appended alongside the ViT encoder to capture fine-grained local patterns

5. Discussion and Future Work

Our work presents a text-driven token pruning approach to accelerate SAM2 as the vision foundation model for video object segmentation. We reduce visual tokens produced by the SAM2 vision encoder to only retain tokens relevant to the object being prompted. We argue that eliminating redundant information at the beginning of the segmentation pipeline can prevent the model from carrying the computational burden of these tokens throughout the entire segmentation process. This is reflected in our result in Table 1, which shows that our method consistently yields competitive acceleration on the inference speed against prior SOTA models. Additionally, in terms of segmentation performance, our method only causes a minor drop on the baseline SAM2. We argue that this capability is the effect of the elimination of the noisy information from redundant visual tokens at the beginning of the pipeline, as well as the restoration of a few ambiguous tokens through Monte Carlo dropout. Moreover, as our method does not perform any modification to the VOS model, we argue that our method can be integrated into other methods that attempt to accelerate the inference speed of the SAM2 (e.g., EfficientTAM [50, 51]). We leave the exploration on this direction for future work.

6. Conclusion

We presented **Fast SAM2 with text-driven token pruning** technique applied post image encoder and pre-memory in SAM2, reducing compute in the propagation stack while preserving accuracy. By integrating semantic relevance, uncertainty, and visual context, we obtain substantial improvements in terms of inference speed and GPU memory efficiency with minimal architectural overhead. Our expanded evaluations and robustness studies demonstrate the viability of token pruning as a practical acceleration strategy for real-world video segmentation pipelines.

7. Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 62572104). The first author conducted this research during a summer internship at the University of Electronic Science and Technology of China (UESTC), Chengdu, China.

References

- [1] Caelles, S., Maninis, K.-K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D. and Van Gool, L., “One-Shot Video Object Segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 221–230, 2017.
- [2] Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L. et al., “Sam 2: Segment Anything in Images and Videos,” arXiv preprint arXiv:2408.00714, 2024.
- [3] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al., “An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale,” arXiv preprint arXiv:2010.11929, 2020.
- [4] Ryali, C., Hu, Y.-T., Bolya, D., Wei, C., Fan, H., Huang, P.-Y., Aggarwal, V., Chowdhury, A., Poursaeed, O., Hoffman, J. et al., “Hiera: A Hierarchical Vision Transformer Without the Bells-and-Whistles,” in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 29441–29454, 2023.

- [5] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y. et al., “Segment Anything,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, 2023.
- [6] Wang, W., Feiszli, M., Wang, H. and Tran, D., “Unidentified Video Objects: A Benchmark for Dense, Open-World Segmentation,” arXiv preprint arXiv:2104.04691, 2021.
- [7] Cheng, H. K. and Schwing, A. G., “XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model,” in *European Conference on Computer Vision (ECCV)*, pp. 640–658, 2022.
- [8] Cheng, H. K., Oh, S. W., Price, B., Lee, J.-Y. and Schwing, A., “Putting the Object Back into Video Object Segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3151–3161, 2024.
- [9] Zhou, C., Zhu, C., Xiong, Y., Suri, S., Xiao, F., Wu, L., Krishnamoorthi, R., Dai, B., Loy, C. C., Chandra, V. et al., “EdgeTAM: On-Device Track Anything Model,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13832–13842, 2025.
- [10] Oh, S. W., Lee, J.-Y., Xu, N. and Kim, S. J., “Video Object Segmentation Using Space-Time Memory Networks,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9226–9235, 2019.
- [11] Yang, Z., Wei, Y. and Yang, Y., “Associating Objects with Transformers for Video Object Segmentation,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 2491–2502, 2021.
- [12] Yang, Z. and Yang, Y., “Decoupling Features in Hierarchical Propagation for Video Object Segmentation,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 36324–36336, 2022.
- [13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I., “Attention Is All You Need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

- [14] Luo, W., Li, J., Yang, J., Xu, W. and Zhang, J., “Convolutional Sparse Autoencoders for Image Classification,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 7, pp. 3289–3294, 2018.
- [15] Kendall, A., Gal, Y., and Cipolla, R., “Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7482–7491, 2018.
- [16] Kendall, A., Badrinarayanan, V. and Cipolla, R., “Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding,” arXiv preprint arXiv:1511.02680, 2015.
- [17] Liu, K., Price, B., Kuen, J., Fan, Y., Wei, Z., Figueroa, L., Geras, K. and Fernandez-Granda, C., “Uncertainty-Aware Fine-Tuning of Segmentation Foundation Models,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 53317–53389, 2024.
- [18] Tang, Q., Zhang, B., Liu, J., Liu, F. and Liu, Y., “Dynamic Token Pruning in Plain Vision Transformers for Semantic Segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 777–786, 2023.
- [19] Liu, Y., Gehrig, M., Messikommer, N., Cannici, M. and Scaramuzza, D., “Revisiting Token Pruning for Object Detection and Instance Segmentation,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2658–2668, 2024.
- [20] Bekuzarov, M., Bermudez, A., Lee, J.-Y. and Li, H., “XMem++: Production-Level Video Segmentation from Few Annotated Frames,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 635–644, Oct. 2023.
- [21] Bekuzarov, M., Bermudez, A., Lee, J.-Y. and Li, H., “XMem++: Production-Level Video Segmentation from Few Annotated Frames,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 635–644, 2023.
- [22] Allan, M., Kondo, S., Bodenstedt, S., Leger, S., Kadkhodamohammadi, R., Luengo, I., Fuentes, F., Flouty, E., Mohammed, A., Pedersen, M.,

- et al.*, “2018 Robotic Scene Segmentation Challenge,” *arXiv preprint arXiv:2001.11190*, 2020.
- [23] Allan, M., Mcleod, J., Wang, C., Rosenthal, J. C., Hu, Z., Gard, N., Eisert, P., Fu, K. X., Zeffiro, T., Xia, W. et al., “Stereo Correspondence and Reconstruction of Endoscopic Data Challenge,” *arXiv preprint arXiv:2101.01133*, 2021.
 - [24] Tokmakov, P., Li, J. and Gaidon, A., “Breaking the ‘Object’ in Video Object Segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22836–22845, 2023.
 - [25] Hong, L., Liu, Z., Chen, W., Tan, C., Feng, Y., Zhou, X., Guo, P., Li, J., Chen, Z., Gao, S., *et al.*, “LVOS: A Benchmark for Large-Scale Long-Term Video Object Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
 - [26] Hong, L., Liu, Z., Chen, W., Tan, C., Feng, Y., Zhou, X., Guo, P., Li, J., Chen, Z., Gao, S. *et al.*, “LVOS: A Benchmark for Large-Scale Long-Term Video Object Segmentation,” *arXiv preprint arXiv:2404.19326*, 2024.
 - [27] Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J. and Hsieh, C.-J., “DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification,” *arXiv preprint arXiv:2106.02034*, 2021.
 - [28] Ryoo, M. S., Piergiovanni, A. J., Arnab, A., Dehghani, M. and Angelova, A., “TokenLearner: Adaptive Space-Time Tokenization for Videos,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
 - [29] Liu, Y., Gehrig, M., Messikommer, N., Cannici, M. and Scaramuzza, D., “Revisiting Token Pruning for Object Detection and Instance Segmentation,” *arXiv preprint arXiv:2306.07050*, 2023.
 - [30] Chen, H., Ni, Y., Huang, W., Liu, Y., Jeong, S., Wen, F., Bastian, N. D., Latapie, H. and Imani, M., “VLTP: Vision-Language Guided Token Pruning for Task-Oriented Segmentation,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 9353–9363, 2025.

- [31] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., *et al.*, “Learning Transferable Visual Models from Natural Language Supervision,” in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 8748–8763, 2021.
- [32] Shrivastava, A., Selvaraju, R. R., Naik, N. and Ordonez, V., “CLIP-Lite: Information Efficient Visual Representation Learning with Language Supervision,” in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 8433–8447, 2023.
- [33] Ren, S., He, K., Girshick, R., and Sun, J., “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, pp. 91–99, 2015.
- [34] Liu, H., Li, C., Wu, Q. and Lee, Y. J., “Visual Instruction Tuning,” in *Advances in Neural Information Processing Systems*, vol. 36, pp. 34892–34916, 2023.
- [35] Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P. and Yuan, L., “Video-LLaVA: Learning United Visual Representation by Alignment Before Projection,” arXiv preprint arXiv:2311.10122, 2023.
- [36] Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P. and Yuan, L., “Video-LLaVA: Learning United Visual Representation by Alignment Before Projection,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5971–5984, 2024.
- [37] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.
- [38] Koroteev, M. V., “BERT: A Review of Applications in Natural Language Processing and Understanding,” arXiv preprint arXiv:2103.11943, 2021.
- [39] Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A., “Do Vision Transformers See Like Convolutional Neural Networks?”,

in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 12116–12128, 2021.

- [40] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B., “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022, 2021.
- [41] Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J. et al., “MLP-Mixer: An All-MLP Architecture for Vision,” *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 24261–24272, 2021.
- [42] Rumelhart, D. E., Hinton, G. E. and Williams, R. J., “Learning Representations by Back-Propagating Errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [43] Haykin, S., *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, 1994.
- [44] Almeida, L. B., “Multilayer Perceptrons,” in *Handbook of Neural Computation*, pp. C1–2, CRC Press, 2020.
- [45] Hendrycks, D., “Gaussian Error Linear Units (GELUs),” *arXiv preprint arXiv:1606.08415*, 2016.
- [46] Zhang, Q., Wang, C., Wu, H., Xin, C. and Phuong, T. V., “GELU-Net: A Globally Encrypted, Locally Unencrypted Deep Neural Network for Privacy-Preserved Learning,” in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3933–3939, 2018.
- [47] Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A. and Van Gool, L., “The 2017 DAVIS Challenge on Video Object Segmentation,” *arXiv preprint arXiv:1704.00675*, 2017.
- [48] Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M. and Sorkine-Hornung, A., “A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 724–732, 2016.

- [49] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T. and Xie, S., “A ConvNet for the 2020s,” arXiv preprint arXiv:2201.03545, 2022.
- [50] Xiong, Y., Zhou, C., Xiang, X., Wu, L., Zhu, C., Liu, Z., Suri, S., Varadarajan, B., Akula, R., Iandola, F. et al., “Efficient Track Anything,” arXiv preprint arXiv:2411.18933, 2024.
- [51] Xiong, Y., Zhou, C., Xiang, X., Wu, L., Zhu, C., Liu, Z., Suri, S., Varadarajan, B., Akula, R., Iandola, F., et al., “Efficient Track Anything,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11513–11524, 2025.