YOUR VIT IS SECRETLY ALSO A VIDEO SEGMENTATION MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing online video segmentation models typically combine a per-frame segmenter with complex specialized tracking modules. While effective, these modules introduce significant architectural complexity and computational overhead. Recent studies suggest that plain Vision Transformer (ViT) encoders, when scaled with sufficient capacity and large-scale pre-training, can conduct accurate image segmentation without requiring such specialized components. Motivated by this observation, we propose the *Video Encoder-only Mask Transformer (VidEoMT)*, a simple encoder-only video segmentation model that eliminates the need for dedicated tracking modules. To enable temporal modeling in an encoder-only ViT, VidEoMT introduces a lightweight query fusion mechanism that merges queries from the previous frame with temporally-agnostic learned queries, enabling information propagation across frames while preserving adaptability to new content. As a result, VidEoMT attains the benefits of a tracker without added complexity and achieves competitive accuracy, while being 5–10$\times$ faster, running at up to 160 FPS with a ViT-L backbone. Code will be made public upon acceptance.

1 INTRODUCTION

Progress in computer vision has long been driven by the introduction of architectural components that perform dedicated visual processing steps. Over time, this layering of components has produced powerful but increasingly complex systems. The *bitter lesson* (Sutton, 2019) reminds us that such hand-designed components can accelerate progress in the short term, yet in the long run, it is scalability and data that most effectively unlock new capabilities. Simplicity is often enabled by re-examining prior insights in the light of how they can be mapped to a simpler, but scalable architecture. This motivates asking not what more can be added, but what can be removed, and whether strong performance can emerge from simplicity and scale rather than complexity.

Recent work illustrates this point. Kerssies et al. (2025) demonstrated that image segmentation can be performed accurately with a highly simplified architecture by adding only a few learnable queries to a plain Vision Transformer (ViT) (Dosovitskiy et al., 2021) with a method called EoMT. They showed that the complex, task-specific components used by prior Mask2Former-style architectures (Cheng et al., 2022) become almost completely redundant when using a large ViT model and large-scale pre-training, such as DINOv2 (Oquab et al., 2024).

This result motivates us to formulate a more profound hypothesis. We argue that, rather than explaining the observed simplification successes by redundancy, we should interpret them as an indication that a sufficiently large and well-pretrained ViT becomes a *general vision model* that can *learn to take over the functionalities of downstream model components*. If this is indeed the case, it would open up an interesting direction for further research, as we could then expect the ViT to also learn to take over the capabilities of *other downstream components*.

In this paper, we take testing this hypothesis one step further. Specifically, we study whether it is possible to obtain a simplified ViT-inspired architecture for the more complex task of *video segmentation*, which we take as an umbrella term for a large variety of video-level segmentation and tracking tasks. Video segmentation requires models to *localize* and *segment* objects, to *classify* them, and to *track* the same instances across frames. Current methods obtain state-of-the-art performance by combining specialized components that improve one or more of those capabilities (Lee et al., 2025; Zhang et al., 2023; 2025; Zhou et al., 2024). As video segmentation has many potential ap-

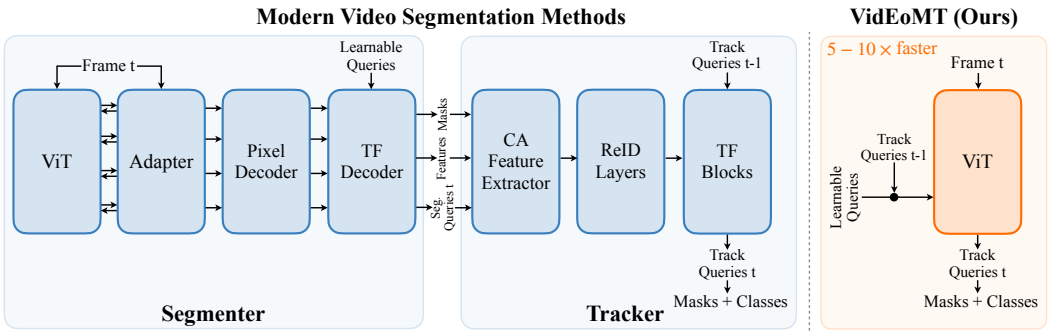


Figure 1: **Modern Video Segmentation Methods vs. VidEoMT (Ours).** We compare the architectures of modern video segmentation methods – using CAVIS (Lee et al., 2025) as a representative example – and our encoder-only VidEoMT method. VidEoMT streamlines the video segmentation framework, relying on the power of large-scale pre-training with vision foundation models rather than handcrafted task-specific components.

plications for mobile or edge devices, efficient processing with lightweight models is critical. This makes video segmentation an ideal testing ground for our endeavor.

Specifically, we start from state-of-the-art video segmentation models, and evaluate the effect of removing specialized components. These models all follow roughly the same paradigm: they first apply a *segmenter*, which predicts segmentation masks and class labels for individual video frames and outputs object-level feature queries, and then use a *tracker* to match object-level feature queries across different video frames. As shown in Figure 1 (left) for the example of CAVIS (Lee et al., 2025), both the segmenter and the tracker consist of many specialized components. Our first step is replacing the complex segmenter with EoMT, followed by eliminating context-aware feature extraction and re-identification layers and fine-tuning EoMT to take over their roles. Simplifying further by naively discarding the tracker and applying EoMT frame-by-frame leads to a large accuracy drop, showing that temporal modeling cannot simply be removed. We therefore move away from the conventional decoupling of segmenter and tracker and instead ask whether temporal modeling can be embedded directly into the ViT encoder. To achieve this, we introduce *query fusion*, a lightweight but crucial mechanism that allows the encoder to propagate and adapt queries across frames. The resulting model, *Video Encoder-only Mask Transformer* (VidEoMT), unifies segmentation and temporal association inside a single encoder, as illustrated in Figure 1 (right).

By no longer requiring complex specialized components and performing all computations within a single ViT-style model, VidEoMT is remarkably efficient. Through experiments, we find that VidEoMT with a ViT-Large backbone is over $10\times$ faster than existing state-of-the-art methods, achieving processing speeds of up to 160 FPS on the YouTube-VIS benchmarks (Yang et al., 2019). Importantly, this speed is obtained while maintaining a comparable accuracy. We further validate our findings on other related benchmarks and tasks, including long-term video instance segmentation, video panoptic segmentation, and video semantic segmentation. In all cases, VidEoMT achieves similar speedups of $5\times$-$10\times$ with negligible negative impact on accuracy. Such speed-up factors can be a veritable game changer for applications, enabling online video processing across a wide range of use cases. Apart from this immediate value, these results also strongly support our hypothesis that a sufficiently large and well-pretrained ViT is a general vision model that can learn to take over the functionality of downstream components, as the ViT in VidEoMT has learned to acquire the capabilities of the many task-specific components that are part of modern video segmentation methods. The qualifiers *sufficiently large* and *well-pretrained* are important here, as we do not observe this effect as prominently with smaller models or weaker pre-training.

In summary, we make the following contributions: (1) We demonstrate that a sufficiently large, pre-trained ViT can learn to take over the functionality of specialized downstream components for video segmentation. (2) We propose VidEoMT, a simple and highly efficient architecture for video segmentation. VidEoMT enhances EoMT with a novel, lightweight temporal modeling mechanism, *Query Fusion*, to unify segmentation and temporal association within a single ViT-style encoder. (3) We show that the resulting VidEoMT, with its simple encoder-only architecture, can perform video segmentation at accuracies comparable to the state of the art, while being up to $10\times$ faster.

2 Related Work

Image Segmentation. Image segmentation requires that objects in an image are segmented and classified. Early image segmentation models treated this task as a *per-pixel classification* problem, predicting a class label for each pixel (Chen et al., 2018a;b; Long et al., 2015). Later works propose an alternative *mask classification* approach, where a model predicts a *segment* – consisting of a segmentation mask and class label – for each object in the image (Cheng et al., 2021). These mask classification methods typically make use of Mask Transformers, which leverage image features from a backbone and learnable queries to predict a segmentation mask and class label for each query with a Transformer decoder (Wang et al., 2021; Cheng et al., 2022; Jain et al., 2023; Cavagnero et al., 2024). Recently, EoMT (Kerssies et al., 2025) has demonstrated that it is possible to conduct accurate image segmentation without requiring a decoder or other task-specific components, by simply feeding the learnable queries directly into a large, pre-trained ViT encoder. In this work, inspired by EoMT, we investigate whether video segmentation models can be simplified in a similar manner, with the goal of improving efficiency while preserving high accuracy.

Video Segmentation. Video segmentation is a well-established computer vision task, encompassing video instance segmentation (VIS) (Yang et al., 2019), video panoptic segmentation (VPS) (Kim et al., 2020), and video semantic segmentation (VSS) (Nilsson & Sminchisescu, 2018), where the primary objective is to segment, classify, and track all objects of interest in a video. Current VIS, VPS, and VSS methods are typically Mask Transformer-based architectures (Heo et al., 2022; Huang et al., 2022; Zhang et al., 2023; 2025; Lee et al., 2025; Zhou et al., 2024; Weng et al., 2023; Shin et al., 2024). They extend Mask Transformers for image segmentation (Cheng et al., 2022) into the video domain by incorporating specialized tracking components or enhancing temporal representations. The most recent methods (Huang et al., 2022; Zhang et al., 2023; 2025; Lee et al., 2025; Zhou et al., 2024) are typically universal models, which can handle VIS, VPS, and VSS within a single framework. They follow a decoupled paradigm, where the segmentation and tracking sub-tasks are separated. First, the segmenter conducts image segmentation for each frame, and then a tracker associates these segmented objects over time. Generally, both the segmenter and the tracker contain various specialized components, which increase accuracy but reduce efficiency. In this work, we consider these universal video segmentation models and demonstrate that they can be simplified to an *encoder-only* design, significantly improving efficiency while achieving competitive accuracy.

3 Method

3.1 Task Definition

We consider the task of *online video segmentation*, where the goal is to assign a class label and a binary segmentation mask to every object entity appearing in each frame of a video, while maintaining temporal consistency of object identities across time.

Formally, a video is a sequence of T frames $\mathcal{V} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_T\}, \quad \mathbf{I}_t \in \mathbb{R}^{3\times H\times W}$, with spatial resolution (H, W). For each frame $\mathbf{I}_t$, a model should yield a set of K_t predictions $\mathcal{Y}_t = \{(\mathbf{m}_{t,i}, c_{t,i})\}_{i=1}^{K_t}$, where $\mathbf{m}_{t,i} \in \{0,1\}^{H\times W}$ is a binary segmentation mask, and $c_{t,i} \in \{1, \ldots, C\}$ is a semantic category label from C classes. Additionally, these predictions should be matched across different frames, applying tracking. The task must be solved in an *online* manner: at timestep t, predictions $\mathcal{Y}_t$ may only depend on the current frame $\mathbf{I}_t$ and earlier frames $\{\mathbf{I}_1, \ldots, \mathbf{I}_{t-1}\}$.

3.2 Preliminaries

Modern online video segmentation models (Zhang et al., 2023; 2025; Lee et al., 2025; Zhou et al., 2024) typically decouple the video segmentation pipeline into two distinct sub-tasks, *segmentation* and *tracking*. First, a *segmenter* $\mathcal{S}$ generates frame-level image segmentation predictions, and outputs refined image features $\mathbf{F}$ and object query embeddings $\mathbf{Q}^{\mathcal{S}} = \{\mathbf{q}_i^{\mathcal{S}} \in \mathbb{R}^D\}_{i=1}^{K}$. Each of these queries represents one object in the frame. Second, a *tracker* $\mathcal{T}$ aligns these representations from consecutive frames, associating the object queries across time and producing temporally aligned queries $\mathbf{Q}^{\mathcal{T}} = \{\mathbf{q}_i^{\mathcal{T}} \in \mathbb{R}^D\}_{i=1}^{K}$ which can be used to yield consistent segmentation masks and object classes for each entity in a video (see Figure 1, left).

Segmenter. The segmenter performs per-frame image segmentation, producing image-level fine-grained features $\mathbf{F}$ and query embeddings $\mathbf{Q}^{\mathcal{S}}$. For this purpose, existing state-of-the-art models

combine a pre-trained ViT (Dosovitskiy et al., 2021; Oquab et al., 2024), a ViT-Adapter (Chen et al., 2023), and a Mask Transformer-based segmentation decoder (Cheng et al., 2022).

The ViT embeds an input image $\mathbf{I}_t$ into non-overlapping patch tokens and processes them with L Transformer blocks. A CNN-based ViT-Adapter augments the encoder with multi-scale features, which are fused and refined in the Mask2Former head by a pixel decoder, producing a set of enriched features $\{\mathbf{F}_4, \mathbf{F}_8, \mathbf{F}_{16}, \mathbf{F}_{32}\}$, with $\mathbf{F}_i \in \mathbb{R}^{D\times(H/i)\times(W/i)}$. A Transformer decoder then updates K learnable queries $\mathbf{Q}^{\text{lrn}} = \{\mathbf{q}_i^{\text{lrn}} \in \mathbb{R}^D\}_{i=1}^K$ through cross- and self-attention, yielding refined queries $\mathbf{Q}^{\mathcal{S}}$. The segmenter then predicts a class label $\mathbf{c}_i \in \mathbb{R}^C$ and a mask $\mathbf{m}_i \in \mathbb{R}^{(H/4)\times(W/4)}$ for the object that each query represents, also leveraging the refined image features.

Tracker. To associate the per-frame object queries generated by the segmenter between different frames, most state-of-the-art approaches employ a separate tracker. This specialized component operates in an online fashion, by jointly processing per-frame object queries from consecutive frames. Concretely, it processes $\mathbf{Q}_t^{\mathcal{S}}$, the output from the segmenter for the current frame, and $\mathbf{Q}_{t-1}^{\mathcal{T}}$, the temporally updated queries from the previous frame:

$$\mathbf{Q}_t^{\mathcal{T}} = \mathcal{T}(\mathbf{Q}_t^{\mathcal{S}}, \mathbf{Q}_{t-1}^{\mathcal{T}}). \tag{1}$$

In practice, the tracker mainly consists of N Transformer blocks with cross-attention, self-attention, and feed-forward layers. The current-frame embeddings act as keys and values, while the temporally refined embeddings from the previous frame, $\mathbf{Q}_{t-1}^{\mathcal{T}}$, serve as queries in cross-attention. Through these Transformer layers, the tracker aligns past queries with present ones, ultimately yielding temporally consistent queries $\mathbf{Q}_t^{\mathcal{T}}$ that represent the objects in frame t, for which the ordering is the same as in the queries for the previous frame, $\mathbf{Q}_{t-1}^{\mathcal{T}}$. This is how temporal association is ensured. Finally, the tracker predicts a segmentation mask and class label for each query $\mathbf{Q}_t^{\mathcal{T}}$, as in the segmenter.

Context-Aware Features. CAVIS (Lee et al., 2025) introduces *context-aware features* $\mathbf{Q}_t^{\mathcal{A}} = \{\mathbf{q}_{t,i}^{\mathcal{A}} \in \mathbb{R}^D\}_{i=1}^K$ to enrich query embeddings $\mathbf{Q}_t^{\mathcal{S}}$ with information from the local neighborhood of each object, before they enter the tracker's Transformer blocks. Concretely, given predicted masks $\mathbf{M}_t = \{\mathbf{m}_{t,i}\}_{i=1}^K$ and features $\mathbf{F}_{4,t}$ at timestep t, binary boundary maps $\mathbf{B}_{t,i} \in \{0,1\}^{(H/4)\times(W/4)}$ are extracted using a Laplacian filter. Next, the features $\mathbf{F}_{4,t}$ are smoothed with an average filter, yielding $\mathbf{F}_{4,t}^{\mathcal{A}}$. Finally, the context-aware features, extracted by pooling the smoothed features at boundary pixels, are concatenated with the per-frame query embeddings $\mathbf{Q}_t^{\mathcal{S}}$. This produces an enriched set of queries $\mathbf{Q}_t^{\mathcal{C}} = \{\mathbf{q}_{t,i}^{\mathcal{C}} \in \mathbb{R}^{2D}\}_{i=1}^K$, which are then fed into the tracker's Transformer blocks. Hence, in presence of context-aware features, the tracker $\mathcal{T}$ is now also a function of highest-resolution features $\mathbf{F}_{4,t}$:

$$\mathbf{Q}_t^{\mathcal{T}} = \mathcal{T}(\mathbf{Q}_t^{\mathcal{S}}, \mathbf{Q}_{t-1}^{\mathcal{T}}, \mathbf{F}_{4,t}). \tag{2}$$

Re-identification Layers. To further improve robustness, modern methods employ *re-identification layers*. These layers are commonly paired with contrastive objectives, which enforce similarity between embeddings of the same instance while separating those of different instances. In practice, query embeddings $\mathbf{Q}_t^{\mathcal{S}}$ from the segmenter are usually fed to the re-identification layers, implemented as a 3-layer `MLP`. In CAVIS, we apply this MLP to the context-aware queries $\mathbf{Q}_t^{\mathcal{C}}$:

$$\mathbf{Q}_t^{\mathcal{R}} = \texttt{MLP}(\mathbf{Q}_t^{\mathcal{C}}). \tag{3}$$

This yields enhanced queries $\mathbf{Q}_t^{\mathcal{R}}$ which are subjected to contrastive learning, and which are fed into the tracker's Transformer blocks, in place of the context-aware queries $\mathbf{Q}_t^{\mathcal{C}}$.

3.3 Removing Task-specific Components

Recently, EoMT (Kerssies et al., 2025) has challenged the dominant paradigm in image segmentation that uses many specialized components, showing that this task can be performed in an encoder-only fashion, given a sufficiently large ViT model and strong pre-training. Learned queries $\mathbf{Q}^{\text{lrn}}$ are injected into the last L_2 layers of a ViT encoder and processed jointly with patch tokens, yielding updated queries $\mathbf{Q}^{\mathcal{S}}$ and predictions $\{(\mathbf{c}_i, \mathbf{m}_i)\}_{i=1}^K$ without auxiliary decoders. Despite its simplicity, EoMT performs competitively with complex frameworks while greatly improving efficiency.

Inspired by this result, we explore a similar simplification for video segmentation, where inference speed is even more critical. Our hypothesis is that a strong ViT encoder can handle both segmentation and temporal association within a unified *encoder-only* architecture, removing the need for explicit tracking modules. To verify this, we start from the state-of-the-art CAVIS model, replace its

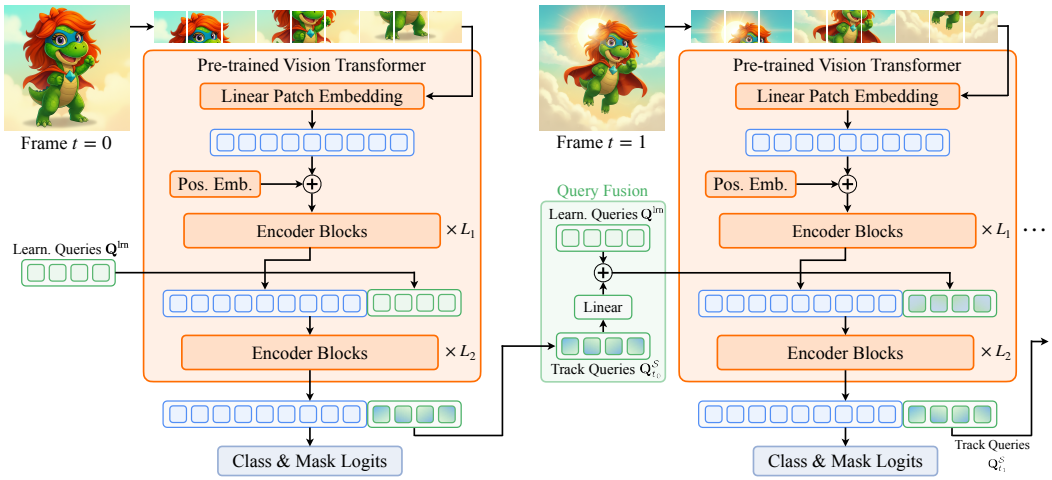


Figure 2: **VidEoMT architecture.** Given frame $t = 0$, initially learnable queries are concatenated to the patch tokens after the first L_1 ViT blocks. Both sets of tokens are then jointly processed in the last L_2 blocks, then track queries fed to the Query Fusion module for temporal propagation.

heavy segmenter with EoMT, and then progressively ablate video-specific components to evaluate whether the encoder can also learn to provide features sufficient for temporal association.

Replacing the Segmenter. In modern video segmentation models, such as CAVIS, the segmenter $\mathcal{S}$ is composed of an inefficient ViT-Adapter and a complex and resource-intensive Mask2Former pixel decoder and Transformer decoder. We replace the entire segmenter with EoMT, which integrates query tokens directly into the ViT and predicts object representations without specialized components, thereby greatly simplifying the pipeline and consistently improving inference speed.

Removing Context-Aware Features. The context-aware features in CAVIS explicitly encode information from the spatial neighborhood of each instance to stabilize predictions under appearance changes or occlusion. Extracting these features requires convolutional filtering over high-resolution features, repeated for every query in all frames of a video, making it inefficient. We hypothesize that the auxiliary context added by these features is not strictly necessary when leveraging a strong pre-trained ViT like DINOv2, as its features are already fine-grained enough to be easily fine-tuned to capture specific object identity and maintain stability under appearance changes or occlusion.

Removing Re-identification Layers. While effective, re-identification layers add complexity both at inference and during training, where the associated contrastive losses are memory-intensive and slow to optimize. We argue that with large-scale pre-training, such as with DINOv2, the features of the ViT encoder already contain rich instance-level information. Since the segmentation queries explicitly cross-attend to these features, they effectively inherit this instance-discriminative knowledge and preserve it across frames. Therefore, eliminating these layers not only simplifies the whole pipeline but also makes training more affordable and scalable.

3.4 VIDEOMT

After the previously described simplifications, the model consists of EoMT combined with a simplified tracker $\mathcal{T}$. The tracker matches object queries across frames and enforces temporal consistency, but at the cost of increased architectural complexity and significant computational overhead.

We hypothesize that strong pre-training, e.g., with DINOv2, already equips the ViT encoder with representations strong enough to support both per-frame segmentation and temporal association within the encoder itself, without specialized tracking modules. Hence, we move away from the conventional decoupling of segmenter and tracker and adopt a unified encoder-only design.

To equip an encoder-only model for temporal modeling, we propagate queries generated for individual frames and fuse these with learnable queries in a lightweight *Query Fusion* mechanism. The resulting model, which we name *Video Encoder-only Mask Transformer* (VidEoMT) (Figure 2), eliminates the need for dedicated tracking modules, while delivering comparable performance to state-of-the-art methods and a significantly faster inference.

Query Propagation. When the tracker is entirely removed, the model reduces to a purely image-level EoMT that processes each frame independently. In this case, queries $\mathbf{Q}^{\mathcal{S}}$ are the model's output, and there are no queries $\mathbf{Q}^{\mathcal{T}}$, as there is no longer a tracker.

The simplest way to reintroduce temporal modeling is through query propagation. At timestep $t = 0$, we feed the learnable queries $\mathbf{Q}^{\text{lrn}}$ to the last L_2 layers of the ViT as in EoMT. At subsequent timesteps t, however, we directly use the output queries from the previous timestep $t-1$, i.e., $\mathbf{Q}^{\mathcal{S}}_{t-1}$.

This strategy enables information to flow across time without additional computational cost, improving temporal consistency across frames. However, since we only provide the ViT with information from the previous frame, the influence of the learnable queries diminishes over time. As a result, the model tends to lose the ability to recognize objects that newly appear in the video, as the contribution of the learnable queries $\mathbf{Q}^{\text{lrn}}$ attenuates over time.

Query Fusion. To address this limitation, we introduce *Query Fusion*, illustrated in Figure 2. In this design, queries from the previous frame $\mathbf{Q}^{\mathcal{S}}_{t-1}$ are first transformed by a lightweight linear layer and then combined with the original learned queries $\mathbf{Q}^{\text{lrn}}$ through element-wise addition:

$$\mathbf{Q}^{\mathcal{F}}_{t} = \texttt{Linear}(\mathbf{Q}^{\mathcal{S}}_{t-1}) + \mathbf{Q}^{\text{lrn}}. \tag{4}$$

The element-wise addition is possible because the supervision strategy ensures that the ordering of the queries remains the same across frames in a video. This fusion ensures that the model has access to the temporal context from the past through $\mathbf{Q}^{\mathcal{S}}_{t-1}$, as well as learnable queries $\mathbf{Q}^{\text{lrn}}$ to enable adaptability to new objects in the current frame. This balance between information propagation and adaptability allows Query Fusion to capture the essential benefits of a tracker in an encoder-only fashion, providing temporal consistency without additional architectural complexity and while maintaining a high level of efficiency.

Training. VidEoMT is trained using the objective function as Mask2Former (Cheng et al., 2022). We use the cross-entropy loss for classification and the binary cross-entropy and Dice losses for segmentation predictions. To ensure temporally consistent supervision, we follow the ground-truth matching strategy of DVIS++ (Zhang et al., 2025). In practice, a ground-truth object is only matched to a query in the frame where the object first appears. In the remaining frames, the ground-truth object stays matched to this query, ensuring temporal consistency.

4 Experiments

Datasets and Evaluation Metrics. We evaluate VidEoMT on six major benchmarks for video segmentation: OVIS (Qi et al., 2022), YT-VIS 2019, 2020, and 2022 (Yang et al., 2019) for Video Instance Segmentation (VIS); VIPSeg (Miao et al., 2022) for Video Panoptic Segmentation (VPS); and VSPW (Miao et al., 2021) for Video Semantic Segmentation (VSS). We use the Average Precision (AP) and Average Recall (AR) metrics (Yang et al., 2019) for VIS, Video Panoptic Quality (VPQ) (Kim et al., 2020), Segmentation and Tracking Quality (STQ) (Weber et al., 2021) for VPS, and mean IoU (mIoU) and Video Consistency (VC) (Miao et al., 2021) for VSS.

Implementation Details. Similar to the state-of-the-art models CAVIS (Lee et al., 2025) and DVIS-DAQ (Zhou et al., 2024), we use a DINOv2-pretrained ViT (Oquab et al., 2024) as the backbone of VidEoMT. We adopt a batch size of 8 with 5 frames as a temporal window, using mixed precision and the AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of 10^{-4}. Following EoMT Kerssies et al. (2025), we apply layer-wise learning rate decay (LLRD) (Devlin et al., 2019) with a factor of 0.6 and a polynomial learning rate decay with a power of 0.9. The number of iterations and training video resolutions follow the settings of CAVIS (Lee et al., 2025) for fair comparison. For more implementation details, see Appendix A.

To assess computational efficiency, we measure both FPS and FLOPs. FPS is reported as the average number of images processed per second on the validation set with a batch size of 1, evaluated on an NVIDIA H100 GPU with FlashAttention v2 (Dao, 2024) and `torch.compile` (Ansel et al., 2024) (default settings) enabled. FLOPs are calculated using fvcore (Meta Research, 2023), averaged over all images in the validation set. We report the results in GFLOPs, i.e., FLOPs $\times 10^9$.

Step	Method	AP	Params	GFLOPs	FPS
(0)	CAVIS	68.9	358M	838	15
(1)	↳ w/ EoMT as *Segmenter*	68.1	328M	699	42
(2)	↳ w/o Context-aware Features	68.4	327M	581	72
(3)	↳ w/o Re-identification Layers	68.0	326M	580	74
(4)	↳ w/o Tracker ⇒ EoMT	61.3	316M	565	162
–	EoMT	61.3	316M	565	162
(5)	↳ w/ Propagation	63.9	316M	565	162
(6)	↳ w/ Fusion ⇒ **VidEoMT (ours)**	68.6	318M	566	160

Table 1: **From CAVIS to VidEoMT.** Stepwise removal of CAVIS modules toward EoMT, and modifications extending it to VidEoMT. Evaluated on YouTube-VIS 2019 val set (Yang et al., 2019).

5 Results

5.1 Main Results

From CAVIS to VidEoMT. In Table 1, we report a stepwise transformation from state-of-the-art video segmentation method CAVIS (Lee et al., 2025) to our proposed VidEoMT. We gradually remove specialized tracking modules to obtain the lightweight EoMT baseline, and we then introduce modifications to EoMT to support tracking. For more details see appendix A.1. In step (1), we find that replacing the segmenter with EoMT (Kerssies et al., 2025) improves FPS by almost **3**×, while AP drops by -0.8. In steps (2)–(3), we observe that removing *context-aware features* and the *re-identification layers* further increases speed by **1.8**× to 74 FPS, with almost no impact on accuracy. While the use of context-aware features facilitates faster convergence during training, we find that with sufficient training iterations the model can achieve comparable performance even without them. These results demonstrate that the DINOv2 ViT encoder can take over the functionality of these components without degrading performance. In step (4), we note that the elimination of the tracker, which results in the naive, per-frame application of EoMT, yields a speedup of more than **10**× to 160 FPS compared to CAVIS's 15 FPS, but suffers a substantial **-7.6** AP drop. Interestingly, though, even without any tracking modules and just relying on the queries, the model still retains reasonable accuracy. This shows that EoMT can learn to output objects in a somewhat consistent order across frames, despite processing them independently.

Applying query propagation in step (5), however, is necessary to introduce temporal modeling in EoMT, improving the AP by **+2.6** without increasing computational cost. However, the model may still struggle with identifying newly appearing objects over time. In the final step (6), we show that fusing the propagated queries with the original learned queries allows VidEoMT to recover nearly all of the original accuracy, while still maintaining a speedup of more than **10**× compared to CAVIS. Notably, the improvement in inference speed is much larger than in terms of FLOPs. This is the case because VidEoMT almost purely consists of a plain ViT encoder. As such, it can better leverage dedicated hardware and software optimizations for the Transformer architecture without being bottlenecked by complex specialized components.

Overall, these results show that VidEoMT achieves an excellent trade-off between efficiency and accuracy, as heavy modules in CAVIS can be safely removed, while our lightweight extensions to EoMT effectively restore performance with negligible computational cost. These results confirm our hypothesis that a DINOv2-pretrained VIT can be trained to conduct both segmentation and tracking within the same encoder, without requiring additional complex tracking components.

5.2 Comparison with State-of-the-Art Models

Video Instance Segmentation (VIS). We first compare VidEoMT with state-of-the-art models on the VIS task across four datasets. The results, reported in Tables 2 and 3, demonstrate that VidEoMT consistently outperforms DVIS (Zhang et al., 2023) and DVIS++ (Zhang et al., 2025), while being 5–8× faster. Compared to DVIS-DAQ (Zhou et al., 2024), VidEoMT achieves higher accuracy on all benchmarks except OVIS, where the gap is within 2 AP points. Similarly, VidEoMT surpasses CAVIS on YT-VIS 2022, and achieves comparable accuracy on YT-VIS 2019 and OVIS, and remains within 2 AP on YT-VIS 2021, while being over 10× faster in some cases. Finally, we note that VidEoMT is also both faster and more accurate than MinVIS (Huang et al., 2022), which was specifically designed for efficiency and simplicity. Overall, VidEoMT demonstrates a significantly superior accuracy *vs.* efficiency trade-off compared to existing approaches.

Method	Backbone	YouTube-VIS 2019 *val*					YouTube-VIS 2021 *val*				
		AP	AP_{75}	AR_{10}	GFLOPs	FPS	AP	AP_{75}	AR_{10}	GFLOPs	FPS
MinVIS	Swin-L	61.6	68.6	66.6	401	29	55.3	62.0	60.8	255	30
DVIS	Swin-L	63.9	70.4	69.0	411	23	58.7	66.6	64.6	405	24
DVIS-DAQ	Swin-L	65.7	73.6	70.7	415	13	61.1	68.2	66.6	410	11
DVIS++	DINOv2-L	67.7	75.3	73.7	846	18	62.3	70.2	68.0	830	17
DVIS-DAQ	DINOv2-L	68.3	76.1	73.5	851	10	62.4	70.8	68.0	836	10
CAVIS	DINOv2-L	68.9	76.2	73.6	838	15	64.6	72.5	69.3	824	15
VidEoMT	DINOv2-L	68.6	75.6	73.9	566	160	63.1	69.3	68.1	560	160

Table 2: **Online VIS on YouTube-VIS 2019 and 2021.**

Method	Backbone	YouTube-VIS 2022 *val*					OVIS *val*				
		AP^L	AP^L_{75}	AR^L_{10}	GFLOPs	FPS	AP	AP_{75}	AR_{10}	GFLOPs	FPS
MinVIS	Swin-L	33.1	33.7	36.6	224	31	39.4	41.3	43.3	408	30
DVIS	Swin-L	39.9	42.6	44.9	401	23	45.9	48.3	51.5	419	24
DVIS-DAQ	Swin-L	–	–	–	–	–	49.5	51.7	54.9	423	12
DVIS++	DINOv2-L	37.5	39.4	43.5	820	18	49.6	55.0	54.6	868	17
CAVIS	DINOv2-L	39.5	40.5	44.9	815	15	53.2	59.1	58.2	863	15
DVIS-DAQ†	DINOv2-L	42.0	43.0	48.4	826	10	54.3	60.2	59.8	1173	8
VidEoMT†	DINOv2-L	42.6	46.1	48.1	557	161	52.5	57.2	57.5	934	115

Table 3: **Online VIS on YouTube-VIS 2022 and OVIS.** † Input resolution of 544 for OVIS.

Method	Backbone	VIPSeg *val*				VSPW *val*				
		VPQ	STQ	GFLOPs	FPS	mVC_8	mVC_{16}	mIoU	GFLOPs	FPS
DVIS	Swin-L	54.7	47.7	879	20	95.0	94.3	61.3	879	22
DVIS++	DINOv2-L	56.0	49.8	2290	13	95.0	94.2	62.8	2290	13
CAVIS	DINOv2-L	56.9	51.0	2612	10	–	–	–	–	–
DVIS-DAQ	DINOv2-L	57.4	52.0	2315	4	–	–	–	–	–
VidEoMT	DINOv2-L	55.2	48.9	1897	75	95.6	95.0	64.9	1909	73

Table 4: **Online VPS on VIPSeg and VSS on VSPW.**

Video Panoptic Segmentation (VPS). Table 4 (left) compares VidEoMT with state-of-the-art methods for the VPS task on the VIPSeg benchmark. VidEoMT achieves nearly the same accuracy as DVIS++ and CAVIS, with only a minor VPS drop, while running 5–7× faster. Compared to DVIS-DAQ, which obtains the highest VPQ of 57.4, but runs at the lowest FPS of 4, VidEoMT sacrifices just 2.2 VPQ while delivering nearly 19× higher speed. These results confirm that VidEoMT also provides a significantly better accuracy and efficiency balance for video panoptic segmentation.

Video Semantic Segmentation (VSS). Table 4 (right) compares VidEoMT to state-of-the-art methods for the VSS task on the VSPW benchmark. VidEoMT outperforms existing methods, improving the mIoU by +2.1 compared to DVIS++ and also achieving a higher temporal consistency with +0.6 mVC_8 and +0.8 mVC_{16}. VidEoMT is also more than 5× faster than DVIS++. These results confirm the general applicability and strength of VidEoMT on yet another video segmentation task.

5.3 Further Analyses

EoMT as a Segmenter. In this work, we extended EoMT with lightweight temporal propagation to obtain VidEoMT. However, there are several other alternative options to enhance EoMT with tracking capabilities. In Table 5, we compare VidEoMT to alternative approaches where EoMT is used as a segmenter, and existing trackers are applied on top. Compared to the best alternative approach, EoMT + CAVIS, VidEoMT achieves slightly better AP, and is $\sim 4\times$ faster. These results demonstrate that our VidEoMT is not only the most streamlined but also considerably faster and even more accurate than alternative strategies.

Segmenter	Tracker	YouTube-VIS 2019 *val*		
		AP	GFLOPs	FPS
EoMT	CAVIS	68.1	699	42
EoMT	DVIS++	67.0	683	69
EoMT	DVIS-DAQ	67.3	703	28
VidEoMT	–	68.6	566	160

Table 5: **Alternative approaches: EoMT as a segmenter.** Comparison of EoMT equipped with modern Trackers and the proposed VidEoMT.

Query Propagation. In VidEoMT, we directly propagate object queries into the ViT encoder. To verify that this is just as accurate as propagating them in a separate decoder, we take a DINOv2 + ViT-Adapter encoder and Mask2Former decoder (Cheng et al., 2022), and apply query propagation in the decoder. We evaluate two variants for propagation: TrackFormer (Meinhardt et al., 2022), and our Query Fusion approach that combines propagated queries with the learned queries. The results in Table 6 show that our encoder-only approach achieves a similar accuracy as the encoder-decoder one, validating the effectiveness of the proposed encoder-only design. At the same time, VidEoMT is considerably faster than both alternative approaches. We refer the reader to Appendix B for additional experiments on temporal propagation.

Encoder	Decoder	YouTube-VIS 2019 *val*		
		AP	GFLOPs	FPS
DINOv2 + ViT-Ad.	TrackFormer	67.8	739	22
DINOv2 + ViT-Ad.	Query Fusion	68.0	718	32
VidEoMT	–	68.6	566	160

Table 6: **Alternative approaches: Query propagation.** Comparison of ViT-Adapter with a temporal decoder or our Query Fusion module and the proposed VidEoMT.

Impact of Pre-training. In this work, we hypothesize that large-scale pre-training with VFMs like DINOv2 enables the ViT encoder in VidEoMT to take over the functionalities of specialized components. To evaluate this, in Table 7, we evaluate the performance of VidEoMT and CAVIS in combination with the default large-scale DINOv2, medium-scale ImageNet-21K, and small-scale ImageNet-1K pre-training (Touvron et al., 2022). We find that, while VidEoMT performs comparably to CAVIS in combination with DINOv2, the performance gap between these methods increases as the pre-training scale decreases. These results support our hypothesis that large-scale pre-training is necessary to unleash the potential of VidEoMT. While Kerssies et al. (2025) showed this effect for image segmentation, our results demonstrate that large-scale pre-training also enables the ViT encoder to take over the functionalities of the specialized video segmentation components.

Model	Pre-train	YouTube-VIS 2019 *val*		
		AP	GFLOPs	FPS
CAVIS	DINOv2	68.9	838	15
VidEoMT	DINOv2	68.6	566	160
CAVIS	IN21K	62.2	838	15
VidEoMT	IN21K	60.8	566	160
CAVIS	IN1K	59.4	838	15
VidEoMT	IN1K	56.7	566	160

Table 7: **Impact of pre-training.** VidEoMT performs consistently better with better pre-training.

Impact of Model Size. Similarly, we hypothesize that increased model size positively impacts the ViT encoder ability to conduct segmentation and tracking. In Table 8, we assess this by evaluating CAVIS and VidEoMT for ViT model sizes L, B and S. The results show that the gap between the CAVIS baseline and VidEoMT decreases as model size increases, confirming our hypothesis. Additionally, while there is a small gap between CAVIS and VidEoMT for smaller model sizes, VidEoMT with a large ViT-L backbone is still an order of magnitude faster than CAVIS with a small ViT-S backbone. This further highlights the strength of VidEoMT and its superior balance between accuracy and speed.

Method	Backbone	YouTube-VIS 2019 *val*			
		AP	Params	GFLOPs	FPS
CAVIS	DINOv2-L	68.9	358M	838	15
VidEoMT	DINOv2-L	68.6	318M	566	160
CAVIS	DINOv2-B	59.5	131M	390	18
VidEoMT	DINOv2-B	58.2	95M	182	251
CAVIS	DINOv2-S	55.5	57M	251	19
VidEoMT	DINOv2-S	52.8	25M	56	294

Table 8: **Impact of model size.** VidEoMT performs consistently better as the encoder size increases.

6 Conclusion

We have introduced VidEoMT, an *encoder-only* video segmentation architecture that unifies segmentation and temporal association within a single ViT encoder. Through a step-by-step reduction of prior models, we showed that heavy task-specific modules can be removed and replaced with a lightweight query fusion mechanism, achieving an order-of-magnitude speedup while preserving or improving accuracy across multiple video segmentation benchmarks. By consolidating video segmentation into a single encoder, VidEoMT not only enables new applications through its efficiency but also provides further evidence that strongly-pretrained ViTs are powerful, general vision models that do not require specialized downstream components.

Reproducibility Statement

We provide detailed descriptions of our model design in Section 3, with the overall architecture illustrated in Figure 2. Training protocols and experimental settings are presented in Section 4, while additional implementation details are provided in Appendix A. Moreover, as stated in the abstract, the code will be made public upon acceptance. Together, these resources are intended to facilitate the reproduction of our results.

References

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, et al. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *ASPLOS*, 2024.

Niccolo Cavagnero, Gabriele Rosi, Claudia Cuttano, Francesca Pistilli, Marco Ciccone, Giuseppe Averta, and Fabio Cermelli. PEM: Prototype-based Efficient MaskFormer for Image Segmentation. In *CVPR*, 2024.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE TPAMI*, 40(4):834–848, 2018a.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*, 2018b.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision Transformer Adapter for Dense Predictions. In *ICLR*, 2023.

Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-Pixel Classification is Not All You Need for Semantic Segmentation. *NeurIPS*, 2021.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. In *CVPR*, 2022.

Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *ICLR*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021.

Miran Heo, Sukjun Hwang, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. VITA: Video Instance Segmentation via Object Token Association. *NeurIPS*, 2022.

De-An Huang, Zhiding Yu, and Anima Anandkumar. MinVIS: A Minimal Video Instance Segmentation Framework without Video-based Training. *NeurIPS*, 2022.

Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer To Rule Universal Image Segmentation. In *CVPR*, 2023.

Tommie Kerssies, Niccolò Cavagnero, Alexander Hermans, Narges Norouzi, Giuseppe Averta, Bastian Leibe, Gijs Dubbelman, and Daan de Geus. Your ViT is Secretly an Image Segmentation Model. In *CVPR*, 2025.

Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video Panoptic Segmentation. In *CVPR*, 2020.

Seunghun Lee, Jiwan Seo, Kiljoon Han, Minwoo Choi, and Sunghoon Im. Context-Aware Video Instance Segmentation. In *ICCV*, 2025.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*, 2015.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019.

Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. TrackFormer: Multi-Object Tracking with Transformers. In *CVPR*, 2022.

Meta Research. fvcore, 2023.

Jiaxu Miao, Yunchao Wei, Yu Wu, Chen Liang, Guangrui Li, and Yi Yang. VSPW: A Large-scale Dataset for Video Scene Parsing in the Wild. In *CVPR*, 2021.

Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-Scale Video Panoptic Segmentation in the Wild: A Benchmark. In *CVPR*, 2022.

David Nilsson and Cristian Sminchisescu. Semantic Video Segmentation by Gated Recurrent Flow Propagation. In *CVPR*, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *TMLR*, 2024.

Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip HS Torr, and Song Bai. Occluded Video Instance Segmentation: A Benchmark. *IJCV*, 130 (8):2022–2039, 2022.

Inkyu Shin, Dahun Kim, Qihang Yu, Jun Xie, Hong-Seok Kim, Bradley Green, In So Kweon, Kuk-Jin Yoon, and Liang-Chieh Chen. Video-kmax: A simple unified approach for online and near-online video panoptic segmentation. In *WACV*, 2024.

Richard S. Sutton. The bitter lesson. `http://www.incompleteideas.net/IncIdeas/BitterLesson.html`, 2019.

Hugo Touvron, Matthieu Cord, and Hervé Jégou. DeiT III: Revenge of the ViT. In *ECCV*, 2022.

Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers. In *CVPR*, 2021.

Mark Weber, Jun Xie, Maxwell Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, et al. STEP: Segmenting and Tracking Every Pixel. In *NeurIPS*, 2021.

Yuetian Weng, Mingfei Han, Haoyu He, Mingjie Li, Lina Yao, Xiaojun Chang, and Bohan Zhuang. Mask propagation for efficient video semantic segmentation. *NeurIPS*, 2023.

Linjie Yang, Yuchen Fan, and Ning Xu. Video Instance Segmentation. In *ICCV*, 2019.

Tao Zhang, Xingye Tian, Yu Wu, Shunping Ji, Xuebo Wang, Yuan Zhang, and Pengfei Wan. Dvis: Decoupled Video Instance Segmentation Framework. In *CVPR*, 2023.

Tao Zhang, Xingye Tian, Yikang Zhou, Shunping Ji, Xuebo Wang, Xin Tao, Yuan Zhang, Pengfei Wan, Zhongyuan Wang, and Yu Wu. DVIS++: Improved Decoupled Framework for Universal Video Segmentation. *IEEE TPAMI*, 2025.

Yikang Zhou, Tao Zhang, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Improving Video Segmentation via Dynamic Anchor Queries. *ECCV*, 2024.

APPENDIX

Table of contents

A IMPLEMENTATION DETAILS

A.1 VISUALIZATIONS OF MODEL CONFIGURATIONS

In Section 3.3 and Table 1, we gradually remove task-specific components from the state-of-the-art video segmentation model CAVIS (Lee et al., 2025), which is visualized in Figure 1 (left). To provide more details, we additionally illustrate the architecture of intermediate steps (1) to (4) in Figure B. In the first step, we replace CAVIS's original segmenter – consisting of DINOv2, ViT-Adapter, and Mask2Former's pixel decoder and Transformer decoder (Oquab et al., 2024; Chen et al., 2023; Cheng et al., 2022) – with EoMT (Kerssies et al., 2025). In the second step, we remove the context-aware features module and directly forward the segmenter's output queries to the re-identification layers. In the third step, we also remove the re-identification layers, sending the segmenter's output queries directly to the tracker's Transformer blocks. Subsequently, in the fourth step, we discard the tracker altogether, and naively apply EoMT only on a per-frame basis. In the next step, we propagate queries by directly feeding the output from frame $t-1$ into the encoder for frame t. As the final step, we introduce our query fusion design where propagated queries are fused with learnable queries. The resulting architecture is visualized in the main paper. See Figure 2 (right).

A.2 TRAINING

Following state-of-the-art models CAVIS (Lee et al., 2025), DVIS-DAQ (Zhou et al., 2024) and DVIS++ (Zhang et al., 2025), we adopt a DINOv2-pretrained ViT (Oquab et al., 2024; Dosovitskiy et al., 2021) as the backbone of VidEoMT, and we train our model in two stages. In stage one, we train the model for image segmentation only. First, we train for COCO instance segmentation, and then we further fine-tune on the video segmentation dataset without any temporal supervision. In the second stage, we introduce our temporal query propagation and fine-tune the model from stage one for video segmentation. Unlike CAVIS, DVIS-DAQ, and DVIS++, which freeze the DINOv2-initialized ViT encoder after stage one, we keep fine-tuning the ViT encoder for VidEoMT. We explore fine-tuning the ViT encoder for the CAVIS and DVIS++ baselines in Tables 1 and 2 as well, but find that the loss diverges or the memory increases beyond the GPU's limits. For our VidEoMT, note that fine-tuning the encoder is necessary because our model is encoder-only, meaning that the encoder weights need to be optimized to allow the model to be trained for video segmentation.

For step (0) in Table 1, we report results using the released CAVIS Lee et al. (2025)[1] checkpoints. For all subsequent steps, we train the models using the same settings as CAVIS with respect to input size, number of iterations, batch size, and number of sampled frames. Specifically, we use a batch size of 8, train on 8 NVIDIA H100 GPUs, and sample 5 frames from a video clip. We train for 160k iterations on YouTube-VIS (Yang et al., 2019) (all versions) and OVIS (Qi et al., 2022), for 40k iterations on VIPSeg (Miao et al., 2022), and for 20k iterations on VSPW (Miao et al., 2021).

We keep our optimization strategy similar to that of EoMT. Concretely, we use automatic mixed precision and the AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of 10^{-4}. We apply layer-wise learning rate decay (LLRD) (Devlin et al., 2019) with a factor of 0.6 and polynomial learning rate decay with a power of 0.9. A two-stage linear warm-up strategy is used for all models, including the baselines. Specifically, we first warm up the randomly initialized parameters for 500 iterations while keeping the pre-trained parameters frozen. Then, after 500 iterations, we warm up the pre-trained parameters for 1000 iterations. In both stages, the initial learning rate is set to 0.

To supervise our models, we adopt the same loss functions as Mask2Former (Cheng et al., 2022). Across all tasks and datasets, we use cross-entropy (CE) loss for the classification predictions, and

[1] https://github.com/Seung-Hun-Lee/CAVIS

Query Update	YouTube-VIS 2019 *val*		
	AP	GFLOPs	FPS
Propagation	63.9	565	162
Full reset	61.3	565	162
Non-object reset	67.8	565	157
Concatenation	67.4	580	159
TrackFormer	67.7	571	117
Fusion ⇒ VidEoMT	68.6	566	160

Table A: **Query propagation methods.** Comparison of alternative lightweight strategies for temporal propagation.

binary cross-entropy (BCE) together with Dice loss for segmentation predictions. The total loss is a weighted sum of these components:

$$\mathcal{L}_{\text{tot}} = \lambda_{\text{bce}}\mathcal{L}_{\text{bce}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}} + \lambda_{\text{ce}}\mathcal{L}_{\text{ce}}. \tag{5}$$

where λ_{bce}, λ_{dice}, and λ_{ce} are set to 5.0, 5.0, and 2.0, respectively, following Mask2Former (Cheng et al., 2022).

A.3 Architecture of Alternative Approaches

In Table 6, we compare VidEoMT with an alternative approach that uses DINOv2 + ViT-Adapter (Chen et al., 2023; Oquab et al., 2024) as the encoder, and uses a decoder that applies tracking following TrackFormer's query propagation method (Meinhardt et al., 2022). For the decoder, we adopt the architecture of Mask2Former's Transformer decoder for segmentation (Cheng et al., 2022). Following Mask2Former, this decoder has 9 layers, each composed of cross-attention, self-attention and feed-forward blocks, and it operates with a hidden dimension of 256. To conduct tracking, we follow the original TrackFormer approach as much as possible. Concretely, we first make predictions for the first frame using a set of 400 learnable queries. Using these predictions, only the N queries with a classification score $s > 0.8$ are kept and converted into *track queries*. For the next frame, these track queries are concatenated with the 400 original learnable queries, which are then fed to the decoder for that frame. In subsequent frames, the decoder updates the propagated track queries such that they predict the masks for the same objects in the new frames. Again, newly detected queries with scores $s > 0.8$ are added as additional track queries, and non-maximum suppression (NMS) with an IoU threshold of $\sigma_{\text{NMS}} = 0.9$ is applied to remove near-duplicate predictions. Note that this NMS operation is the main reason for the TrackFormer approach's inefficiency compared to VidEoMT's query propagation mechanism. Finally, at each frame, track queries are removed if their score remains below $s < 0.8$ for five consecutive frames, indicating that the object they are tracking has disappeared from the scene.

A.4 Evaluation

During evaluation, we process videos in a frame-by-frame fashion, as is required for online video segmentation. We evaluate efficiency in terms of FPS and GFLOPs. All metrics are measured on a single NVIDIA H100 GPU using PyTorch 2.7 and CUDA 12.6. We use a batch size of 1 frame to report mean values computed across all frames in the entire validation set. **FPS** is measured using FlashAttention v2 (Dao, 2024) and `torch.compile` (Ansel et al., 2024) with default settings and automatic mixed precision, after 100 warm-up iterations. **FLOPs** are measured with fvcore (Meta Research, 2023), and reported in GFLOPs (FLOPs $\times 10^{-9}$).

B Additional Experiments

Query propagation methods. VidEoMT propagates queries by fusing the learnable queries with the propagated queries. In Table A, we compare this approach with alternative methods to propagate queries. The *propagation* variant directly propagates output queries from the previous frame into the current frame's encoder, but cannot properly detect new objects as the impact of the learnable queries diminishes. For *full reset*, on the other hand, the model only receives the learnable queries like in EoMT, but is now fine-tuned for video segmentation. This variant performs the worst as there is no explicit temporal propagation. *Non-object reset* improves over this by replacing a propagated query with a learnable query if it did not predict an object in the previous frame, but this still underperforms the default fusion approach. Next, we try *concatenation* of propagated queries and

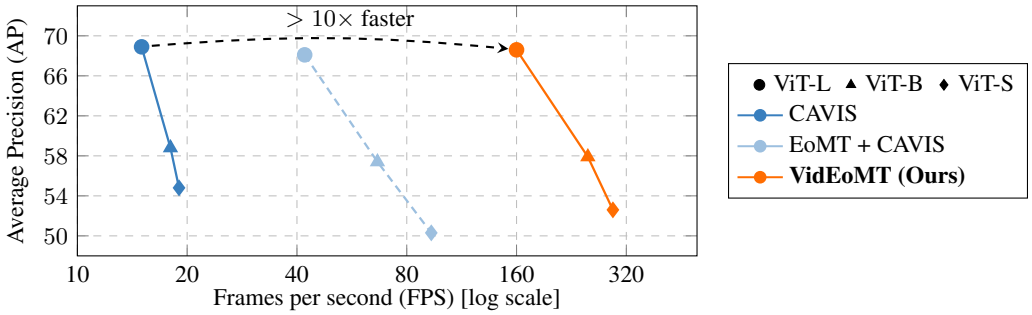


Figure A: **CAVIS (Lee et al., 2025) vs. VidEoMT (Ours).** VidEoMT provides consistently faster inference while maintaining competitive AP across different sizes of a DINOv2 pretrained ViT. Evaluated on the YouTube-VIS 2019 validation set (Yang et al., 2019).

learnable queries, but find that this introduces redundancy and harms performance. Finally, we evaluate the *TrackFormer* approach (Meinhardt et al., 2022) of only propagating queries for detected objects and introducing new learnable queries to detect new objects. This approach performs slightly worse than our *fusion* approach, but most importantly it is considerably slower because it requires filtering out duplicate detections that should not be propagated. Overall, these results demonstrate that our *fusion* approach is the most accurate and efficient.

Efficiency *vs.* Accuracy. In Figure A, we visualize the efficiency and accuracy of CAVIS and our proposed EoMT across different backbone sizes ViT-L, ViT-B, and ViT-S on the YouTube-VIS 2019 validation set. This figure illustrates the impressive speed of VidEoMT, as it is considerably faster across all model sizes, obtaining consistent speedups of over $10\times$ while only incurring small accuracy drops. Even when using the large ViT-L backbone, VidEoMT is considerably faster than CAVIS with a small ViT-S backbone, while yielding a much higher accuracy. In addition, compared to the alternative approach of extending EoMT with a CAVIS tracker, VidEoMT clearly also performs considerably better, both in terms of efficiency and accuracy. This emphasizes the strength of VidEoMT.

C LLM USAGE

We acknowledge that we used Large Language Models (LLMs) to assist us in polishing the text of this paper, on the level of one or a few sentences. Specifically, we used the GPT models available through ChatGPT by OpenAI. No LLMs were used to develop parts of the methodology or write major parts of this paper.

D QUALITATIVE RESULTS

In Figures C to E, we visualize the predictions of CAVIS (Lee et al., 2025) and VidEoMT for VIS and VPS on the YouTube-VIS 2019 (Yang et al., 2019), OVIS (Qi et al., 2022), and VIPSeg (Miao et al., 2022) datasets.

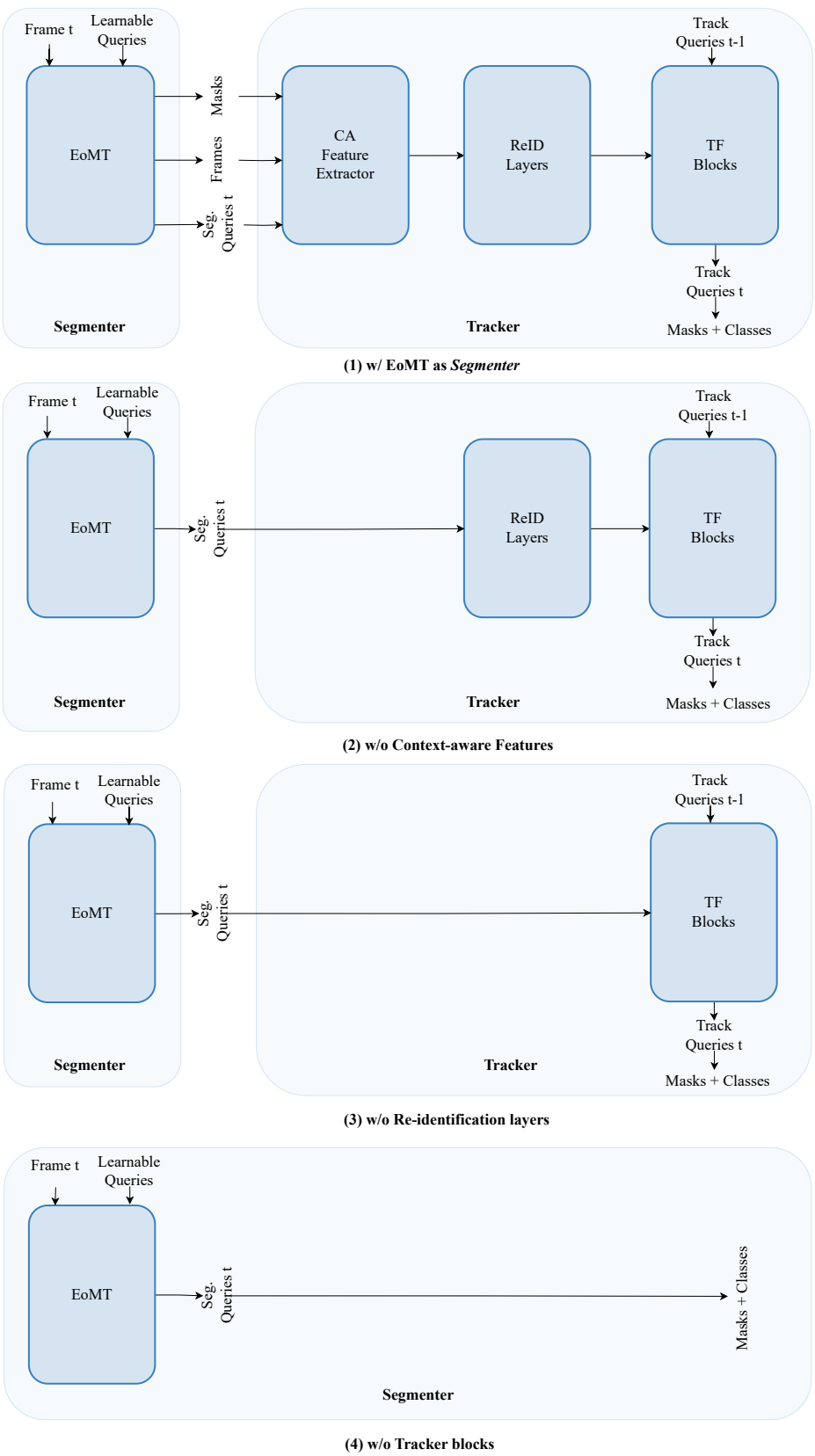
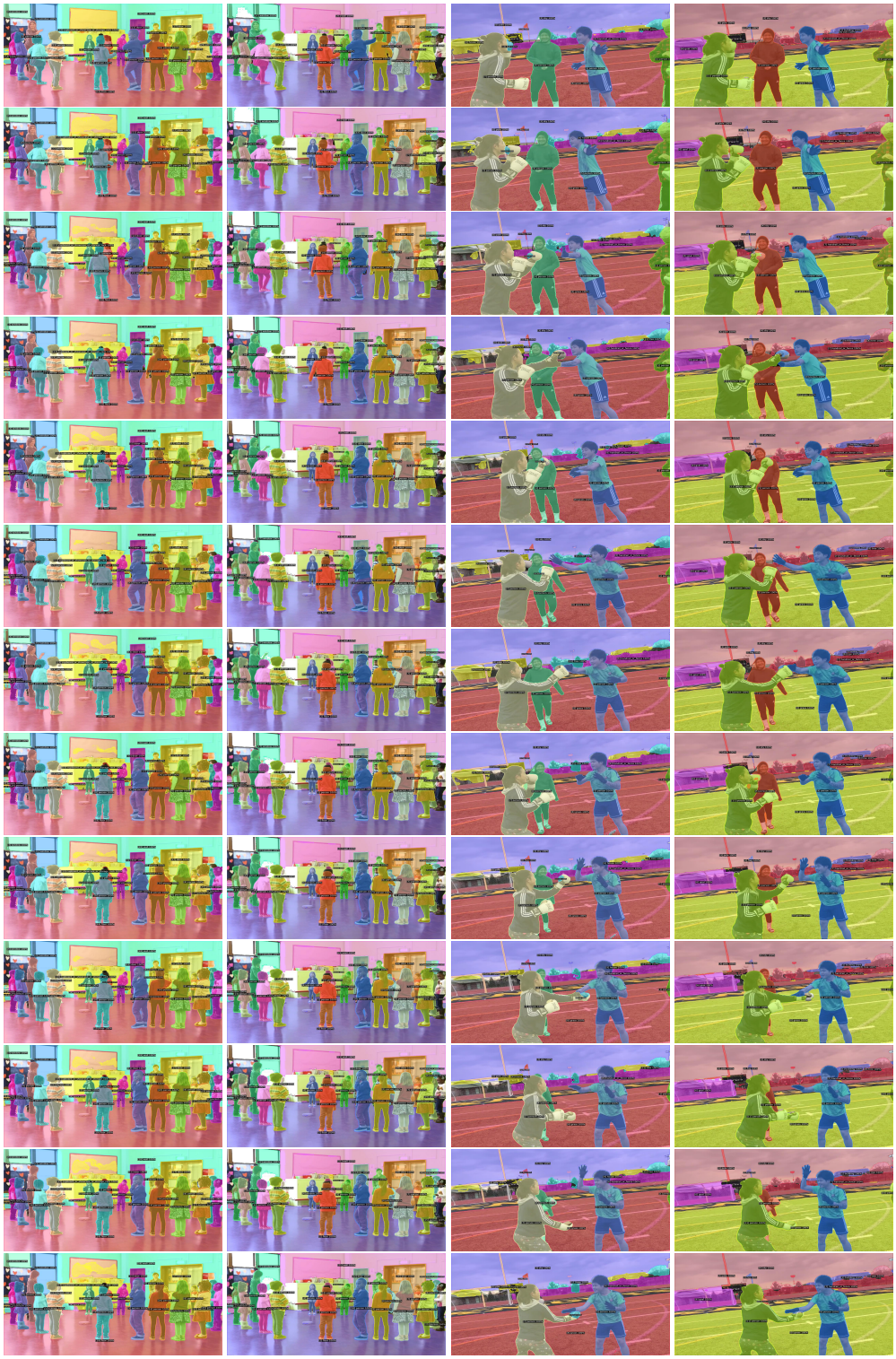


Figure B: **Removing task-specific components.**

Figure C: **Qualitative results for video instance segmentation.** We compare CAVIS (Lee et al., 2025) to VidEoMT on the YouTube-VIS 2019 dataset (Yang et al., 2019).

CAVIS (15 FPS) VidEoMT (**112 FPS**) CAVIS (15 FPS) VidEoMT (**112 FPS**)

Figure D: **Qualitative results for video instance segmentation.** We compare CAVIS (Lee et al., 2025) to VidEoMT on the OVIS dataset (Qi et al., 2022).



CAVIS (10 FPS) VidEoMT (**75 FPS**) CAVIS (10 FPS) VidEoMT (**75 FPS**)

Figure E: **Qualitative results for video panoptic segmentation.** We compare CAVIS (Lee et al., 2025) to VidEoMT on the VIPSeg dataset (Miao et al., 2022).