TarViS: A Unified Approach for Target-based Video Segmentation

Ali Athar\textsuperscript{1}  Alexander Hermans\textsuperscript{1}  Jonathon Luiten\textsuperscript{1,2}  Deva Ramanan\textsuperscript{2}  Bastian Leibe\textsuperscript{1}

\textsuperscript{1}RWTH Aachen University, Germany \textsuperscript{2}Carnegie Mellon University, USA

\{athar,hermans,luiten,leibe\}@vision.rwth-aachen.de  deva@cs.cmu.edu

Abstract

The general domain of video segmentation is currently fragmented into different tasks spanning multiple benchmarks. Despite rapid progress in the state-of-the-art, current methods are overwhelmingly task-specific and cannot conceptually generalize to other tasks. Inspired by recent approaches with multi-task capability, we propose TarViS: a novel, unified network architecture that can be applied to any task that requires segmenting a set of arbitrarily defined ‘targets’ in video. Our approach is flexible with respect to how tasks define these targets, since it models the latter as abstract ‘queries’ which are then used to predict pixel-precise target masks. A single TarViS model can be trained jointly on a collection of datasets spanning different tasks, and can hot-swap between tasks during inference without any task-specific retraining. To demonstrate its effectiveness, we apply TarViS to four different tasks, namely Video Instance Segmentation (VIS), Video Panoptic Segmentation (VPS), Video Object Segmentation (VOS) and Point Exemplar-guided Tracking (PET). Our unified, jointly trained model achieves state-of-the-art performance on 5/7 benchmarks spanning these four tasks, and competitive performance on the remaining two. Code and model weights are available at: https://github.com/Ali2500/TarViS

1. Introduction

The ability to understand video scenes has been a long-standing goal of computer vision research because of wide-ranging applications in intelligent vehicles and robots. Early approaches tackled simpler tasks involving contour-based [33, 39] and box-level tracking [21, 25, 40, 52], background subtraction [20, 61], and motion segmentation [8, 49]. The deep learning boom then revolutionized the landscape by enabling methods to perform pixel-precise segmentation on challenging, real-world videos. In the past few years, a number of benchmarks have emerged, which evaluate how well methods can perform video segmentation according to various task formulations. Over time, these tasks/benchmarks have ballooned into separate research sub-communities. Although existing methods are rapidly improving the state-of-the-art for these benchmarks, each of them typically tackles only one narrowly-defined task, and generalizing them is non-trivial since the task definition is baked into the core approach.

We argue that this fragmentation is unnecessary because video target segmentation tasks all require the same high-level capability, namely that of identifying, localizing and tracking rich semantic concepts. Meanwhile, recent progress on Transformer networks has enabled the wider AI research community to move towards unified, multi-task architectures [1, 30, 31, 38, 58], because the attention operation [62] is well-suited for processing feature sets with arbitrary structure and data modality. These developments give us the opportunity to unify the fractured landscape of target-based video segmentation. In this paper, we propose TarViS: a novel architecture which enables a single, unified model to be jointly trained for multiple video segmentation tasks. During inference, the same model can perform differ-
ent tasks at runtime by specifying the segmentation target.

The core idea is that TarViS tackles the generic task of segmenting a set of arbitrary targets in video (defined as semantic classes or as specific objects). These targets are encoded as queries which, together with the video features, are input to a Transformer-based model. The model iteratively refines these queries and produces a pixel-precise mask for each target entity. This formulation conceptually fuses all video segmentation tasks [3, 54, 66, 72] which fall under the umbrella of the above-mentioned generic task, because they differ only in how the targets are defined. During both training and inference, TarViS can hot-swap between tasks at run-time by providing the desired target query set.

To demonstrate our generalization capability, we tackle four different tasks: (1) Video Instance Segmentation (VIS) [54, 72], (2) Video Panoptic Segmentation (VPS) [35], (3) Video Object Segmentation [53], and (4) Point Exemplar-guided Tracking [3] (PET). For VIS, the segmentation targets are all objects in the video belonging to a predefined set of classes. The target set for VPS includes that for VIS, and additionally, a set of non-instantiable stuff semantic classes. For VOS, the targets are a specific set of objects for which the first-frame ground-truth mask is provided. PET is a more constrained version of VOS which only provides the location of a single point inside the object, rather than the full object mask.

Existing methods for these tasks lack generalization capability because task-specific assumptions are typically baked into the approach (see Sec. 2 and 3 for details). In contrast, TarViS can tackle all four tasks with a unified model because we encode the task-specific targets as a set of queries, thus decoupling the network architecture from the task definition. Moreover, our approach can theoretically generalize further, e.g., one could potentially define the target set as all objects described by a given text prompt, though this is beyond the scope of this paper.

To summarize, our contributions are as follows: we propose TarViS, a novel architecture that can perform any task requiring segmentation of a set of targets from video. For the first time, we are able to jointly train and infer a single model on a collection of datasets spanning the four aforementioned tasks (VIS, VPS, VOS, PET). Our experimental results show that TarViS performs competitively for VOS, and achieves state-of-the-art results for VIS, VPS and PET.

2. Related Work

Multi-task Models. Multi-task learning has a long history [11] with several architectures and training strategies [24, 36, 37, 51, 59, 76]. Earlier approaches mostly consist of a shared backbone with fixed task-specific heads, whereas we design a more general architecture for video segmentation with task-specific targets to specify what to segment. Our approach is inspired by recent attention-based models, e.g., PerceiverIO [30, 31], which can be trained on diverse data modalities and task-specific heads are replaced with output queries. UViM [38] follows a similar direction by creating a unified architecture for diverse dense prediction tasks. However, both of these models are trained separately for different tasks. Recent, powerful multi-task vision language models such as Flamingo [1] and GATO [58] tackle a multitude of tasks by requiring a sequence of task-specific input-output examples to prime the model. This is conceptually similar to our task-specific targets, however, our model does not require per-task priming. Moreover, our targets are not modeled as sequence prompts, and we aim for a video segmentation model which is several orders of magnitude smaller. In the realm of video tracking and segmentation, the recently proposed UNICORN [71] model tackles multiple object tracking-related tasks with a unified architecture. Unlike TarViS, however, UNICORN follows the task-specific output head approach and is generally oriented towards box-level tracking tasks [22, 45, 47, 75], thus requiring non-trivial modifications to tackle VPS or PET.

Query-based Transformer Architectures. Several works [2, 10, 13, 30, 31, 46, 65, 78] use query-based Transformer architectures for various tasks. The workhorse for task learning here is the iterative application of self- and cross-attention, where a set of query vectors (e.g., representing objects) are refined by interacting with each other, and with the input data (e.g., an image). Unlike existing methods which use queries in a task-specific context, TarViS adopts a query-based Transformer architecture in which the queries serve as a mechanism for decoupling the task definition from the architecture, i.e., our model can learn to tackle different tasks while being agnostic to their definition because the latter is abstracted behind a set of queries.

Task-specific Video Segmentation. Current Video Instance Segmentation (VIS) methods broadly work by predicting object tracks in the video, followed by classification into a pre-defined set of categories. Several approaches [6, 9, 23, 28, 34, 42, 54, 64, 69, 72] are based on the tracking-by-detection paradigm, some model video as a joint spatio-temporal volume [4, 5], whereas recent works [12, 26, 29, 65, 68] adopt Transformer-based architectures.

For Video Panoptic Segmentation (VPS), methods [35, 55, 66] generally extend image-level panoptic approaches [14] by employing multi-head architectures for semantic segmentation and instance mask regression, classification, and temporal association. In the Video Object Segmentation (VOS) community, state-of-the-art methods are broadly based on the seminal work of Oh et al. [50], which learns space-time correspondences between pixels in different video frames, and then uses these to propagate the first-frame masks across the video. Subsequent
methods [15–17, 60, 63, 70, 73, 73, 74] have significantly improved the performance and efficiency of this approach. Point Exemplar-guided Tracking (PET) [3, 27] is a fairly new task for which the current best approach [3] regresses a pseudo-ground-truth mask from the given point coordinates, and then applies a VOS method [17] to this mask.

The above methods thus incorporate task-specific assumptions into their core approach. This can benefit per-task performance, but makes it difficult for them to generalize across tasks. By contrast, TarViS can tackle all four aforementioned tasks, and generally any target-based video segmentation task, with a single, jointly trained model.

3. Method

TarViS can segment arbitrary targets in video since the architecture is flexible with respect to how these targets are defined, thus enabling us to conceptually unify and jointly tackle the four aforementioned tasks (VIS, VPS, VOS, PET). The architecture is illustrated in Fig. 2.

For all tasks, the common input to the network is an RGB video clip of length \( T \) denoted by \( V \in \mathbb{R}^{H \times W \times T \times 3} \). This is input to a 2D backbone network which produces image-level feature maps, followed by a Temporal Neck, which enables feature interaction across time and outputs a set of temporally consistent, multi-scale, \( D \)-dimensional feature maps \( \mathcal{F} = \{ F_{32}, F_{16}, F_{8}, F_{4} \} \) where \( F_{s} \in \mathbb{R}^{D \times H \times W \times T \times D} \). The feature maps \( \mathcal{F} \) are then fed to our Transformer decoder, together with a set of queries \( Q_{in} \) which represent the segmentation targets. The decoder applies successive layers of self- and masked cross-attention wherein the queries are iteratively refined by attending to each other, and to the feature maps, respectively. The refined queries output by the decoder are denoted with \( Q_{out} \). The following subsections explain how TarViS tackles each task in detail.

3.1. Video Instance Segmentation

VIS defines the segmentation target set as all objects belonging to a set of predefined classes. Accordingly, the input query set \( Q_{in} \) for VIS contains three types of queries: (1) semantic queries denoted by \( Q_{sem} \in \mathbb{R}^{C \times D} \) where \( C \) is the number of classes defined by the dataset, i.e., each \( D \)-dimensional vector in \( Q_{sem} \) represents a particular semantic class. (2) instance queries denoted by \( Q_{inst} \in \mathbb{R}^{I \times D} \) where \( I \) is assumed to be an upper bound on the number of instances in the video clip, and (3) a background query denoted by \( Q_{bg} \in \mathbb{R}^{1 \times D} \) to capture inactive instance queries.

The three query sets are concatenated, i.e., \( Q_{in} = \text{concat}(Q_{sem}, Q_{inst}, Q_{bg}) \), and input to the Transformer decoder, which refines their feature representation through successive attention layers and outputs a set of queries \( Q_{out} = \text{concat}(Q'_{sem}, Q'_{inst}, Q'_{bg}) \). These are then used to produce temporally consistent instance mask logits by computing the inner product \( \langle F_{s}, Q'_{inst} \rangle \in \mathbb{R}^{H \times W \times T \times 1} \). To obtain classification logits, we compute the inner product \( \langle Q'_{inst}, \text{concat}(Q'_{sem}, Q'_{bg}) \rangle \in \mathbb{R}^{I \times (C+1)} \).

The three types of queries are initialized randomly at the start of training and optimized thereafter. The instance queries \( Q_{inst} \) enable us to segment a varying number of objects from the input clip. During training, we apply Hungarian matching between the predicted and ground-truth instance masks to assign instance queries to video instances, and then supervise their predicted masks and classification logits accordingly. When training on multiple datasets with heterogeneous classes, the semantic query sets are separately initialized per dataset, but \( Q_{inst} \) and \( Q_{bg} \) are shared.

Comparison to Instance Segmentation Methods. Several Transformer-based methods [10, 13, 65, 78] for image/video instance segmentation also use queries to seg-
ment a variable number of input instances. The key difference to our approach is the handling of object classes: existing works employ only instance queries which are input to a fully-connected layer with a fan-out of \( C + 1 \) to obtain classification (and background) scores. The notion of class-guided instance segmentation is thus baked into the approach. By contrast, TarViS isagnostic to the task-specific notion of object classes because it models them as arbitrary queries which are dynamic inputs to the network. The semantic representation for these queries is thus decoupled from the core architecture and is only learned via loss supervision. An important enabler for this approach is the background query \( Q_{\text{bg}} \), which serves as a ‘catch-all’ class to represent everything that is not in \( Q_{\text{sem}} \). It is used to classify non-active instance queries, and its mask logits are supervised to segment all non-object input pixels.

### 3.2. Video Panoptic Segmentation

VPS defines the segmentation targets as all objects belonging to a set of \( \text{thing} \) classes (e.g., ‘person’, ‘car’), and additionally, a set of non-instantiable \( \text{stuff} \) classes (e.g., ‘sky’, ‘grass’) which cover all non-object pixels. TarViS can tackle VPS with virtually no modification to the workflow in Sec. 3.1. We can compute semantic segmentation masks for the input clip by simply taking the inner product between \( Q_{\text{sem}} \) and the video features: \( \langle F_4, Q'_{\text{sem}} \rangle \in \mathbb{R}^{H \times W \times T \times C} \). Note that here, \( Q_{\text{sem}} \) contains queries representing both \( \text{thing} \) and \( \text{stuff} \) classes.

**Comparison to VPS Methods.** Current VPS datasets \( [35, 66] \) involve driving scene videos captured from moving vehicles. Methods tackling this task \( [35, 55] \) are based on earlier image panoptic segmentation approaches \( [14] \) which involve multi-head networks for semantic and instance segmentation prediction. In terms of image-level panoptic segmentation, Mask2Former \( [13] \) uses a Transformer-based architecture, but it models \( \text{stuff} \) classes as instances which are Hungarian-matched to the ground-truth target during training, whereas TarViS models semantic classes and instances using separate, designated queries.

### 3.3. Video Object Segmentation and Point Exemplar-guided Tracking

VOS and PET are instantiations of a general task where the segmentation targets are a set of \( O \) objects for which some ground-truth cue \( G \) is given. For VOS, \( G \) is provided as the first-frame object masks \( M_{\text{obj}} \in \mathbb{R}^{O \times H \times W} \), whereas for PET, \( G \) is provided as the \((x, y)\) coordinates \( P_{\text{obj}} \in \mathbb{R}^{O \times 2} \) of a point inside each of the objects. TarViS jointly tackles both tasks by adopting a generalized approach in which the \( O \) target objects are encoded into a set of object queries \( Q_{\text{obj}} \). Thus, both VOS and PET boil down to designing a function \( \text{EncodeObjects}(\cdot) \) which regresses \( Q_{\text{obj}} \) from the ground-truth cues \( G \) and feature maps \( F \):

\[
Q_{\text{obj}} \leftarrow \text{EncodeObjects}(G, F). \tag{1}
\]

Note that \( Q_{\text{obj}} \) is conceptually analogous to \( Q_{\text{sem}} \) and \( Q_{\text{inst}} \) used for VIS in that all three are abstract representations for their respective task-specific segmentation targets.

**Video Object Segmentation.** We seek inspiration from HODOR \( [2] \) to implement \( \text{EncodeObjects} \) for VOS, a recent method for weakly-supervised VOS, which encodes objects into concise descriptors as follows: the descriptors are initialized by average pooling the image features inside the object masks, followed by iterative refinement where the descriptors attend to each other (self-attention) and to their respective soft-masked image features (cross-attention).

For TarViS, we employ a lightweight \textit{Object Encoder} with a similar workflow to encode the objects as a set of queries \( Q_{\text{obj}} \), but with two differences to HODOR \( [2] \): instead of cross-attending to the entire image feature map \((H \cdot W\) points\) with soft-masked attention, we apply hard-masked cross-attention to at most \( p_{\text{max}} \) feature points per object, where \( p_{\text{max}} \ll H \cdot W \). Object masks containing more than \( p_{\text{max}} \) points are sub-sampled accordingly. This significantly improves the memory/run-time overhead of our Object Encoder. Secondly, we note that the process of distilling object features into a single descriptor involves a loss of object appearance information, which degrades performance. We therefore model each object with \( q_o \) queries (instead of one) by spatially dividing each object mask into \( q_o \) segments, \( i.e. \), \( Q_{\text{obj}} \in \mathbb{R}^{O \times q_o \times D} \) (we use \( q_o = 4 \)).

In addition to \( Q_{\text{obj}} \), we initialize a set of background queries \( Q_{\text{bg}} \) to model the non-target pixels in the reference frame. Following HODOR \( [2] \), we employ multiple background queries, which are initialized dynamically by dividing the video frame containing the ground-truth masks \( M_{\text{obj}} \) into a \( 4 \times 4 \) grid and average pooling the non-object pixels in each grid cell. The Object Encoder jointly refines the background and object queries to yield \( Q_{\text{in}} = \text{concat}(Q_{\text{obj}}, Q_{\text{bg}}) \). During training, the mask logits for the multiple background queries are aggregated per-pixel by applying \( \text{max}(\cdot) \) and supervised to segment all pixels not part of the target object set.

The remaining workflow follows that for VIS and VPS: \( Q_{\text{in}} \) is input to the Transformer decoder together with the video features \( F \). The refined output query set \( Q_{\text{out}} = \text{concat}(Q'_{\text{obj}}, Q'_{\text{bg}}) \) is then used to compute the inner product \( \langle F_4, Q'_{\text{obj}} \rangle \in \mathbb{R}^{H \times W \times T \times O} \). Subsequently, \( \text{max}(\cdot) \) is applied on the \( q_o \)-sized dimension to obtain the final mask logits for the \( O \) target objects.

**Point Exemplar-guided Tracking.** For PET we implement \( \text{EncodeObjects} \) in the exactly same way as VOS: the given point coordinates \( P_{\text{obj}} \) are converted into a mask with just one non-zero pixel, followed by iterative refinement by the Object Encoder (with shared weights for VOS and PET).
Deformable Attention apply gets. Some image instance segmentation methods [13, 78] propose a novel Temporal Neck consistent across the entire input video clip. To achieve this, we TarViS, however, the features must also be temporally con-

Fig. 3. Temporal Neck Layer. Colored regions denote the attention field w.r.t the selected pixel (darkened). Deformable Attention is spatially unrestricted but temporally limited to a single frame, whereas Temporal Attention is spatially localized, but temporally unrestricted. $F_8$ is inactive for the temporal attention.

interaction across multiple scales. Both attention operations are illustrated in Fig. 3. We exclude $F_8$ from temporal attention since we found this to be more memory-efficient without negatively impacting prediction quality.

**Transformer Decoder.** The decoder architecture follows that of Mask2Former [13]: the input queries are iteratively refined over multiple layers. In each layer, the queries first cross-attend to their respective masked video features, then self-attend to each other, followed by feed-forward layers.

### 3.5. Inference

To infer on videos with arbitrary length, we split videos into clips of length $T_{\text{clip}}$ with an overlap of $T_{ov}$ between successive clips. Object tracks are associated across clips based on their mask IoU in the overlapping frames. For the VOS tasks, the object queries for an intermediate clip are initialized by using the predicted masks in the overlapping frames from the previous clip as a pseudo-ground-truth. For VPS, we average the semantic segmentation logits in the overlapping frames. Our approach is thus near-online because the time delay in obtaining the output for a given frame is at most $T_{\text{clip}} - T_{ov} - 1$ (except for the first clip in the video).

### 4. Experiments

**4.1. Implementation Details**

Our Temporal Neck contains 6 layers of Deformable and Temporal Attention. We pretrain for 500k iterations on pseudo-video clips generated by applying on-the-fly augmentations to images from COCO [43], ADE20k [77], Mapillary [48] and Cityscapes [18]. The samples are either trained for VPS, VIS, VOS or PET. This is followed by fine-tuning for 90k iterations jointly on samples from YouTube-VIS [72], OVIS [54], KITTI-STEP [66],
Table 1. Results for Video Instance Segmentation (VIS) on the YouTube-VIS 2021 [72] and OVIS [54] validation sets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Shared</th>
<th>YouTube-VIS 2021</th>
<th>OVIS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Model</td>
<td>AP</td>
<td>AP50</td>
</tr>
<tr>
<td>Mask2Former-VIS</td>
<td>R-50</td>
<td>✗</td>
<td>40.6</td>
<td>60.9</td>
</tr>
<tr>
<td>IDOL [69]</td>
<td>R-50</td>
<td>✗</td>
<td>43.9</td>
<td>68.0</td>
</tr>
<tr>
<td>MinVIS [28]</td>
<td>R-50</td>
<td>✗</td>
<td>44.2</td>
<td>66.0</td>
</tr>
<tr>
<td>VITA [26]</td>
<td>R-50</td>
<td>✗</td>
<td>45.7</td>
<td>67.4</td>
</tr>
<tr>
<td>TarViS</td>
<td>R-50</td>
<td>✓</td>
<td>48.3</td>
<td>69.6</td>
</tr>
<tr>
<td>Mask2Former-VIS</td>
<td>Swin-T</td>
<td>✗</td>
<td>45.9</td>
<td>68.7</td>
</tr>
<tr>
<td>IDOL [69]</td>
<td>Swin-L</td>
<td>✓</td>
<td>56.1</td>
<td>80.8</td>
</tr>
<tr>
<td>VITA [26]</td>
<td>Swin-L</td>
<td>✓</td>
<td>57.5</td>
<td>80.6</td>
</tr>
<tr>
<td>TarViS</td>
<td>Swin-L</td>
<td>✓</td>
<td>60.2</td>
<td>81.4</td>
</tr>
</tbody>
</table>

Table 2. Video Panoptic Segmentation (VPS) results for validation sets of KITTI-STEP [66], CityscapesVPS [35] and VIPSeg [44].

<table>
<thead>
<tr>
<th>Method</th>
<th>Shared</th>
<th>KITTI-STEP</th>
<th>CityscapesVPS</th>
<th>VIPSeg</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Model</td>
<td>STQ</td>
<td>AQ</td>
<td>SQ</td>
</tr>
<tr>
<td>Mask Propagation</td>
<td>✗</td>
<td>0.67</td>
<td>0.63</td>
<td>0.71</td>
</tr>
<tr>
<td>Track [35]</td>
<td>-</td>
<td>55.9</td>
<td>43.7</td>
<td>64.8</td>
</tr>
<tr>
<td>VPSNet [35]</td>
<td>✗</td>
<td>57.0</td>
<td>44.7</td>
<td>66.0</td>
</tr>
<tr>
<td>VPSNet-SiamTrack</td>
<td>✗</td>
<td>57.3</td>
<td>44.7</td>
<td>66.4</td>
</tr>
<tr>
<td>VIP-Deeplab [35]</td>
<td>✗</td>
<td>63.1</td>
<td>49.5</td>
<td>73.0</td>
</tr>
<tr>
<td>Clip-PanoFCN [44]</td>
<td>✗</td>
<td>22.9</td>
<td>25.0</td>
<td>20.8</td>
</tr>
<tr>
<td>TarViS (R-50)</td>
<td>✓</td>
<td>53.3</td>
<td>35.9</td>
<td>66.0</td>
</tr>
<tr>
<td>TarViS (Swin-T)</td>
<td>✓</td>
<td>58.0</td>
<td>42.9</td>
<td>69.0</td>
</tr>
<tr>
<td>TarViS (Swin-L)</td>
<td>✓</td>
<td>58.9</td>
<td>43.7</td>
<td>69.9</td>
</tr>
</tbody>
</table>

Video Panoptic Segmentation (VPS). We evaluate VPS on three datasets: (1) KITTI-STEP [66], which contains 12/9 lengthy driving scene videos for training/validation with 19 semantic classes (2 thing and 17 stuff classes), (2) CityscapesVPS [35], which contains 50 short driving scene clips, each with 6 annotated frames, and (3) VIPSeg [44], which is a larger dataset with 2806/343 in-the-wild videos for training/validation and 124 semantic classes. The results are reported in Tab. 2. For KITTI-STEP, TarViS achieves 70% STQ with a ResNet-50 backbone which is better than all existing approaches. The performance further improves to 72% with Swin-L. For CityscapesVPS, TarViS achieves 58.9 VPQ which is higher than all other methods except VIP-Deeplab [55] (63.1). However, VIP-Deeplab performs monocular depth estimation for additional guidance, and therefore requires ground-truth depth-maps for training.

For VIPSeg, TarViS outperforms existing approaches by a significant margin. With a ResNet-50 backbone, our 33.5 VPQ is 10.6% higher than the 22.9 by Clip-PanoFCN [44]. With a Swin-Large backbone, TarViS achieves 48.0 VPQ.
based methods as well as all non-STC based methods, e.g., using instance queries $Q$ in conjunction with a linear layer for classification (separate for each object guidance (point or mask) is abstracted behind the $\text{EncodeObjects}(\cdot)$ function.

TarViS achieves a HOTA$_{val}$ score of 37.5 and 36.4 on the validation and test sets, respectively, which is significantly better than the 24.4 and 24.9 achieved by the best performing baseline method which casts PET as a VOS problem by regressing a pseudo-ground-truth mask from the given point, followed by applying a VOS approach (STCN [17]).

### 4.3. Ablations

Table 4 shows several architecture/training ablations.

#### Task-specific Training (row 1-3).

The first three rows show results for task-specific models. We train a single model for VOS and PET since both tasks are closely related. We note that the VIS-only model performs worse than the multi-task model on YouTube-VIS (46.3 vs. 48.3) but slightly better on OVIS (31.5 vs. 31.1). For VPS, the performance on KITTI-STEP is unchanged, but Cityscapes-VPS and VIPSeg both show improvements with the multi-task model. Lastly, for VOS the task-specific model performs slightly better on BURST for PET (34.7 vs. 30.9). To summarize, the final, multi-task model performs better on 4/7 benchmarks, worse on 2/7, and matches performance on 1/7 when compared to task-specific models. We thus conclude that the combination of multi-task supervision and more data is generally beneficial for performance.

#### Semantic Queries for VIS (row 4).

TarViS represents object classes as dynamic query inputs to the network ($Q_{\text{sem}}$, Sec. 3.1). We ablate this by modifying our network to work for only VIS by discarding the semantic/background queries and adopting a technique similar to existing methods [13, 65], i.e., using instance queries $Q_{\text{inst}}$ in conjunction with a linear layer for classification (separate for each dataset). Comparing the results with the VIS-only setting which is trained on similar data, we see that this architecture

<table>
<thead>
<tr>
<th>Method</th>
<th>DAVIS (VOS)</th>
<th>BURST (PET)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mathcal{J}$ &amp; $\mathcal{F}$</td>
<td>$\mathcal{J}$</td>
</tr>
<tr>
<td>UNICORN+ [71]</td>
<td>70.6</td>
<td>66.1</td>
</tr>
<tr>
<td>HODOR [2]</td>
<td>81.3</td>
<td>78.4</td>
</tr>
<tr>
<td>STM [50]</td>
<td>81.8</td>
<td>79.2</td>
</tr>
<tr>
<td>CFBI [73]</td>
<td>81.9</td>
<td>79.1</td>
</tr>
<tr>
<td>HMMN [60]</td>
<td>84.7</td>
<td>81.9</td>
</tr>
<tr>
<td>AOT [74]</td>
<td>84.9</td>
<td>82.3</td>
</tr>
<tr>
<td>STCN [17]</td>
<td>85.4</td>
<td>82.2</td>
</tr>
<tr>
<td>XMem [15]</td>
<td>86.2</td>
<td>82.9</td>
</tr>
<tr>
<td>Box Tracker [32]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>STCN+M2F [13, 17]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TarViS (R-50)</td>
<td>82.6</td>
<td>79.3</td>
</tr>
<tr>
<td>TarViS (Swin-T)</td>
<td>82.8</td>
<td>79.6</td>
</tr>
<tr>
<td>TarViS (Swin-L)</td>
<td>85.3</td>
<td>81.7</td>
</tr>
</tbody>
</table>

#### Table 3. Results for VOS on DAVIS [53] and PET on BURST [3]. Detailed PET metrics are provided in supplementary.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Video Training Data</th>
<th>VIS</th>
<th>VPS</th>
<th>VOS</th>
<th>PET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>YTVIS</td>
<td>OVIS</td>
<td>KITTI</td>
<td>C-VPS</td>
<td>VIPSeg</td>
</tr>
<tr>
<td>1. VIS</td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YTVIS</td>
<td>OVIS</td>
<td>KITTI</td>
<td>C-VPS</td>
<td>VIPSeg</td>
</tr>
<tr>
<td>2. VPS</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YTVIS</td>
<td>OVIS</td>
<td>KITTI</td>
<td>C-VPS</td>
<td>VIPSeg</td>
</tr>
<tr>
<td>3. VOS + PET</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YTVIS</td>
<td>OVIS</td>
<td>KITTI</td>
<td>C-VPS</td>
<td>VIPSeg</td>
</tr>
<tr>
<td>4. No Semantic Queries</td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>YTVIS</td>
<td>OVIS</td>
<td>KITTI</td>
<td>C-VPS</td>
<td>VIPSeg</td>
</tr>
<tr>
<td>5. No Temporal Neck</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
</tr>
<tr>
<td></td>
<td>YTVIS</td>
<td>OVIS</td>
<td>KITTI</td>
<td>C-VPS</td>
<td>VIPSeg</td>
</tr>
<tr>
<td>Final</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
</tr>
</tbody>
</table>

performs worse than the VIS-only setting on both YouTube-VIS (44.7 vs. 46.3) and OVIS (29.8 vs. 31.5). Thus, our semantic query based classification makes the network architecture task-agnostic and also yields better performance.

Temporal Neck (row 4). We validate our novel Temporal Neck (Sec. 3.4) by training a model with a simpler neck that contains only Deformable Attention layers [78], similar to Mask2Former [13], i.e. there is no feature interaction across frames. Doing this degrades performance across all datasets, with particularly large drops for YouTube-VIS (42.8 vs. 48.3) and OVIS (22.3 vs. 31.1). This shows that inter-frame feature interactions enabled by our Temporal Neck are highly beneficial for downstream tasks.

5. Discussion

Limitations. Training on multiple datasets/tasks does not necessarily improve performance on all benchmarks. For VOS, the model exhibits class bias and sometimes fails to track unusual objects which were not seen during training.

Future Outlook. We jointly trained TarViS for four different tasks to validate its generalization capability. The architecture can, however, tackle any video segmentation task for which the targets can be encoded as queries. The recent emergence of joint language-vision models [41,56,57] thus makes it possible to perform multi-object segmentation based on a text prompt if the latter can be encoded as a target query using a language encoder [19]. Another interesting possibility is that TarViS could be applied to multiple tasks in the same forward pass by simply concatenating the task-specific queries. Fig. 5 offers a promising outlook for this: it shows our model’s output for a video clip from a popular TV series where we perform VIS and VOS simultaneously by providing the semantic query for the ‘person’ class (from YouTube-VIS [72]), and the VOS-based object queries for the dragon by annotating its first frame mask, i.e., \( Q_{in} = \text{concat}(Q_{\text{sem}}, Q_{\text{inst}}, Q_{\text{obj}}, Q_{\text{bg}}) \). TarViS successfully segments all four persons in the scene (VIS) and the dragon (VOS), even though our model was never trained to simultaneously tackle both tasks in a single forward pass.

Figure 5. TarViS performing VIS and VOS in a single forward pass. We provide the mask for the dragon on the left, and the semantic query for the ‘person’ class.

6. Conclusion

We presented TarViS: a novel, unified approach for tackling any task requiring pixel-precise segmentation of a set of targets in video. We adopt a generalized paradigm where the task-specific targets are encoded into a set of queries which are then input to our network together with the video features. The network is trained to produce segmentation masks for each target entity, but is inherently agnostic to the task-specific definition of these targets. To demonstrate the effectiveness of our approach, we applied it to four different video segmentation tasks (VIS, VPS, VOS, PET). We showed that a single TarViS model can be jointly trained for all tasks, and during inference can hot-swap between tasks without any task-specific fine-tuning. Our model achieved state-of-the-art performance on five benchmarks and has multiple, promising directions for future work.

Acknowledgments. This project was partially funded by ERC CoG DeeVise (ERC-2017-COG-773161) and BMBF project NeuroSys-D (03ZU1106DA). Compute resources were granted by RWTH Aachen under project ID supp0003, and by the Gauss Centre for Supercomputing e.V. through the John von Neumann Institute for Computing on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre.
References

[18] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In CVPR, 2016. 5
[27] Namdar Homayounfar, Justin Liang, Wei-Chiu Ma, and Raquel Urtasun. Videoclick: Video object segmentation with a single click. Arxiv, 2021. 3
[34] Lei Ke, Xia Li, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Prototypical cross-attention networks for multiple object tracking and segmentation. In NeurIPS, 2021. 2


[49] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. PAMI, 2013. 1


[65] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Kolesnikov, Andr ´e Susano Pinto, Lucas Beyer, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko-

[67] Sanghyun Woo, Dahun Kim, Joon-Young Lee, and In So Kweon. Learning to associate every segment for video panoptic segmentation. In CVPR, 2021. 6


[71] Bin Yan, Yi Jiang, Peize Sun, Dong Wang, Zehuan Yuan, Ping Luo, and Huchuan Lu. Towards grand unification of object tracking. In ECCV, 2022. 2, 7

[72] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In ICCV, 2019. 2, 5, 6, 8


