Universal Instance Perception as Object Discovery and Retrieval

Bin Yan\(^1\)*, Yi Jiang\(^2\)†, Jiannan Wu\(^3\), Dong Wang\(^1\), Ping Luo\(^3\), Zehuan Yuan\(^2\), Huchuan Lu\(^{1,4,†}\)

\(^1\) School of Information and Communication Engineering, Dalian University of Technology, China
\(^2\) ByteDance
\(^3\) The University of Hong Kong
\(^4\) Peng Cheng Laboratory

Abstract

All instance perception tasks aim at finding certain objects specified by some queries such as category names, language expressions, and target annotations, but this complete field has been split into multiple independent sub-tasks. In this work, we present a universal instance perception model of the next generation, termed UNINEXT. UNINEXT reformulates diverse instance perception tasks into a unified object discovery and retrieval paradigm and can flexibly perceive different types of objects by simply changing the input prompts. This unified formulation brings the following benefits: (1) enormous data from different tasks and label vocabularies can be exploited for jointly training general instance-level representations, which is especially beneficial for tasks lacking in training data. (2) the unified model is parameter-efficient and can save redundant computation when handling multiple tasks simultaneously. UNINEXT shows superior performance on 20 challenging benchmarks from 10 instance-level tasks including classical image-level tasks (object detection and instance segmentation), vision-and-language tasks (referring expression comprehension and segmentation), and six video-level object tracking tasks. Code is available at https://github.com/MasterBin-IIAU/UNINEXT.

1. Introduction

Object-centric understanding is one of the most essential and challenging problems in computer vision. Over the years, the diversity of this field increases substantially. In this work, we mainly discuss 10 sub-tasks, distributed on the vertices of the cube shown in Figure 1. As the most fundamental tasks, object detection \([8, 9, 30, 59, 82, 84, 91]\) and instance segmentation \([6, 37, 64, 90, 97]\) require finding all objects of specific categories by boxes and masks respectively. Extending inputs from static images to dynamic videos, Multiple Object Tracking (MOT) \([3, 74, 126, 128]\), Multi-Object Tracking and Segmentation (MOTS) \([47, 93, 108]\), and Video Instance Segmentation (VIS) \([43, 100, 103, 112]\) require finding all object trajectories of specific categories in videos. Except for category names, some tasks provide other reference information. For example, Referring Expression Comprehension (REC) \([115, 121, 130]\), Referring Expression Segmentation (RES) \([116, 119, 121]\), and Referring Video Object Segmentation (R-VOS) \([7, 86, 104]\) aim at finding objects matched with the given language expressions like “The fourth person from the left”. Besides, Single Object Tracking (SOT) \([5, 51, 106]\) and Video Object Segmentation (VOS) \([18, 78, 107]\) take the target annotations (boxes or masks) given in the first frame as the reference, requiring to predict the trajectories of the tracked objects in the subsequent frames. Since all the above tasks aim to perceive instances of certain properties, we refer to them collectively as instance perception.

Although bringing convenience to specific applications, such diverse task definitions split the whole field into fragmented pieces. As the result, most current instance perception methods are developed for only a single or a part of sub-tasks and trained on data from specific domains. Such
We propose a unified prompt-guided formulation for universal instance perception, reuniting previously fragmented instance-level sub-tasks into a whole.

Benefiting from the flexible object discovery and retrieval paradigm, UNINEXT can train on different tasks and domains, in no need of task-specific heads.

UNINEXT achieves superior performance on 20 challenging benchmarks from 10 instance perception tasks using a single model with the same model parameters.

2. Related Work

Instance Perception. The goals and typical methods of 10 instance perception tasks are introduced as follows.

Retrieval by Category Names. Object detection and instance segmentation aim at finding all objects of specific classes on the images in the format of boxes or masks. Early object detectors can be mainly divided into two-stage methods \(8, 12, 84\) and one-stage methods \(33, 60, 82, 91, 124\) according to whether to use RoI-level operations \(35, 37\). Recently, Transformer-based detectors \(9, 53, 131\) have drawn great attention for their conceptually simple and flexible frameworks. Besides, instance segmentation approaches can also be divided into detector-based \(8, 12, 37, 49, 90\) and detector-free \(16, 97\) fashions according to whether box-level detectors are needed. Object detection and instance segmentation play critical roles and are foundations for all other instance perception tasks. For example, MOT, MOTS, and VIS extend image-level detection and segmentation to videos, requiring finding all object trajectories of specific classes in videos. Mainstream algorithms \(47, 79, 101, 102, 109, 125\) of MOT and MOTS follow an online “detection-then-association” paradigm. However, due to the intrinsic difference in benchmarks of MOTS \(93, 120\) (high-resolution long videos) and VIS \(112\) (low-resolution short videos), most recent VIS methods \(43, 58, 100, 103\) adopt an offline fashion. This strategy performs well on relatively simple VIS2019 \(112\), but the performance drops drastically on challenging OVIS \(81\) benchmark. Recently, IDOL \(105\) bridges the performance gap between online fashion and its offline counterparts by discriminative instance embeddings, showing the potential of the online paradigm in unifying MOT, MOTS, and VIS.

Retrieval by Language Expressions. REC, RES, and R-VOS aim at finding one specific target referred by a language expression using boxes or masks on the given images or videos. Similar to object detection, REC methods can be categorized into three paradigms: two-stage \(40, 62, 65, 113\), one-stage \(57, 70, 114, 115\), and Transformer-based \(23, 45, 129\) ones. Different from REC, RES approaches \(11, 25, 31, 41, 44, 69, 119\) focus more on designing diverse attention mechanisms to achieve vision-language alignment. Recently, SeqTR \(130\) unifies REC and RES.
as a point prediction problem and obtains promising results. Finally, R-VOS can be seen as a natural extension of RES from images to videos. Current state-of-the-art methods [7, 104] are Transformer-based and process the whole video in an offline fashion. However, the offline paradigm hinders the applications in the real world such as long videos and ongoing videos (e.g. autonomous driving).

Retrieval by Reference Annotations. SOT and VOS first specify tracked objects on the first frame of a video using boxes or masks, then require algorithms to predict the trajectories of the tracked objects in boxes or masks respectively. The core problems of these two tasks include (1) How to extract informative target features? (2) How to fuse the target information with representations of the current frame? For the first question, most SOT methods [5, 14, 50, 51, 110] encode target information by passing a template to a siamese backbone. While VOS approaches [18, 78, 117] usually pass multiple previous frames together with corresponding mask results to a memory encoder for extracting fine-grained target information. For the second question, correlations are widely adopted by early SOT algorithms [5, 51, 111]. However, these simple linear operations may cause serious information loss. To alleviate this problem, later works [14, 19, 110, 118] resort to Transformer for more discriminative representations. Besides, feature fusion in VOS is almost dominated by space-time memory networks [17, 18, 78, 117].

Unified Vision Models. Recently, unified vision models [13, 16, 34, 37, 54, 68, 83, 87, 95, 109, 132] have drawn great attention and achieved significant progress due to their strong generalizability and flexibility. Unified vision models attempt to solve multiple vision or multi-modal tasks by a single model. Existing works can be categorized into unified learning paradigms and unified model architectures.

Unified Learning Paradigms. These works [2, 34, 68, 83, 87, 95, 132] usually present a universal learning paradigm for covering as many tasks and modalities as possible. For example, MuST [34] presents a multi-task self-training approach for 6 vision tasks. INTERN [87] introduces a continuous learning scheme, showing strong generalization ability on 26 popular benchmarks. Unified-IO [68] and OFA [95] proposes a unified sequence-to-sequence framework that can handle a variety of vision, language, and multi-modal tasks. Although these works can perform many tasks, the commonality and inner relationship among different tasks are less explored and exploited.

Unified Model Architectures. These works [13, 16, 37, 54, 109] usually designs a unified formulation or model architecture for a group of closely related tasks. For example, Mask R-CNN [37] proposes a unified network to perform object detection and instance segmentation simultaneously. Mask2Former [16] presents a universal architecture capable of handling panoptic, instance, and semantic segmentation. Pix2SeqV2 [13] designs a unified pixel-to-sequence interface for four vision tasks, namely object detection, instance segmentation, keypoint detection, and image captioning. GLIP [54] cleverly reformulates object detection as phrase grounding by replacing classical classification with word-region alignment. This new formulation allows joint training on both detection and grounding data, showing strong transferability to various object-level recognition tasks. However, GLIP [54] supports neither prompts in other modalities such as images & annotations nor video-level tracking tasks. In terms of object tracking, Unicorn [109] proposes a unified solution for SOT, VOS, MOT, and MOTS, achieving superior performance on 8 benchmarks with the same model weights. However, it is still difficult for Unicorn to handle diverse label vocabularies [22, 61, 74, 81, 112, 120] during training and inference. In this work, we propose a universal prompt-guided architecture for 10 instance perception tasks, conquering the drawbacks of GLIP [54] and Unicorn [109] simultaneously.

3. Approach

Before introducing detailed methods, we first categorize existing instance perception tasks into three classes.

- Object detection, instance segmentation, MOT, MOTS, and VIS take category names as prompts to find all instances of specific classes.
- REC, RES, and R-VOS exploit an expression as the prompt to localize a certain target.
- SOT and VOS use the annotation given in the first frame as the prompt for predicting the trajectories of the tracked target.

Essentially, all the above tasks aim to find objects specified by some prompts. This commonality motivates us to reformulate all instance perception tasks into a prompt-guided object discovery and retrieval problem and solve it by a unified model architecture and learning paradigm. As demonstrated in Figure 2, UNINEXT consists of three main components: (1) prompt generation (2) image-prompt feature fusion (3) object discovery and retrieval.

3.1. Prompt Generation

First, a prompt generation module is adopted to transform the original diverse prompt inputs into a unified form. According to different modalities, we introduce the corresponding strategies in the next two paragraphs respectively.

To deal with language-related prompts, a language encoder [24] is adopted. To be specific, for category-guided tasks, we concatenate class names that appeared in the current dataset [61, 81, 112, 120] as the language expression. Take COCO [61] as an example, the expression can be written as “person. bicycle. ... . toothbrush”. Then
for both category-guided and expression-guided tasks, the language expression is passed into $\text{Enc}_L$, getting a prompt embedding $F_p \in \mathbb{R}^{L \times d}$ with a sequence length of $L$.

For the annotation-guided tasks, to extract fine-grained visual features and fully exploit the target annotations, an additional reference visual encoder $\text{Enc}_V^{\text{ref}}$ is introduced. Specifically, first a template with $2^2$ times target box area is cropped centered on the target location on the reference frame. Then the template is resized to a fixed size of $256 \times 256$. To introduce more precise target information, an extra channel named the target prior is concatenated to the template image, forming a 4-channel input. In more detail, the value of the target prior is 1 on the target region otherwise 0. Then the template image together with the target prior is passed through another visual encoder $\text{Enc}_V$, obtaining a hierarchical feature pyramid $\{C_3, C_4, C_5, C_6\}$. The corresponding spatial sizes are $32 \times 32, 16 \times 16, 8 \times 8$, and $4 \times 4$. To keep fine target information and get the prompt embedding in the same format as other tasks, a merging module is applied. Namely, all levels of features are first upsampled to $32 \times 32$ then added, and flattened as the final prompt embedding $F_p' \in \mathbb{R}^{1024 \times d}$.

The prompt generation process can be formulated as

$$F_p = \begin{cases} 
\text{Enc}_L^{\text{ref}}(\text{expression}) & \text{expression-guided} \\
\text{Enc}_L^{\text{ref}}(\text{concat(categories)}) & \text{category-guided} \\
\text{merge}([\text{Enc}_V^{\text{ref}}(\text{template, prior})]) & \text{annotation-guided}
\end{cases}$$

### 3.2. Image-Prompt Feature Fusion

In parallel with the prompt generation, the whole current image is passed through another visual encoder $\text{Enc}_V$, obtaining hierarchical visual features $F_v$. To enhance the original prompt embedding by the image contexts and to make the original visual features prompt-aware, an early fusion module is adopted. To be specific, a bi-directional cross-attention module (Bi-XAtt) is used to retrieve information from different inputs, and then the retrieved representations are added to the original features. This process can be formulated as

$$F_{p2v}, F_{v2p} = \text{Bi-XAtt}(F_v, F_p)$$

$$F_v' = F_v + F_{p2v}; F_p' = F_p + F_{v2p}$$

Different from GLIP [54], which adopts 6 vision-language fusion layers and 6 additional BERT layers for feature enhancement, our early fusion module is much more efficient.

### 3.3. Object Discovery and Retrieval

With discriminative visual and prompt representations, the next crucial step is to transform input features into instances for various perception tasks. UNINEXT adopts the encoder-decoder architecture proposed by Deformable DETR [131] for its flexible query-to-instance fashion. We introduce the detailed architectures as follows.

The Transformer encoder takes hierarchical prompt-aware visual features as the inputs. With the help of efficient Multi-scale Deformable Self-Attention [131], target information from different scales can be fully exchanged, bringing stronger instance features for the subsequent instance decoding. Besides, as performed in two-stage Deformable DETR [131], an auxiliary prediction head is appended at the end of the encoder, generating $N$ initial reference points with the highest scores as the inputs of the decoder.

The Transformer decoder takes the enhanced multi-scale features, $N$ reference points from the encoder, as well as $N$ object queries as the inputs. As shown in previous works [73, 100, 104, 123], object queries play a critical role in instance perception tasks. In this work, we attempt two query generation strategies: (1) static queries which do not change with images or prompts. (2) dynamic queries conditioned on the prompts. The first strategy can be easily implemented
with \texttt{nn.Embedding}(N,d). The second one can be performed by first pooling the enhanced prompt features $F_p'$ along the sequence dimension, getting a global representation, then repeating it by $N$ times. The above two methods are compared in Sec 4.3 and we find that static queries usually perform better than dynamic queries. The potential reason could be that static queries contain richer information and possess better training stability than dynamic queries. With the help of the deformable attention, the object queries can efficiently retrieve prompt-aware visual features and learn strong instance embedding $F_{\text{ins}} \in \mathbb{R}^{N \times d}$.

At the end of the decoder, a group of prediction heads is exploited to obtain the final instance predictions. Specifically, an instance head produces both boxes and masks of the targets. Besides, an embedding head \cite{105} is introduced for associating the current detected results with previous trajectories in MOT, MOTS, and VIS. Until now, we have mined $N$ potential instance proposals, which are represented with gray masks in Figure 2. However, not all proposals are what the prompts really refer to. Therefore, we need to further retrieve truly matched objects from these proposals according to the prompt embeddings as demonstrated in the right half of Figure 2. Specifically, given the prompt embeddings $F_p'$ after early fusion, for category-guided tasks, we take the embedding of each category name as a weight matrix $W \in \mathbb{R}^{1 \times d}$. Besides, for expression-guided and annotation-guided tasks, the weight matrix $W$ is obtained by aggregating the prompt embedding $F_p'$ using global average pooling (GAP) along the sequence dimension.

$$W = \begin{cases} F_p'[i], & i \in \{0, 1, ..., C - 1\} \text{ category} \\ \frac{1}{L} \sum_{i=0}^{L} F_p'(i, j) & \text{expression/annotation} \end{cases}$$

Finally, the instance-prompt matching scores $S$ can be computed as the matrix multiplication of the target features and the transposed weight matrix. $S = F_{\text{ins}}W^\top$. Following previous work \cite{54}, the matching scores can be supervised by Focal Loss \cite{60}. Different from previous fixed-size classifiers \cite{131}, the proposed retrieval head selects objects by the prompt-instance matching mechanism. This flexible design enables UNINEXT to jointly train on enormous datasets with diverse label vocabularies from different tasks, learning universal instance representations.

3.4. Training and Inference

Training. The whole training process consists of three consecutive stages: (1) general perception pretraining (2) image-level joint training (3) video-level joint training.

In the first stage, we pretrain UNINEXT on the large-scale object detection dataset Objects365 \cite{88} for learning universal knowledge about objects. Since Objects365 does not have mask annotations, we introduce two auxiliary losses proposed by BoxInst \cite{92} for training the mask branch. The loss function can be formulated as

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{retrieve}} + \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{boxinst}}^\text{mask}$$

Then based on the pretrained weights of the first stage, we finetune UNINEXT jointly on image datasets, namely COCO \cite{61} and the mixed dataset of RefCOCO \cite{121}, RefCOCO+ \cite{121}, and RefCOCOg \cite{77}. With manually labeled mask annotations, the traditional loss functions like Dice Loss \cite{75} and Focal Loss \cite{60} can be used for the mask learning. After this step, UNINEXT can achieve superior performance on object detection, instance segmentation, REC, and RES.

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{retrieve}} + \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{mask}}$$

Finally, we further finetune UNINEXT on video-level datasets for various downstream object tracking tasks and benchmarks. In this stage, the model is trained on two frames randomly chosen from the original videos. Besides, to avoid the model forgetting previously learned knowledge on image-level tasks, we also transform image-level datasets to pseudo videos for joint training with other video datasets. In summary, the training data in the third stage includes pseudo videos generated from COCO \cite{61}, RefCOCO/g/+ \cite{77, 121, 121}, SOT&VOS datasets (GOT-10K \cite{42}, LaSOT \cite{28}, TrackingNet \cite{76}, and Youtube-VOS \cite{107}), MOT&VIS datasets (BDD100K \cite{120}, VIS19 \cite{112}, OV1S \cite{81}), and R-VOS dataset Ref-Youtube-VOS \cite{86}. Meanwhile, a reference visual encoder for SOT&VOS and an extra embedding head for association are introduced and optimized in this period.

$$\mathcal{L}_{\text{stage3}} = \mathcal{L}_{\text{retrieve}} + \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{embed}}$$

Inference. For category-guided tasks, UNINEXT predicts instances of different categories and associates them with previous trajectories. The association proceeds in an online fashion and is purely based on the learned instance embedding following \cite{79, 105}. For expression-guided and annotation-guided tasks, we directly pick the object with the highest matching score with the given prompt as the final result. Different from previous works \cite{94, 104} restricted by the offline fashion or complex post-processing, our method is simple, online, and post-processing free.

4. Experiments

4.1. Implementation Details

We attempt two different backbones, ResNet-50 \cite{38} and ConvNeXt-Large \cite{66} as the visual encoder. We adopt BERT \cite{24} as the text encoder and its parameters are trained in the first and second training stages while being frozen in
the last training stage. The Transformer encoder-decoder architecture follows [131] with 6 encoder layers and 6 decoder layers. The number of object queries \( N \) is set to 900. The optimizer is AdamW [67] with weight decay of 0.05. The model is trained on 32 and 16 A100 GPUs for Objects365 pretraining and other stages respectively. More details can be found in the supplementary materials.

4.2. Evaluations on 10 Tasks

We compare UNINEXT with task-specific counterparts in 20 datasets. In each benchmark, the best two results are indicated in bold and with underline. UNINEXT in all benchmarks uses the same model parameters.

Object Detection and Instance Segmentation. We compare UNINEXT with state-of-the-art object detection and instance segmentation methods on VOC val2017 (5k images) and test-dev split (20k images) respectively. As shown in Table 1, UNINEXT surpasses state-of-the-art query-based detector DN-Deformable DETR [53] by 2.7 AP. By replacing ResNet-50 [38] with stronger ConvNeXt-Large [66] as the backbone, UNINEXT achieves a box AP of 58.1, surpassing competitive rivals Cascade Mask-RCNN [8] by 3.3. Besides, the results of instance segmentation are shown in Table 2. With the same ResNet-50 backbone, UNINEXT outperforms state-of-the-art QueryInst by 4.3 AP and 6.2 AP\(_L\). When using ConvNeXt-Large as the backbone, UNINEXT achieves a mask AP of 49.6, surpassing Cascade Mask R-CNN [8] by 2.0.

REC and RES. RefCOCO [121], RefCOCO+ [121], and RefCOCOg [71] are three representative benchmarks for REC and RES proposed by different institutions. Following previous literature, we adopt Precision@0.5 and overall intersection-over-union (oIoU) as the evaluation metrics for REC and RES respectively and results are rounded to two decimal places. As shown in Table 3 and Table 4, our method with ResNet-50 backbone surpasses all previous approaches on all splits. Furthermore, when using ConvNeXt-Large backbone, UNINEXT obtains new state-of-the-art results, exceeding the previous best method by a large margin. Especially on RES, UNINEXT-L outperforms LAVT [116] by 8.63 on average.

SOT. We compare UNINEXT with state-of-the-art SOT methods on four large-scale benchmarks: LaSOT [28], LaSOT-ext [27], TrackingNet [76], and TNL-2K [98]. These benchmarks adopt the area under the success curve (AUC), normalized precision (P\(_{Norm}\)), and precision (P) as the evaluation metrics and include 280, 150, 511, and 700 videos in the test set respectively. As shown in Table 5, UNINEXT achieves the best results in terms of AUC and P among all trackers with ResNet-50 backbone. Especially on TNL-2K, UNINEXT outperforms the second best method TransT [14] by 5.3 AUC and 5.8 P respectively. Besides, UNINEXT with ConvNeXt-Large backbone obtains the best AUC on all four benchmarks, exceeding Unicorn [109] with the same backbone by 3.9 on LaSOT.

VOS. The comparisons between UNINEXT with previous semi-supervised VOS methods are demonstrated in Table 6. DAVIS-2017 [80] adopts region similarity \( \mathcal{J} \), contour accuracy \( \mathcal{F} \), and the averaged score \( \mathcal{J} \& \mathcal{F} \) as the metrics. Similarly, Youtube-VOS 2018 [107] reports \( \mathcal{J} \) and \( \mathcal{F} \) for both seen and unseen categories, and the averaged overall score \( \mathcal{G} \). UNINEXT achieves the best results among all non-memory-based methods, further bridging the per-
Table 5. State-of-the-art comparison on SOT.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>LaSOT [28]</th>
<th>LaSOT_ext [27]</th>
<th>TrackingNet [76]</th>
<th>TNL-2K [98]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AUC</td>
<td>P$_{Norm}$</td>
<td>P</td>
<td>AUC</td>
</tr>
<tr>
<td>PrDiMP [21]</td>
<td></td>
<td>59.8</td>
<td>68.8</td>
<td>60.8</td>
<td>-</td>
</tr>
<tr>
<td>LTMU [20]</td>
<td>ResNet-50</td>
<td>57.2</td>
<td>-</td>
<td>57.2</td>
<td>41.4</td>
</tr>
<tr>
<td>TransT [14]</td>
<td></td>
<td>64.9</td>
<td>73.8</td>
<td>69.0</td>
<td>-</td>
</tr>
<tr>
<td>KeepTrack [72]</td>
<td></td>
<td>67.1</td>
<td>77.2</td>
<td>70.2</td>
<td>-</td>
</tr>
<tr>
<td>UNINEXT</td>
<td></td>
<td>69.2</td>
<td>77.1</td>
<td>75.5</td>
<td>51.2</td>
</tr>
<tr>
<td>SimTrack [10]</td>
<td>ViT-B</td>
<td>69.3</td>
<td>78.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OSTAR [18]</td>
<td></td>
<td>71.1</td>
<td>81.1</td>
<td>77.6</td>
<td>50.5</td>
</tr>
<tr>
<td>Unicorn [109]</td>
<td></td>
<td>68.5</td>
<td>76.6</td>
<td>74.1</td>
<td>-</td>
</tr>
<tr>
<td>UNINEXT</td>
<td>ConvNeXt-L</td>
<td>72.4</td>
<td>80.7</td>
<td>78.9</td>
<td>54.4</td>
</tr>
</tbody>
</table>

Table 6. State-of-the-art comparison on VOS.

<table>
<thead>
<tr>
<th>Method</th>
<th>YT-VOS 2018 val [107]</th>
<th>DAVIS 2017 val [80]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mathcal{G}$</td>
<td>$\mathcal{J}_s$</td>
</tr>
<tr>
<td>STM [78]</td>
<td>79.4</td>
<td>79.7</td>
</tr>
<tr>
<td>CFBI [117]</td>
<td>81.4</td>
<td>81.1</td>
</tr>
<tr>
<td>STC [18]</td>
<td>83.0</td>
<td>81.9</td>
</tr>
<tr>
<td>XM [17]</td>
<td>86.1</td>
<td>85.1</td>
</tr>
<tr>
<td>SiamMask [96]</td>
<td>52.8</td>
<td>60.2</td>
</tr>
<tr>
<td>Unicorn [109]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Siam R-CNN [94]</td>
<td>73.2</td>
<td>73.5</td>
</tr>
<tr>
<td>TVOS [127]</td>
<td>67.8</td>
<td>67.1</td>
</tr>
<tr>
<td>FRTM [85]</td>
<td>72.1</td>
<td>72.3</td>
</tr>
<tr>
<td>UNINEXT-R50</td>
<td>70.7</td>
<td>70.8</td>
</tr>
<tr>
<td>UNINEXT-L</td>
<td>78.1</td>
<td>79.1</td>
</tr>
</tbody>
</table>

Table 7. State-of-the-art comparison on MOT.

<table>
<thead>
<tr>
<th>Method</th>
<th>Split</th>
<th>mMOTA$^\dagger$</th>
<th>mDF1$^\dagger$</th>
<th>MOTA$^\dagger$</th>
<th>IDF1$^\dagger$</th>
<th>ID Sw.$\downarrow$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yu et al. [120]</td>
<td>val</td>
<td>25.9</td>
<td>44.5</td>
<td>56.9</td>
<td>66.8</td>
<td>8315</td>
</tr>
<tr>
<td>QDTrack [79]</td>
<td>val</td>
<td>36.6</td>
<td>50.8</td>
<td>63.5</td>
<td>71.5</td>
<td>6262</td>
</tr>
<tr>
<td>Unicorn [109]</td>
<td>val</td>
<td>41.2</td>
<td>54.0</td>
<td>66.6</td>
<td>71.3</td>
<td>10876</td>
</tr>
<tr>
<td>UNINEXT-L</td>
<td>val</td>
<td>41.8</td>
<td>54.9</td>
<td>64.6</td>
<td>68.7</td>
<td>9134</td>
</tr>
</tbody>
</table>

Table 8. State-of-the-art comparison on MOTS.

<table>
<thead>
<tr>
<th>Method</th>
<th>Online</th>
<th>mMOTS$^\dagger$</th>
<th>mMOTS$^p$</th>
<th>mDF1$^\dagger$</th>
<th>ID Sw.$\downarrow$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaskTrackRCNN [112]</td>
<td>✓</td>
<td>12.3</td>
<td>59.9</td>
<td>26.2</td>
<td>9116</td>
</tr>
<tr>
<td>STImSeg [11]</td>
<td>✓</td>
<td>12.2</td>
<td>58.2</td>
<td>25.4</td>
<td>8732</td>
</tr>
<tr>
<td>QDTrack-mots [79]</td>
<td>✓</td>
<td>22.5</td>
<td>59.6</td>
<td>40.8</td>
<td>1340</td>
</tr>
<tr>
<td>PCAN [47]</td>
<td>✓</td>
<td>27.4</td>
<td>66.7</td>
<td>45.1</td>
<td>876</td>
</tr>
<tr>
<td>VMT [46]</td>
<td>✓</td>
<td>28.7</td>
<td>67.3</td>
<td>45.7</td>
<td>823</td>
</tr>
<tr>
<td>Unicorn [109]</td>
<td>✓</td>
<td>29.6</td>
<td>87.7</td>
<td>44.2</td>
<td>1731</td>
</tr>
<tr>
<td>UNINEXT-L</td>
<td>✓</td>
<td>32.0</td>
<td>60.2</td>
<td>45.4</td>
<td>1634</td>
</tr>
</tbody>
</table>

Performance gap between non-memory-based approaches and memory-based ones. Furthermore, compared with traditional memory-based methods [18,78], UNINEXT does not rely on the intermediate mask predictions. This leads to constant memory consumption and zero cumulative error, enabling UNINEXT to handle long sequences of any length.

**MOT.** We compare UNINEXT with state-of-the-art MOT methods on BDD100K [120], which requires tracking 8 classes of instances in the autonomous driving scenario. Except for classical evaluation metrics Multiple-Object Tracking Accuracy (MOTA), Identity F1 Score (IDF1), and Identity Switches (IDS), BDD100K additionally introduces mMOTA, and mIDF1 to evaluate the average performance across 8 classes. As shown in Table 7, UNINEXT surpasses Unicorn [109] by 0.6 mMOTA and 0.9 mIDF1 respectively.

**MOTS.** Similar to MOT, BDD100K MOTS Challenge [120] evaluates the performance on multi-class tracking by mMOTSA, mMOTSP, mIDF1, and ID Sw. This benchmark contains 37 sequences with mask annotations in the validation set. As shown in Table 8, UNINEXT achieves state-of-the-art performance, surpassing the previous best method Unicorn [109] by 2.4 mMOTSA.

**VIS.** We compare UNINEXT against state-of-the-art VIS methods on Youtube-VIS 2019 [112] and OVIS [81] validation sets. Specifically, Youtube-VIS 2019 and OVIS have 40 and 25 object categories, containing 302 and 140 videos respectively in the validation set. Both benchmarks take AP as the main metric. When using the same ResNet-50 backbone, UNINEXT obtains the best results on both datasets. Especially on more challenging OVIS, UNINEXT exceeds the previous best method IDOL [105] by 3.8 AP. As shown in Table 9, when using stronger ConvNeXt-Large backbone, UNINEXT achieves much higher AP (64.3 on Youtube-VIS 2019 and 41.1 on OVIS). Although the performance is slightly lower than IDOL, UNINEXT can handle diverse label vocabularies [61,81,112,120] by a single model, being much more flexible than IDOL.

**R-VOS.** Ref-Youtube-VOS [86] and Ref-DAVIS17 [48] are two popular R-VOS benchmarks, which are constructed by introducing language expressions for the objects in the original Youtube-VOS [107] and DAVIS17 [80] datasets. As same as semi-supervised VOS, region similarity $\mathcal{J}$, contour accuracy $\mathcal{F}$, and the averaged score $\mathcal{J}$&$\mathcal{F}$ are adopted as the metrics. As demonstrated in Table 10, UNINEXT outperforms all previous R-VOS approaches by a large margin, when using the same ResNet-50 backbone. Especially on Ref-DAVIS17, UNINEXT exceeds previous best Refer-Former [104] by 5.4 $\mathcal{J}$&$\mathcal{F}$. Furthermore, when adopting stronger ConvNeXt-Large backbone, UNINEXT achieves new state-of-the-art $\mathcal{J}$&$\mathcal{F}$ (66.2 on Ref-Youtube-VOS and 66.7 on Ref-DAVIS17). Besides, different from offline Ref-
Former, UNINEXT works in a flexible online fashion, making it applicable to ongoing videos in the real world.

### 4.3. Ablations and Other Analysis

In this section, we conduct component-wise analysis for better understanding our method. All models take ResNet-50 as the backbone. The methods are evaluated on five benchmarks (COCO [61], RefCOCO [121], Youtube-VOS [107], Ref-Youtube-VOS [86], and Youtube-VIS 2019 [112]) from five tasks (object detection, REC, VOS, R-VOS, and VIS). The results are shown in Table 11.

**Fusion.** To study the effect of feature fusion between visual features and prompt embeddings, we implement a variant without any early fusion. In this version, prompt embeddings do not have an influence on proposal generation but are only used in the final object retrieval process. Experiments show that early fusion has the greatest impact on VOS, the performance on VOS drops drastically by 21.4 \( AP \) without feature fusion. This is mainly caused by the following reasons (1) Without the guidance of prompt embeddings, the network can hardly find rare referred targets like trees and sinks. (2) Without early fusion, the network cannot fully exploit fine mask annotations in the first frame, causing degradation of the mask quality. Besides, the removal of feature fusion also causes performance drop of 2.3 \( AP \) on REC and RVOS respectively, showing the importance of early fusion in expression-guided tasks. Finally, feature fusion has minimum influence on object detection and VIS. This can be understood because both tasks aim to find all objects as completely as possible rather than locating one specific target referred by the prompt.

**Queries.** We compare two different query generation strategies: static queries by \( nn.Embedding(N, d) \) and dynamic queries conditioned on the prompt embeddings. Experiments show that dynamic queries perform slightly better than static queries on the first four tasks. However, static queries outperform dynamic ones by 2.8 \( AP \) on the VIS task, obtaining higher overall performance. A potential reason is that \( N \) different object queries can encode richer inner relationship among different targets than simply copying the pooled prompt by \( N \) times as queries. This is especially important for VIS because targets need to be associated according to their affinity in appearance and space.

**Unification.** We compare two different model design philosophies, one unified model or multiple task-specific models. Except for the unified model, we also retrain five task-specific models only on data from corresponding tasks. Experiments show that the unified model achieves better performance than its task-specific counterparts on five tasks, demonstrating the superiority of the unified formulation and joint training on all instance perception tasks. Finally, the unified model can save tons of parameters, being much more parameter-efficient.

### 5. Conclusions

We propose UNINEXT, a universal instance perception model of the next generation. For the first time, UNINEXT unifies 10 instance perception tasks with a prompt-guided object discovery and retrieval paradigm. Extensive experiments demonstrate that UNINEXT achieves superior performance on 20 challenging benchmarks with a single model with the same model parameters. We hope that UNINEXT can serve as a solid baseline for the research of instance perception in the future.

**Acknowledgement.** We would like to thank the reviewers for their insightful comments. The paper is supported in part by the National Key R&D Program of China under Grant No. 2018AAA0102001, 2022ZD0161000 and National Natural Science Foundation of China under grant No. 62935342, U1903215, 62022021.
References


[26] Zihan Ding, Tianrui Hui, Shaofei Huang, Si Liu, Xuan Luo, Junshi Huang, and Xiaoming Wei. Progressive multimodal interaction network for referring video object segmentation. The 3rd Large-scale Video Object Segmentation Challenge, 2021.


[37] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In ICCV. 2017. 1, 2, 3
[40] Richang Hong, Daqing Liu, Xiaoyu Mo, Xiangnan He, and Hanwang Zhang. Learning to compose and reason with language tree structures for visual grounding. TPAMI, 2019. 2
[42] Lianghua Huang, Xin Zhao, and Kaiqi Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. TPAMI, 2019. 5
[52] Dezhuang Li, Ruoqi Li, Lijun Wang, Yifan Wang, Jingqing Qi, Lu Zhang, Ting Liu, Qingquan Xu, and Huchuan Lu. You only infer once: Cross-modal meta-transfer for referring video object segmentation. In AAAI, 2022. 8
[57] Yue Liao, Si Liu, Guanbin Li, Fei Wang, Yanjie Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In CVPR, 2020. 2
[63] Si Li, Tianrui Hui, Shaohei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. TPAMI, 2021. 6
[64] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In CVPR, 2018. 1
[65] Xihui Liu, Zihao Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In CVPR, 2019. 2
[69] Gen Luo, Yiyi Zhou, Rongrong Ji, Xiaoshuai Sun, Jingsong Su, Chia-Wen Lin, and Qi Tian. Cascade grouped


