A. Dataset and Settings.

In our experiments, we jointly train on two types of data: segmentation data with semantic labels and segmentation data with only pixel annotations (SA-1B [13]. For semantically-labeled data, we use COCO2017 [21] panoptic segmentation dataset with around 110K images. For SA-1B, we employ a 20% portion subset with around 2M images. We evaluate our model on a wide range of tasks and datasets with only visual prompts, including: 1. Open-set panoptic segmentation on COCO2017 [21] and ADE20K [48]; 2. Segmentation in the wild (SegInW) [50] which includes 25 instance segmentation datasets; 3. Object detection in the wild (ODinW) [19] that encompasses over 35 datasets; 4. Zero-shot Video object segmentation (VOS) on DAVIS2017 [28], DAVIS2016-Interactive [28], and Youtube-VOS 2018 [39].

B. Implementation Details

Our model framework is mainly based on Mask DINO [17], which is a unified framework for detection and segmentation. *DINOv* is a general encoder-decoder architecture composed of a vision encoder. We use Swin-T/L [25] as the vision encoder. As our decoder supports both generic query and point query, we adopt 300 latent generic queries following Mask DINO [16] and six level queries for each input point following Semantic-SAM [17]. Especially, when using point query, we sample 10 foreground and 40 background points during training and employ grid sample for 20×20 points during inference. For inference on general segmentation and detection tasks, we use 16 in-context examples for each category by default. For VOS inference, we average eight previous frame predictions as the reference to segment the current frame.

C. Evaluation Metrics.

For all segmentation and detection tasks, we use standard evaluation metrics: PQ (Panoptic Quality) for panoptic segmentation, AP (Average Precision) for instance segmentation (mask AP) and detection (box AP), and mIoU (mean Intersection over Union) for semantic segmentation. For VOS tasks, we follow previous semi-supervised models to use region similarity J and contour accuracy F. We also adopt the averaged score J&F as the metric for DAVIS2017 and averaged overall score G for Youtube-VOS 2018. Note that Youtube-VOS 2018 also reports J and F for seen and unseen splits.

D. Video Object Segmentation Inference

Video object segmentation (VOS) aims to segment an interested object in a video by giving text or visual clues. Our model focuses on the semi-supervised setting, which segments a particular object throughout a video by giving visual clues in the first frame. When doing VOS, an intuitive way is to first extract reference visual prompt features from the first frame image and the corresponding visual prompts with our prompt encoder. When processing each frame in a video, we are able to utilize reference visual prompt features in the first frame as in the current frame.

Table 8. **Ablation** of Inference Memory Length on DAVIS2017 with a SwinL backbone.

Method	Memory Length	DAVIS2017		
		J& F	J	F
DINOv-SwinT	1	62.1	58.7	65.4
DINOv-SwinT	2	69.6	66.7	72.6
DINOv-SwinT	4	71.5	68.7	74.3
DINOv-SwinT	8	72.3	69.8	74.8
DINOv-SwinT	16	68.0	65.4	70.7

In *DINOv*, as we train with visual in-context prompting with multiple examples for generic segmentation, we can also apply this strategy to VOS for better performance. More concretely, we also compute and store the reference visual features of the predicted mask in previous frames. These features, denoted as memory reference visual prompts, will be averaged together with the first frame's given prompt to construct the visual prompt of the current frame. We employ a priority queue to manage the memory. For simplicity, the priority score of each prompt is positively correlated to the frame number, which indicates that we only store the memory prompts that are near the current frame in time sequence. By default, the memory length is set to 8. In Tab. 8, we show the influence of using different number of memory length.

E. Pseudo Code of Prompt Sampling

In Algorithm. 1, we provide the pseudo code for prompt sampling in our generic segmentation task.

F. Visualization

We show the visualization for open-set segmentation in Fig. 7, in which the user can scribble on a reference image and our model can segment all the objects of the same semantic catogory in the target image. We also show our model can do panoptic segmentation in Fig. 8.

Algorithm 1: Pseudo code of Prompt Sampling for Generic Segmentation Task.

```
# Inputs: A list of encoded reference visual prompt F with length M, M is the total number of possible
  prompting examples. The ground-truth semantic category (i.e., dogs) of each reference visual prompt forms
  another list C with length M. During training, F is acquired from a batch of training images (i.e., 64
  images). During inference, the batch is the whole training image set.
# Variables: Defined maximum in-context length N for each semantic category.
# Output: Query visual prompt Q_p
def Prompt_Sample(F):
    C=Unique(C);# C is a list that contains all the semantic categories in this training batch.
    F_c=Dict();# F_c is visual prompt dict, where key is the semantic category and value are the reference prompt
      features.
    F_c[c]=[] for c in C;# Initialize visual prompt dict by semantic category.
    F_c[c].append(f) for c, f in zip(C, F);# Gather visual prompts of the same semantic category.
    n = Randint(1, N);# Randomly select the number of in-context examples.
    S_c = RandomSelect(F[c], n) for c in C;# For each semantic category, randomly select n prompts to represent
      a semantic category.
    Q_p = [Aggregate(S_c[c]) for c in C];# Perform prompt aggregation to get one reference prompt token from
      multiple in-context prompt features for each semantic category.
```

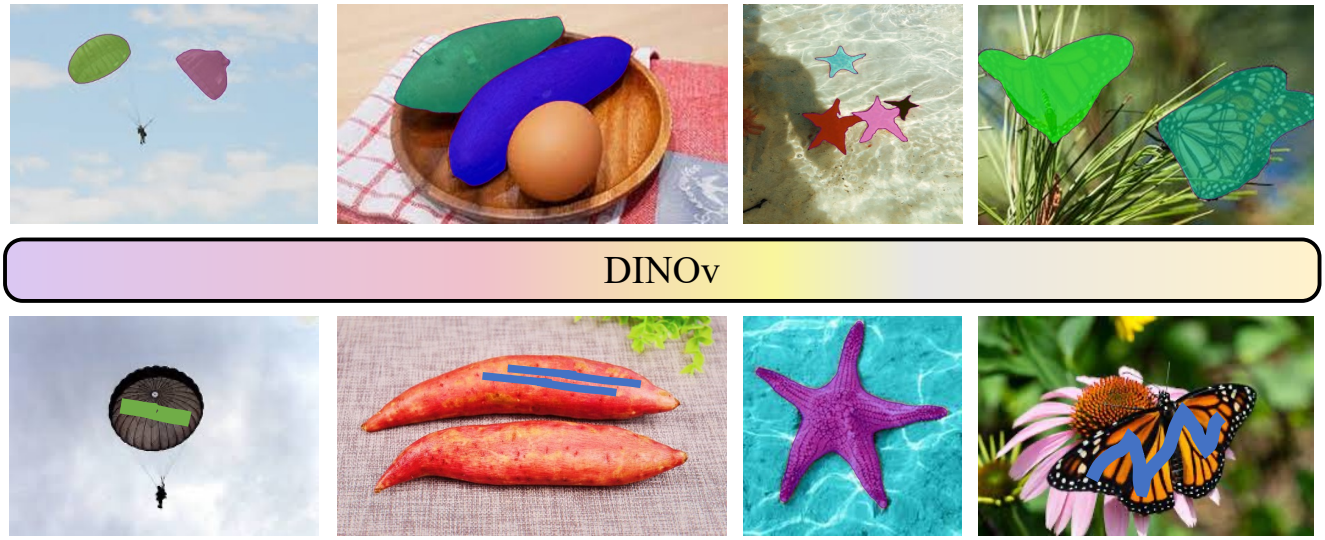

Figure 7. *DINOv* can do open-set segmentation by giving visual prompts.

Figure 8. *DINOv* can do panoptic segmentation by giving visual prompts. We randomly sample some visual prompts of the same semantic concept as visual in-context prompts on COCO.