TOIST: Task Oriented Instance Segmentation Transformer with Noun-Pronoun Distillation

Pengfei Li[1] **Beiwen Tian**[1] **Yongliang Shi**[1] **Xiaoxue Chen**[1]
Hao Zhao[2,3] **Guyue Zhou**[1] **Ya-Qin Zhang**[1]
[1]AIR, Tsinghua University [2] Peking University [3]Intel Labs
li-pf22@mails.tsinghua.edu.cn
zhao-hao@pku.edu hao.zhao@intel.com

Abstract

Current referring expression comprehension algorithms can effectively detect or segment objects indicated by nouns, but how to understand verb reference is still under-explored. As such, we study the challenging problem of task oriented detection, which aims to find objects that best afford an action indicated by verbs like *sit comfortably on*. Towards a finer localization that better serves downstream applications like robot interaction, we extend the problem into task oriented instance segmentation. A unique requirement of this task is to select *preferred* candidates among possible alternatives. Thus we resort to the transformer architecture which naturally models pair-wise query relationships with attention, leading to the TOIST method. In order to leverage pre-trained noun referring expression comprehension models and the fact that we can access privileged noun ground truth during training, a novel noun-pronoun distillation framework is proposed. Noun prototypes are generated in an unsupervised manner and contextual pronoun features are trained to select prototypes. As such, the network remains noun-agnostic during inference. We evaluate TOIST on the large-scale task oriented dataset COCO-Tasks and achieve +10.9% higher $\mathrm{mAP}^{\mathrm{box}}$ than the best-reported results. The proposed noun-pronoun distillation can boost $\mathrm{mAP}^{\mathrm{box}}$ and $\mathrm{mAP}^{\mathrm{mask}}$ by +2.8% and +3.8%. Codes and models are publicly available at `https://github.com/AIR-DISCOVER/TOIST`.

1 Introduction

As benchmarked by the RefCOCO, RefCOCO+ [27][61] and RefCOCO-g [45] datasets, noun referring expression comprehension models have seen tremendous progress, thanks to large-scale vision-language pre-training models like VL-BERT [57], VilBERT [43], OSCAR [33], UNITER [10], 12-in-1 [44] and MDETR [26]. As shown in the left top part of Fig. 1, these algorithms take noun prompts like *hatchback car* as inputs and generate a bounding box or an instance mask of that car. However, in real-world applications like intelligent service robots, system inputs usually come in the form of *affordance* (i.e., the capability to support an action or say a verb phrase). Whether modern vision-language model designs can effectively understand verb reference remains under-explored.

To this end, we focus on the challenging problem of task oriented detection, as introduced by the COCO-Tasks benchmark [55]. As shown in the right top part of Fig. 1, a task oriented detector outputs three boxes of forks as they can be used to *smear butter*. We also extend the problem to an upgraded instance segmentation version using existing COCO masks [40], as the masks can provide finer localization. When RGB-D pairs are available, instance masks can be used to obtain object point clouds. When image sequences are available, instance masks can be used to reconstruct objects using visual hull [31][11][67]. As such, the newly proposed task oriented instance segmentation formulation (Fig. 1 bottom) is useful for down-stream robot interaction applications.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

While noun referring expression comprehension datasets aim to minimize ambiguity [45], an interesting and challenging feature of task oriented detection/segmentation is the intrinsic ambiguity. For example, in the right top panel of Fig. 1, the pizza peel can also be used to *smear butter*. If we have neither forks nor pizza peels at hand, it is still possible to use the plate to *smear butter*. Another example is shown in Fig. 1 bottom. When we consider an object to *step on*, the chair is a better choice because the sofa is soft and the table is heavy to move. When the need switches to *sit comfortably on*, sofas are obviously the best candidates. In one word, objects that afford a verb are ambiguous and the algorithm needs to model *preference*.

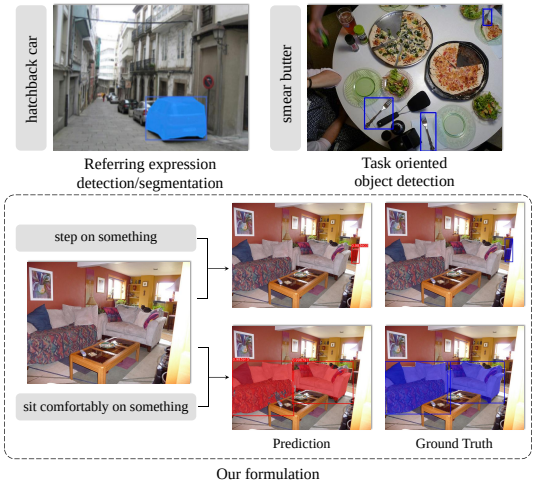


Figure 1: Left top: Noun referring expression comprehension. Right top: Task oriented detection. Bottom: Task oriented instance segmentation.

To this end, current models [55] use a two-stage pipeline, in which objects are firstly detected then relatively ranked. Inspired by the success of DETR-like methods [6][65][41] and the advantage of the attention mechanism in revealing the relationship between visual elements [34][64], we resort to the transformer architecture as it imposes self-attention on object queries thus naturally models the pair-wise relative preference between object candidates. Our one-stage method is named as **T**ask **O**riented **I**nstance **S**egmentation **T**ransformer and abbreviated as TOIST. Transformers are considered to be data hungry [5][14], but obtaining large-scale visually grounded verb reference data with relative preference (e.g., COCO-Tasks [55]) is difficult. This inspires us to explore the possibility of reusing knowledge in noun referring expression comprehension models. We propose to use pronouns like *something* as a proxy and distill knowledge from noun embedding prototypes generated by clustering.

Specifically, we first train a TOIST model with verb-noun input (e.g., *step on chair* for the bottom panel of Fig. 1 bottom), using the privileged noun ground truth. But during inference, we cannot access the noun *chair*, thus we train the second TOIST model with verb-pronoun input (e.g., *step on something*) and distill knowledge from the first TOIST model. As such, the second TOIST model remains noun-agnostic during inference and achieves better performance than directly training a model with verb-pronoun input. This framework is named as *noun-pronoun distillation*. Although leveraging knowledge from models with privileged information has been used in robotics research like autonomous driving [7] and quadrupedal locomotion [32], the proposed paradigm of distilling privileged noun information into pronoun features is novel, to the best of our knowledge.

To summarize, this paper has the following four contributions:

- We upgrade the task oriented detection task into task oriented instance segmentation and provide the first solution to it. Although this is a natural extension, this new formulation is of practical value to robotics applications.
- Unlike existing two-stage models that firstly detect objects then rank them, we propose the first transformer-based method TOIST, for task oriented detection/segmentation. It has only one stage and naturally models relative preference with self-attention on object queries.
- In order to leverage the privileged information in noun referring expression understanding models, we propose a novel noun-pronoun distillation framework. It improves TOIST by +2.8% and +3.8% for mAP^{box} and mAP^{mask}, respectively.
- We achieve new state-of-the-art (SOTA) results on the COCO-Tasks dataset, out-performing the best reported results by +10.9% mAP^{box}. Codes and models are publicly available.

2 Related Works

Vision and Language. Connecting vision and language is a long-existing topic for visual scene understanding. Barnard et al. [4] propose a system that translates image regions into nouns. Babytalk

[30] is an early method that turns images into sentences, based upon conditional random fields. Visual question answering [3] aims to answer questions about an image, with potential applications in helping visually impaired people [19]. The CLEVR dataset [25] focuses on the reasoning ability of question answering models, thanks to a full control over the synthetic data. The Flickr30k benchmark [51] addresses the phrase grounding task that links image regions and descriptions. Vision-language navigation [2][16] aims to learn navigation policies that fulfill language commands. DALL-E [52] shows impressive text-to-image generation capability. Video dense captioning [28][15] generates language descriptions for detected salient regions. Visual madlibs [60] focuses on fill-the-blank question answering. Visual commonsense reasoning [62] proposes the more challenging task of justifying an answer. Referring expression detection [61][45] and segmentation [24] localize objects specified by nouns. Although this literature is very large with many problem formulations proposed, the task of detecting objects that afford *verbs* (e.g., COCO-Tasks[55]) is still under-explored.

Action and Affordance. Verbs, as the link between subjects and objects, have been extensively studied in the both vision [58][48] and language [23][46] communities before, while we focus on the vision side. It is difficult to define standalone *verb recognition* tasks, so existing problem formulations depend on the focus on subjects or objects. A simple taxonomy can be considered as such: recognizing *subjects and verbs* is named as action recognition [56][17]; recognizing *verbs and objects* is named as affordance recognition [66][59][13][9]; recognizing *triplets* is named as human-object-interaction recognition [35][63]. Task oriented object detection/segmentation, as an affordance understanding task, is very challenging and this study explores the noun-pronoun distillation framework to borrow rich knowledge from more visually grounded noun targets.

Knowledge Distillation. This technique is proposed in the deep learning literature by Hinton et al. [22] to distill knowledge from large models to small models. The insight is that soft logit targets generated by large models contain richer information that better serves as a supervision signal than hard one-hot labels. Knowledge distillation has been extended to show effectiveness in other domains like continual learning [37] and object detection [8]. The survey of Guo et al. [18] provides a comprehensive summary of knowledge distillation variants and applications. Most related to our method is the privileged knowledge distillation methods in robotics research [7][32], in which the teacher model has access to privileged information that the student cannot access during inference. Our noun-pronoun distillation method is tailored for the task oriented detection/segmentation problem which borrows rich knowledge from noun referring expression compression teacher models while still allowing the student model to be noun-agnostic.

3 Formulation

The problem is to detect and segment objects that are preferred to afford a specific task indicated by verb phrases, from an input image. Yet clearly defining **affordance** and **preference** is actually challenging so we follow the existing annotation protocol of the COCO-Tasks dataset [55]:

Affordance. Firstly, the target objects *afford* a specific task. In an input image, it is possible that no objects or multiple objects afford the task. And in the latter case, the objects may belong to multiple classes. For example, in the right top panel of Fig.1, nothing affords the task *sit comfortably on*. Instead, in the bottom panel, there are at least two sofas, a table and a chair that afford the task.

Preference. Secondly, we need to find the *best* ones from the objects which afford the task. In other words, the *preference* among multiple objects needs to be understood. In Fig.1 bottom, the two sofas are obviously more suitable for the task *sit comfortably on* than other objects enumerated above. Thus the ground truth objects for this task are the two sofas (covered in blue).

Now we formally define the task. The input is an RGB image $\mathbf{X}_v \in \mathbb{R}^{3\times \mathrm{H}_0 \times \mathrm{W}_0}$ (v represents *visual*) and a piece of text $\mathbf{X}_l$ (l represents *language*). $\mathbf{X}_l$ describes a specific task like *sit comfortably on*. The targets are bounding boxes $\mathbf{B}_{\mathrm{gt}} = [b_1, \ldots, b_{n_{\mathrm{gt}}}] \in [0,1]^{n_{\mathrm{gt}}\times 4}$ and instance segmentation masks $\mathbf{M}_{\mathrm{gt}} = [m_1, \ldots, m_{n_{\mathrm{gt}}}] \in \mathbb{R}^{n_{\mathrm{gt}}\times \mathrm{H}_0 \times \mathrm{W}_0}$ of target objects $\mathbf{O}_{\mathrm{gt}} = \langle \mathbf{B}_{\mathrm{gt}}, \mathbf{M}_{\mathrm{gt}} \rangle$, where $n_{\mathrm{gt}} \geq 0$ is the count of targets. The four components of $b_i \in [0,1]^4$ are normalized center coordinates, height and width of the i-th box. An algorithm f that addresses this problem works as:

$$f(\langle \mathbf{X}_v, \mathbf{X}_l \rangle) = \langle \mathbf{B}_{\mathrm{pred}}, \mathbf{M}_{\mathrm{pred}}, \mathbf{S}_{\mathrm{pred}} \rangle, \tag{1}$$

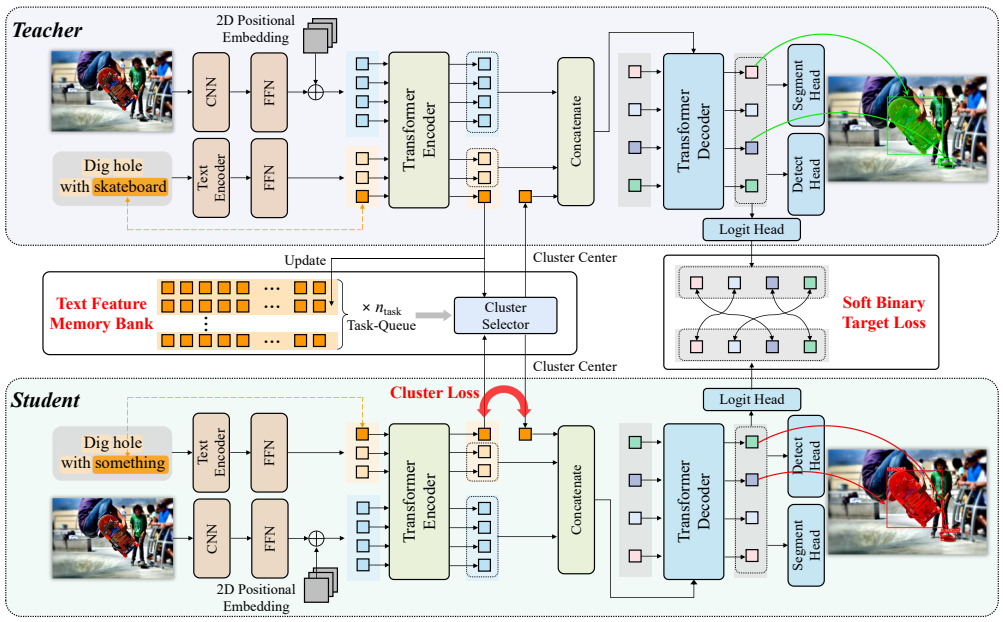


Figure 2: TOIST network architecture and the noun-pronoun distillation framework. The cluster loss and soft binary target loss distill privileged noun knowledge and preference knowledge respectively.

where $\mathbf{B}_{\text{pred}}$ and $\mathbf{M}_{\text{pred}}$ are predicted boxes and masks, respectively. $\mathbf{S}_{\text{pred}} = [\hat{s}_1, \ldots, \hat{s}_{n_{\text{pred}}}] \in [0, 1]^{n_{\text{pred}}}$ is the probability scores of the predicted objects being selected, which reflects preference. We denote the predicted objects as $\mathbf{O}_{\text{pred}} = \langle \mathbf{B}_{\text{pred}}, \mathbf{M}_{\text{pred}}, \mathbf{S}_{\text{pred}} \rangle$.

4 Method

We propose an end-to-end **T**ask **O**riented **I**nstance **S**egmentation **T**ransformer, abbreviated as TOIST (Section 4.1). Leveraging pre-trained noun referring expression comprehension models, we further adopt a teacher-student framework for noun-pronoun distillation (Section 4.2).

4.1 Task Oriented Instance Segmentation Transformer

The SOTA method [55] uses a two-stage pipeline to solve the problem. Taking a single image $\mathbf{X}_v$ as input, it first detects the bounding boxes $\mathbf{B}_{\text{pred}}$ of all objects with a Faster-RCNN [53]. Then it ranks $\mathbf{B}_{\text{pred}}$ with a GNN [36], predicting the probabilities $\mathbf{S}_{\text{pred}}$ of the objects being selected for a task. Our method differs from it in three ways: (1) We address the task with a one-stage architecture, allowing joint representation learning for detection and preference modeling. (2) We specify tasks using the text $\mathbf{X}_l$. (3) We predict instance masks $\mathbf{M}_{\text{pred}}$ along with bounding boxes.

We choose to build our method upon the transformer architecture [6], because the self-attention operators in the decoder can naturally model pair-wise relative preference between object candidates. As shown in Fig.2 bottom, TOIST contains three main components: a multi-modal encoder (color brown) to extract tokenized features, a transformer encoder (color green) to aggregate features of two modalities and a transformer decoder (color blue) to predict the most suitable objects with attention.

Two Input Forms. To find an object that affords the task of *dig hole*, the first step is to construct a task description input $\mathbf{X}_l$. To achieve this goal, we can extend the task name with the ground truth object category to the **verb-noun form** like *dig hole with skateboard* or with a pronoun to the **verb-pronoun form** like *dig hole with something*. While the former violates the noun-agnostic constraint during inference, it can be leveraged to improve the latter within the proposed noun-pronoun distillation framework, which will be detailed later. For a plain TOIST, the verb-pronoun form is selected as task description $\mathbf{X}_l$, which is fed into the multi-modal encoder along with visual input $\mathbf{X}_v$.

Multi-Modal Encoder. For $\mathbf{X}_v$, a pre-trained CNN-based backbone and a one-layer feed forward network (FFN) are leveraged to extract a low-resolution feature map $\mathbf{F}_v \in \mathbb{R}^{d\times \mathrm{H}\times \mathrm{W}}$. Flattening the spatial dimensions of $\mathbf{F}_v$ into one dimension, we obtain a sequence of tokenized feature vectors $\mathbf{V} = [v_1, \ldots, v_{n_v}] \in \mathbb{R}^{n_v \times d}$ (light blue squares ■ on the left of the transformer encoder in Fig.2), where $n_v = \mathrm{H} \times \mathrm{W}$. To preserve the spatial information, 2D positional embeddings are added to $\mathbf{V}$. For $\mathbf{X}_l$, we use a pre-trained text encoder and another FFN to produce corresponding feature vectors $\mathbf{L} = [l_1, \ldots, l_{n_l}] \in \mathbb{R}^{n_l \times d}$ (light orange squares ■), where n_l is the total count of language tokens. Among these n_l features, we denotes the one corresponding to the pronoun (or noun) token as l_{pron} (or l_{noun}) (dark orange squares ■). We concatenate these vectors and obtain the final feature sequence $[\mathbf{V}, \mathbf{L}] = [v_1, \ldots, v_{n_v}, l_1, \ldots, l_{n_l}] \in \mathbb{R}^{(n_v+n_l)\times d}$.

Transformer Encoder. The transformer encoder consists of n_{tr} sequential blocks of multi-head self-attention layers. Given the sequence of features $[\mathbf{V}, \mathbf{L}]$, it outputs processed feature sequence $[\mathbf{V}^{\text{tr}}, \mathbf{L}^{\text{tr}}] = [v_1^{\text{tr}}, \ldots, v_{n_v}^{\text{tr}}, l_1^{\text{tr}}, \ldots, l_{n_l}^{\text{tr}}] \in \mathbb{R}^{(n_v+n_l)\times d}$. The pronoun (or noun) feature l_{pron} (or l_{noun}) is encoded into $l_{\text{pron}}^{\text{tr}}$ (or $l_{\text{noun}}^{\text{tr}}$), which will be used for noun-pronoun distillation (Section 4.2). Here for the plain TOIST, we directly use the features $[\mathbf{V}^{\text{tr}}, \mathbf{L}^{\text{tr}}]$ for later processing.

Transformer Decoder. The transformer decoder consists of n_{tr} blocks of self-attention and cross-attention layers. It takes as input a set of learnable parameters serving as an object query sequence $\mathbf{Q} = [q_1, \ldots, q_{n_{\text{pred}}}] \in \mathbb{R}^{n_{\text{pred}}\times d}$. $[\mathbf{V}^{\text{tr}}, \mathbf{L}^{\text{tr}}]$ are used as keys and values for cross-attention layers. The outputs of the transformer decoder are feature vectors $\mathbf{Q}^{\text{tr}} = [q_1^{\text{tr}}, \ldots, q_{n_{\text{pred}}}^{\text{tr}}] \in \mathbb{R}^{n_{\text{pred}}\times d}$, which are projected to final results by three prediction heads. Specifically, the detect head predicts bounding boxes $\mathbf{B}_{\text{pred}}$ and the segment head outputs binary segmentation masks $\mathbf{M}_{\text{pred}}$. The logit head outputs logits $\mathbf{G}_{\text{pred}} = [\hat{\mathbf{g}}_1, \ldots, \hat{\mathbf{g}}_{n_{\text{pred}}}] \in \mathbb{R}^{n_{\text{pred}}\times n_{\max}}$, where $\hat{\mathbf{g}}_i = [\hat{g}_1^i, \ldots, \hat{g}_{n_{\max}}^i]$. $[\hat{g}_1^i, \ldots, \hat{g}_{n_l}^i]$ corresponds to text tokens $\mathbf{L} = [l_1, \ldots, l_{n_l}] \in \mathbb{R}^{n_l\times d}$. $[\hat{g}_{n_l+1}^i, \ldots, \hat{g}_{n_{\max}-1}^i]$ is used to pad $\hat{\mathbf{g}}_i$ to length $n_{\max}$ (by default $n_{\max}$=256) and the last one $\hat{g}_{n_{\max}}^i$ stands for the logit of "no-object". With the output logits, we define the preference score $\hat{s}_i \in \mathbf{S}_{\text{pred}}$ of each predicted object as:

$$\hat{s}_i = 1 - \frac{\exp\left(\hat{g}_{n_{\max}}^i\right)}{\sum_{j=1}^{n_{\max}} \exp\left(\hat{g}_j^i\right)}. \tag{2}$$

During training, as in DETR [6], a bipartite matching is computed between n_{pred} predicted objects $\mathbf{O}_{\text{pred}}$ and ground truth objects $\mathbf{O}_{\text{gt}}$ with the Hungarian algorithm [29]. The matched object predictions are supervised with L1 loss and Generalized Intersection over Union (GIoU) loss [54] for localization while Dice/F-1 loss [47] and Focal cross-entropy loss [39] for segmentation. We also adopt the soft-token prediction loss and the contrastive alignment loss used in MDETR [26]. But different from them, we do not use a single noun or pronoun as the ground truth token span for a matched object prediction. Instead, we use the whole verb-pronoun description as token span such that the network can understand the verbs rather than noun/pronoun only. The total loss for TOIST is:

$$\mathcal{L}_{\text{TOIST}} = \lambda_1 \mathcal{L}_{\text{l1}} + \lambda_2 \mathcal{L}_{\text{giou}} + \lambda_3 \mathcal{L}_{\text{dice}} + \lambda_4 \mathcal{L}_{\text{cross}} + \lambda_5 \mathcal{L}_{\text{token}} + \lambda_6 \mathcal{L}_{\text{align}}, \tag{3}$$

where λ_1~λ_6 are the weights of losses.

4.2 Noun-Pronoun Distillation

Motivation. As mentioned above, there are two possible input forms. Due to the privileged information of target names (nouns), TOIST using verb-noun input out-performs its counterpart by +11.8% and +12.0% on mAP^{box} and mAP^{mask}, as demonstrated in Table. 1. We use two preliminary experiments as a motivation: directly replacing the pronoun feature l_{pron} or $l_{\text{pron}}^{\text{tr}}$ in the verb-pronoun model with the corresponding noun feature l_{noun} or $l_{\text{noun}}^{\text{tr}}$ in the verb-noun model can boost performance (see Table 1). However, during inference, the noun of the ground truth object is unavailable. Thus we believe a properly designed noun-pronoun distillation framework can leverage rich knowledge from the verb-noun model without violating the noun-agnostic constraint.

Table 1: TOIST quantitative results under several different settings related to text.

Text related settings	mAP^{box}	mAP^{mask}
verb-pronoun input	41.3	35.2
verb-noun input	53.1 (+11.8)	47.2 (+12.0)
replace l_{pron} with l_{noun}	43.7 (+2.4)	37.3 (+2.1)
replace $l_{\text{pron}}^{\text{tr}}$ with $l_{\text{noun}}^{\text{tr}}$	41.9 (+0.6)	35.6 (+0.4)
noun-pronoun distillation	**44.1 (+2.8)**	**39.0 (+3.8)**

Distillation Framework Overview. Two TOIST models are trained simultaneously. The teacher (Fig.2 top) and the student (Fig.2 bottom) take as input verb-noun and verb-pronoun descriptions, respectively. A clustering distillation method with a memory bank and a tailored cluster loss is used to distill privileged object-centric knowledge from noun to pronoun (Fig.2 middle left). Besides, we also use a soft binary target loss imposed on $\mathbf{G}_{\text{pred}}$ to distill preference knowledge (Fig.2 middle right), in which $\mathbf{G}_{\text{pred}}$ are logits used to calculate preference scores $\mathbf{S}_{\text{pred}}$.

Clustering Distillation. Since one task can be afforded by objects of many different categories, we build a text feature memory bank to store noun features, with which a prototype can be selected and used to replace pronoun feature and distill knowledge. We term this process as clustering distillation. Specifically, we use $l^{\text{tr}}_{\text{pron}}$ and $l^{\text{tr}}_{\text{noun}}$ instead of l_{pron} and l_{noun} for this process. The reason is that the former ones are conditioned on the image input and verb tokens of the task by self-attention layers, and thus it is meaningful to select a cluster center that suits the image and the task input.

Memory and Selector. The size of the memory bank is $n_{\text{task}} \times n_{\text{mem}} \times d$. It consists of n_{task} queues of length n_{mem} for n_{task} tasks. During training, for each sample from task j, we update the j-th queue $\mathbf{L}^j_{\text{mem}} = [l^j_1, \ldots, l^j_{n_{\text{mem}}}]$ by adding the noun feature $l^{\text{tr}}_{\text{noun}}$ generated by the teacher model and removing the existing one closest to $l^{\text{tr}}_{\text{noun}}$. The updated queue is clustered with the K-means clustering method, leading to K cluster centers $\mathbf{L}^j_{\text{c}} = \{l^j_{c_1}, \ldots, l^j_{c_{\text{K}}}\}$. Then the student model uses a cluster selector, which is implemented as the nearest neighbor classifier, to select a prototype $l^j_{c_s} \in \mathbf{L}^j_{\text{c}}$ according to the pronoun feature $l^{\text{tr}}_{\text{pron}}$ and replace $l^{\text{tr}}_{\text{pron}}$ with $l^j_{c_s}$. Concatenating other tokens and the selected prototype together, the output of student transformer encoder $[v^{\text{tr}}_{s_1}, \ldots, v^{\text{tr}}_{s_n}, l^{\text{tr}}_{s_1}, \ldots, l^{\text{tr}}_{\text{pron}}, \ldots, l^{\text{tr}}_{s_{n_l}}]$ is modified into $[v^{\text{tr}}_{s_1}, \ldots, v^{\text{tr}}_{s_n}, l^{\text{tr}}_{s_1}, \ldots, l^j_{c_s}, \ldots, l^{\text{tr}}_{s_{n_l}}]$ and fed into the transformer decoder. To distill knowledge to the student transformer encoder, we define cluster loss as:

$$\mathcal{L}_{\text{cluster}} = \| l^{\text{tr}}_{\text{pron}} - l^j_{c_s} \|_2, \tag{4}$$

with which the privileged object-centric knowledge is distilled from clustered noun features to pronoun feature and further to the student TOIST encoder.

Preference Distillation. We use a soft binary target loss to distill preference knowledge from teacher to student. For an object query, we first define binary query probabilities of being positive-query or negative-query, which denotes whether or not the query is matched to a ground truth object target, as $\mathbf{p} = [p^{\text{pos}}, p^{\text{neg}}] \in \mathbb{R}^{1\times 2}$. The probabilities can be calculated by the softmax function:

$$p^{\text{pos}} = \frac{\sum_{j=1}^{n_{\max}-1} \exp(\hat{g}_j)}{\sum_{j=1}^{n_{\max}} \exp(\hat{g}_j)}, p^{\text{neg}} = \frac{\exp(\hat{g}_{n_{\max}})}{\sum_{j=1}^{n_{\max}} \exp(\hat{g}_j)}, \tag{5}$$

where $\hat{g}_j$ and $\hat{g}_{n_{\max}}$ represent the logits corresponding to i-th text token and "no-object" token, respectively. For all the object queries in teacher and student, the probability sequences are denoted by $\mathbf{P}_t = [\mathbf{p}_{t_1}, \ldots, \mathbf{p}_{t_{n_{\text{pred}}}}]$ and $\mathbf{P}_s = [\mathbf{p}_{s_1}, \ldots, \mathbf{p}_{s_{n_{\text{pred}}}}]$. Then we use the Hungarian algorithm to find a bipartite matching between the two sequences of object queries with $\mathbf{P}_t$ and $\mathbf{P}_s$. Formally, we search for a permutation of n_{pred} elements $\sigma \in \mathfrak{S}_{n_{\text{pred}}}$ which minimizes the matching loss:

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_{n_{\text{pred}}}}{\arg\min} \sum_{i}^{n_{\text{pred}}} \mathcal{L}_{\text{match}}(y_{t_i}, y_{s_{\sigma(i)}}), \tag{6}$$

where $y_{t_i} = (\hat{b}_{t_i} \mathbf{p}_{t_i})$ and $\hat{b}_{t_i}$ is the predicted bounding box. $\mathcal{L}_{\text{match}}$ is a linear combination of box prediction losses (L1 & GIoU) and KL-Divergence. The KL-Divergence $\mathcal{L}_{\text{KL}}$ can be written as:

$$\mathcal{L}_{\text{KL}}(\mathbf{p}_{t_i}, \mathbf{p}_{s_{\sigma(i)}}) = \mathbf{KL}\left(\mathbf{p}_{t_i} \| \mathbf{p}_{s_{\sigma(i)}}\right) = p^{\text{pos}}_{t_i} \log\left(\frac{p^{\text{pos}}_{t_i}}{p^{\text{pos}}_{s_{\sigma(i)}}}\right) + p^{\text{neg}}_{t_i} \log\left(\frac{p^{\text{neg}}_{t_i}}{p^{\text{neg}}_{s_{\sigma(i)}}}\right). \tag{7}$$

With the optimal permutation $\hat{\sigma}$, we define the soft binary target loss as:

$$\mathcal{L}_{\text{binary}} = \sum_{i}^{n_{\text{pred}}} \mathcal{L}_{\text{KL}}(\mathbf{p}_{t_i}, \mathbf{p}_{s_{\hat{\sigma}(i)}}). \tag{8}$$

It makes the binary query probabilities of the student model similar to the matched ones of the teacher model. And because the preference score $\hat{s}$ (Eq.2) is defined in the same way as the probability p^{pos} (Eq.5), the preference knowledge is distilled from teacher to student as the loss decreases.

Table 2: Comparison of the proposed method to SOTA baselines on the extended COCO-Tasks dataset. The methods with tags † and ‡ are used to calculate mAP margins. TOIST with noun-pronoun distillation achieves best results: +10.9% and +6.6% mAP on detection and segmentation.

Object Detection		Instance Segmentation		
Method	mAP^{box}	Method	mAP^{box}	mAP^{mask}
Faster-RCNN	20.6	Mask-RCNN	23.4	20.0
Faster-RCNN + pick best	14.1	Mask-RCNN + pick best	18.8	16.8
Faster-RCNN + ranker	9.1	Mask-RCNN + ranker	10.6	9.3
Faster-RCNN + classifier	28.8	Mask-RCNN + classifier	33.7	29.2
Faster-RCNN + GGNN	32.6	Mask-RCNN + GGNN‡	37.4	32.4
Yolo + classifier	29.1	**TOIST**	41.3 (+8.1)	35.2 (+2.8)
Yolo + GGNN†	33.2	**TOIST w/ distillation**	**44.1 (+10.9)**	**39.0 (+6.6)**

Summary. The final training loss function for TOIST with noun-pronoun distillation is:

$$\mathcal{L}_{\text{TOIST-NP}} = \mathcal{L}^{t}_{\text{TOIST}} + \mathcal{L}^{s}_{\text{TOIST}} + \lambda_7 \mathcal{L}^{s}_{\text{cluster}} + \lambda_8 \mathcal{L}^{s}_{\text{binary}}, \tag{9}$$

where λ_7 and λ_8 are the weights of losses. $\mathcal{L}^{t}_{\text{TOIST}}$ and $\mathcal{L}^{s}_{\text{TOIST}}$ are separate TOIST loss terms defined by Eq.3 for teacher and student, respectively. Note that the cluster loss $\mathcal{L}^{s}_{\text{cluster}}$ and the soft binary target loss $\mathcal{L}^{s}_{\text{binary}}$ are only used for supervising the student model.

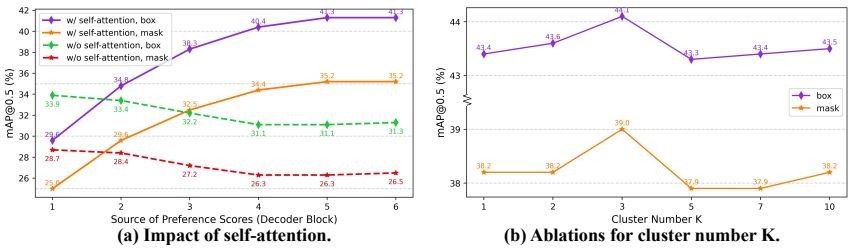
As a reminder, during inference, we only use the student TOIST model and the fixed memory bank to find the most suitable objects, without violating the noun-agnostic constraint.

5 Experiments

Dataset. We conduct experiments on the COCO-Tasks dataset [55] which re-annotates the COCO dataset [40] with preference-aware affordance labels. This is the only dataset that involves instance-level preference in affordance. Though there are other datasets for affordance detection, they neither distinguish between instances nor involve preference, such as ADE-Affordance [12] and IIT-AFF [49]. The COCO-Tasks dataset contains 14 tasks. For each task, there are 3600 train images and 900 test images. In each image, the boxes of preferred objects (one or more) are taken as ground truth labels for detection. Using existing COCO masks, we extend the dataset to an instance segmentation version. In Appendix B and D, we present more dataset details and show its diversity. **Metric.** We use the AP@0.5 metric for both detection and segmentation, where predicted preference scores $\mathbf{S}_{\text{pred}}$ are used to rank objects. Averaging the AP@0.5 values of all tasks leads to mAP@0.5. The implementation details for TOIST and distillation can be found in Appendix A.

5.1 Comparisons with State-of-the-art Methods

Table 2 shows that TOIST with noun-pronoun distillation achieves state-of-the-art results compared to existing methods on COCO-Tasks. For object detection, we use the results reported by [55] as baselines. For instance segmentation, following the same experiment settings of [55], we build new baselines using Mask-RCNN [20]. The methods in the first row treat the problem as a standard detection or segmentation task. All other baselines use two-stage pipelines, in which objects are firstly detected or segmented then ranked. The proposed one-stage method achieves 41.3% mAP^{box} and 35.2% mAP^{mask}, which are +8.1% and +2.8% better than the previous best results (*Yolo+GGNN* and *Mask-RCNN+GGNN*). Noun-pronoun distillation further boosts the performance of TOIST to 44.1% (+10.9%) mAP^{box} and 39.0% (+6.6%) mAP^{mask}. Our method also out-performs another



(a) Impact of self-attention. **(b) Ablations for cluster number K.**

Figure 3: Experiments to study the impact of (a) self-attention and (b) cluster number.

baseline with the same backbone, as shown in Table 9 of Appendix C. These results demonstrate the effectiveness of the proposed method for the new problem of task oriented instance segmentation. Per-task quantitative results and precision-recall curves are provided in Appendix C and D.

5.2 Preference Modeling with Self-Attention

Protocol. Our design principle is that the self-attention layers in transformers can naturally model preference. But is this really the case? Fig.3 (a) shows its effect. Two plain TOIST models are trained separately, with the only difference being that one model does not contain self-attention operators in the decoder. Note that the removal of self-attention does not impact the number of parameters. The mAP metric is impacted by two kinds of errors: inaccurate box/mask localization or improper preference. To analyze the preference scores alone, for all object queries, we use the boxes and masks predicted by the last block of TOIST decoder, which are arguably the most accurate. We use the corresponding preference scores predicted by each block to calculate mAP values.

Interpretation. For the TOIST with self-attention, the performance is gradually boosted as the source of preference scores becomes deeper: from 29.6% mAP^{box} and 25.0% mAP^{mask} to 41.3% and 35.2%. For the one without self-attention, the preference scores from the first block lead to the best performance: 33.9% mAP^{box} and 28.7% mAP^{mask}, which is -7.5% and -6.5% lower than another TOIST. The results demonstrate that the self-attention in TOIST decoder models pair-wise relative preference between object candidates. As the decoder deepens, the preference relationship between object candidates is gradually extracted by self-attention.

5.3 Effect of Clustering Distillation

In Table 3, we show the effects of using cluster loss and replacing pronoun features with cluster centers (noun prototypes). In (c) and (e), leveraging the two components alone brings an increase of +0.7% mAP^{box}, +1.9% mAP^{mask} and +0.7% mAP^{box}, +1.8% mAP^{mask} over baseline (a) respectively. In (g), the complete clustering distillation leads to a performance improvement of +1.0% mAP^{box} and +2.3% mAP^{mask}. These results show that the clustering distillation method can improve student TOIST and enhance verb referring expression understanding.

Table 3: Ablations for distillation settings. CCR, CL and SBTL are short for cluster center replacement, cluster loss and soft binary target loss, respectively.

Index	CCR	CL	SBTL	mAP^{box}	mAP^{mask}
(a)	×	×	×	41.3	35.2
(b)	×	×	✓	43.4 (+2.1)	38.0 (+2.8)
(c)	×	✓	×	42.0 (+0.7)	37.1 (+1.9)
(d)	×	✓	✓	43.8 (+2.5)	38.6 (+3.4)
(e)	✓	×	×	42.0 (+0.7)	37.0 (+1.8)
(f)	✓	×	✓	42.3 (+1.0)	37.3 (+2.1)
(g)	✓	✓	×	42.3 (+1.0)	37.5 (+2.3)
(h)	✓	✓	✓	**44.1 (+2.8)**	**39.0 (+3.8)**

In Fig.4, we visualize the predicted results (filtered by a preference threshold of 0.9) and the attention maps of pronoun tokens. In the first row, when there is no clustering distillation, TOIST wrongly prefers the flower to the cup, which is also confirmed by the attention map. But the TOIST with clustering distillation correctly selects the cup, and the attention on the flower is weakened. This shows that clustering distillation enables the student TOIST to reduce the ambiguity of verb-pronoun referring expression. In the second row, the bounding box of the knife is correctly detected by both two models. However, in the absence of the distillation, extra instance masks are predicted on the spoon and fork within the box. Instead, with the distillation, the masks predicted by TOIST are concentrated on the knife and the attention is more focused on it. This demonstrates that in the case of clustering distillation, TOIST can better ground the task into pixels within an object box.

Meanwhile, the fact that predicted masks may be inaccurate even if the box is correct makes it challenging for a robot to accurately grasp the preferred object when performing a specific task. This proves the importance of extending task oriented object detection to instance segmentation.

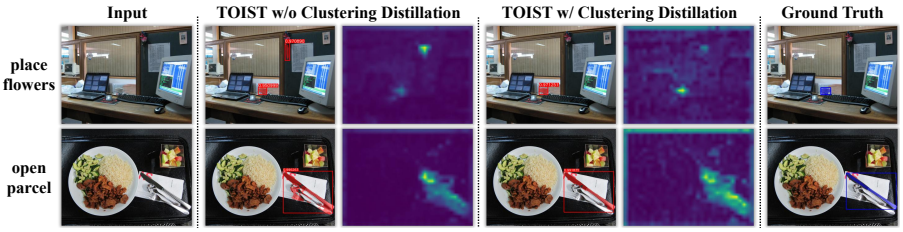


Figure 4: Visualization of the predicted results and attention maps of pronoun tokens.

5.4 Effect of Preference Distillation

In Table 3 (b), preference distillation with soft binary target loss achieves +2.1% $\mathrm{mAP}^{\mathrm{box}}$ and +2.8% $\mathrm{mAP}^{\mathrm{mask}}$ higher results than baseline (a). This loss acts on the preference probabilities of each object candidate in student TOIST. And the probabilities are used as scores to sort the object candidates for the calculation of mAP values. Therefore, the result of Table 3 (b) strongly supports that the preference information is distilled to the student TOIST.

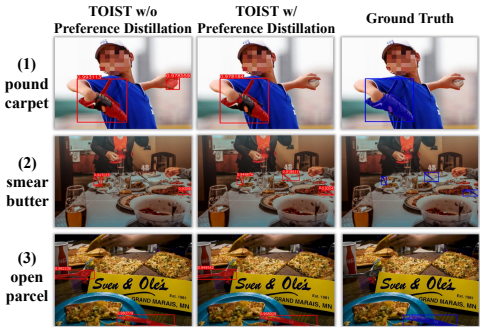


Figure 5: Examples of three scenarios where preference distillation clearly works.

A simple taxonomy differs three scenarios where preference distillation works. As shown in Fig.5, the predicted results (filtered by a preference threshold of 0.9) of the TOIST models w/ or w/o preference distillation are compared. (1) Preference distillation makes the preference score of the false positive object (the baseball in the left picture) lower than the threshold. (2) The preference score of the false negative object (the spoon in the middle) is raised above the threshold with the distillation. (3) When there is no distillation, the false positive object (fork) scores higher than the true positive object (knife) ($0.9822 > 0.9808$). Although the distillation fails to lower the preference score of the false positive object below the threshold, its score is updated to be lower than the true positive one ($0.9495 < 0.9680$). These specific results demonstrate that the information of noun referring expression is distilled to the noun-agnostic student model in the form of preference scores.

5.5 Ablation Study and Qualitative Results

Distillation Methods. Instead of minimizing the distance between $l^{\mathrm{tr}}_{\mathrm{pron}}$ and $l^{j}_{c_s}$, a straightforward way is to directly minimize the distance between $l^{\mathrm{tr}}_{\mathrm{pron}}$ and $l^{\mathrm{tr}}_{\mathrm{noun}}$. As shown in Table 4, this simplified method does not work well, which prompts us to develop the distillation framework.

Table 4: Different distillation methods.

Method	$\mathrm{mAP}^{\mathrm{box}}$	$\mathrm{mAP}^{\mathrm{mask}}$
TOIST	41.3	35.2
distill from $l^{j}_{c_s}$ to $l^{\mathrm{tr}}_{\mathrm{pron}}$	**44.1** (+2.8)	**39.0** (+3.8)
distill from $l^{\mathrm{tr}}_{\mathrm{noun}}$ to $l^{\mathrm{tr}}_{\mathrm{pron}}$	41.9 (+0.6)	36.0 (+0.8)

Table 5: Results without pre-training.

Method	$\mathrm{mAP}^{\mathrm{box}}$	$\mathrm{mAP}^{\mathrm{mask}}$
verb-pronoun input	3.65	5.74
verb-noun input	11.19	12.67
noun-pronoun distillation	7.43 (+3.78)	11.28 (+5.54)

Interaction of the Two Distillation Components. In Table 3 (d) and (f), we show the effects of cluster loss or cluster center replacement together with soft binary target loss. (d) achieves +2.5% $\mathrm{mAP}^{\mathrm{box}}$ and +3.4% $\mathrm{mAP}^{\mathrm{mask}}$ improvement, which demonstrates the two distillation losses collaborate well. (f) only achieves +1.0% $\mathrm{mAP}^{\mathrm{box}}$ and +2.1% $\mathrm{mAP}^{\mathrm{mask}}$ improvement, slightly higher than (e) (using cluster loss only) but lower than (b) (using soft binary target loss only). This shows that preference distillation effectively improves object preference modeling. But solely replacing pronoun features to indicate target objects weakens the effect of preference distillation.

Ablations for Cluster Number K**.** Fig.3 (b) shows the ablations for cluster number K. We perform distillation experiments on different K values between 1 to 10 because increasing K to an even higher value makes the clustering task more difficult. All of the experiments yield better results than the plain TOIST (41.3% $\mathrm{mAP}^{\mathrm{box}}$, 35.2% $\mathrm{mAP}^{\mathrm{mask}}$) and $\mathrm{K} = 3$ works the best. This demonstrates that a modest K can better cluster the information of noun features and distill it to the student TOIST.

Table 6: Ablations for pronoun input.

Method	Pronoun	$\mathrm{mAP}^{\mathrm{box}}$	$\mathrm{mAP}^{\mathrm{mask}}$
TOIST	something	41.3	35.2
	it	41.3	35.2
	them	41.4	35.0
	abcd	39.0	33.2
TOIST w/ distillation	something	44.1	39.0
	it	43.8	38.4
	them	43.8	38.1
	abcd	42.8	37.4

Ablations for Pronoun Input. Table 6 shows the results of TOIST with different pronoun input. In the plain TOIST and TOIST with distillation, the usage of *something*, *it* or *them* leads to similar results,

Table 7: Ablations for the task number n_{task} on task oriented object detection.

Method \ Task	step on something	sit comfortably	place flowers	get potatoes out of fire	water plant
TOIST w/o dis	44.0	39.5	46.7	43.1	53.6
dis n_{task} = 14	46.2 (+2.2)	39.6 (+0.1)	49.9 (+3.2)	47.1 (+4.0)	54.5 (+0.9)
dis n_{task} = 5	46.4 (+2.4)	40.7 (+1.2)	**51.3 (+4.6)**	46.8 (+3.7)	54.6 (+1.0)
dis n_{task} = 1	**47.0 (+3.0)**	**42.1 (+2.6)**	50.8 (+4.1)	**47.4 (+4.3)**	**55.2 (+1.6)**

while a meaningless string *abcd* yields less improvement. Nevertheless, the proposed distillation framework can still work well in the last case, which demonstrates the robustness of our method.

Results without Pre-training. In our architecture, the pre-trained noun referring expression comprehension models are leveraged. To investigate whether the noun-pronoun distillation framework is a standalone technical contribution, we conduct experiments without pre-training. The models are trained from scratch on the COCO-Tasks dataset and the results are shown in Table 5, which demonstrates that the proposed distillation can still improve performance even without pre-training.

Ablations for Task Number. Table 7 shows the ablation study of different task numbers, in which the first row corresponds to the plain TOIST without distillation and the others show the results with distillation under different n_{task}. The results demonstrate our proposed distillation works for different n_{task}, even if n_{task} = 1. And overall, smaller n_{task} leads to better performance. We attribute this to the reduced problem complexity due to the less interaction between different tasks, which makes it easier to improve the ability of the model to understand verbs through noun-pronoun distillation.

Qualitative Results. Fig.6 shows more qualitative results. In (a), two toilets are taken as target objects and annotated partially or totally. But TOIST simultaneously predicts the two kinds of results for each toilet. In (b), no object is annotated, while TOIST keenly detects two water bottles that afford the task. In (c), TOIST predicts more accurate mask result than ground truth. In (d), the table is selected and interestingly it does afford the task as the table edge can be used to open beer bottles. More qualitative results can be found in Appendix E.

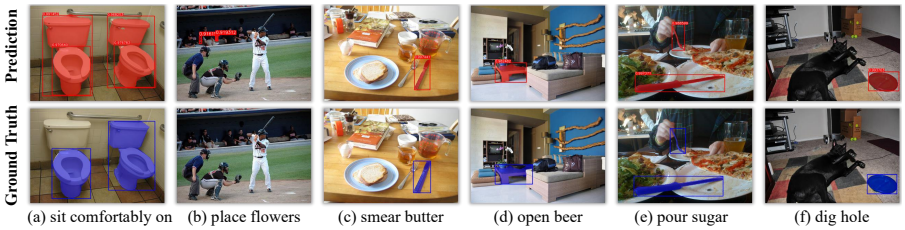


Figure 6: Qualitative results of proposed method for task oriented instance segmentation.

6 Conclusion and Discussion

We explore the problem of task oriented instance segmentation and propose a transformer-based method named as TOIST with a novel noun-pronoun distillation framework. Experiments show our method successfully models affordance and preference, achieving SOTA results on the COCO-Tasks dataset. **Limitations.** Due to the lack of large-scale datasets with more abundant tasks, TOIST is only evaluated on limited tasks. While this is sufficient for many robotics applications, it would be interesting to explore general verb reference understanding on more tasks. **Potential Negative Social Impact.** Because TOIST is not perfect, when it is used in robotics applications, robots may have difficulty in selecting the most suitable object to carry out a task or even cause damage.

References

[1] Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3159–3166, 2019.

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting

visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[4] Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Freitas, David M Blei, and Michael I Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[7] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Conference on Robot Learning*, pages 66–75. PMLR, 2020.

[8] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017.

[9] Xiaoxue Chen, Tianyu Liu, Hao Zhao, Guyue Zhou, and Ya-Qin Zhang. Cerberus transformer: Joint semantic, affordance and attribute parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19649–19658, 2022.

[10] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020.

[11] German KM Cheung, Simon Baker, and Takeo Kanade. Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–375. IEEE, 2003.

[12] Ching-Yao Chuang, Jiaman Li, Antonio Torralba, and Sanja Fidler. Learning to act properly: Predicting and explaining affordances from images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 975–983, 2018.

[13] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5882–5889. IEEE, 2018.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[15] Xuguang Duan, Wenbing Huang, Chuang Gan, Jingdong Wang, Wenwu Zhu, and Junzhou Huang. Weakly supervised dense event captioning in videos. *Advances in Neural Information Processing Systems*, 31, 2018.

[16] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 31, 2018.

[17] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. *Advances in Neural Information Processing Systems*, 30, 2017.

[18] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[19] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3608–3617, 2018.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network.

[23] Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. Learning to solve arithmetic word problems with verb categorization. In *EMNLP*, volume 523533. Citeseer, 2014.

[24] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *European Conference on Computer Vision*, pages 108–124. Springer, 2016.

[25] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

[26] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021.

[27] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[28] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017.

[29] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[30] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2891–2903, 2013.

[31] Svetlana Lazebnik, Edmond Boyer, and Jean Ponce. On computing exact visual hulls of solids bounded by smooth surfaces. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[32] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[33] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020.

[34] Yang Li, Yucheng Tu, Xiaoxue Chen, Hao Zhao, and Guyue Zhou. Distance-aware occlusion detection with focused attention. *IEEE Transactions on Image Processing*, 31:5661–5676, 2022.

[35] Yong-Lu Li, Xinpeng Liu, Xiaoqian Wu, Yizhuo Li, and Cewu Lu. Hoi analysis: Integrating and decomposing human-object interaction. *Advances in Neural Information Processing Systems*, 33:5011–5022, 2020.

[36] Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. Gated graph sequence neural networks. In *Proceedings of ICLR'16*, 2016.

[37] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[38] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[39] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[41] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. In *International Conference on Learning Representations*, 2021.

[42] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[43] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[44] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446, 2020.

[45] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.

[46] Diana McCarthy, Bill Keller, and John A Carroll. Detecting a continuum of compositionality in phrasal verbs. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment*, pages 73–80, 2003.

[47] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016.

[48] Tushar Nagarajan and Kristen Grauman. Learning affordance landscapes for interaction exploration in 3d environments. *Advances in Neural Information Processing Systems*, 33:2005–2015, 2020.

[49] Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5908–5915. IEEE, 2017.

[50] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[51] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[52] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[53] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[54] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[55] Johann Sawatzky, Yaser Souri, Christian Grund, and Jurgen Gall. What object should i use?-task driven object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7605–7614, 2019.

[56] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.

[57] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2019.

[58] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.

[59] Xiaolong Wang, Rohit Girdhar, and Abhinav Gupta. Binge watching: Scaling affordance learning from sitcoms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2596–2605, 2017.

[60] Licheng Yu, Eunbyung Park, Alexander C Berg, and Tamara L Berg. Visual madlibs: Fill in the blank description generation and question answering. In *Proceedings of the ieee international conference on computer vision*, pages 2461–2469, 2015.

[61] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European Conference on Computer Vision*, pages 69–85. Springer, 2016.

[62] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731, 2019.

[63] Aixi Zhang, Yue Liao, Si Liu, Miao Lu, Yongliang Wang, Chen Gao, and Xiaobo Li. Mining the benefits of two-stage and one-stage hoi detection. *Advances in Neural Information Processing Systems*, 34, 2021.

[64] Hao Zhao, Ming Lu, Anbang Yao, Yurong Chen, and Li Zhang. Learning to draw sight lines. *International Journal of Computer Vision*, 128(5):1076–1100, 2020.

[65] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020.

[66] Yixin Zhu, Yibiao Zhao, and Song Chun Zhu. Understanding tools: Task-oriented object modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2855–2864, 2015.

[67] Xinxin Zuo, Chao Du, Sen Wang, Jiangbin Zheng, and Ruigang Yang. Interactive visual hull refinement for specular and transparent object surface reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2237–2245, 2015.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See `https://github.com/AIR-DISCOVER/TOIST`.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix A and B.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix C.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.3.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] (a) MIT License for the COCO-Tasks dataset. (b) Creative Commons Attribution 4.0 License for the Microsoft COCO dataset. (c) Apache License 2.0 for MDETR, DETR and Mask-RCNN implemented by Detectron2.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See Appendix F.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Appendix F.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Appendix F.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Implementation Details

A.1 Method

Noun Features. As mentioned in Section 3, in an input image, it is possible that no objects or multiple objects afford a specific task. And in the latter case, the objects may belong to multiple classes. But for the language input $\mathbf{X}_l$ of verb-noun form, the noun corresponds to the ground truth object categories. Therefore, when the count of targets $n_{\rm gt} = 0$, we use an empty string to construct $\mathbf{X}_l$. When $n_{\rm gt} > 0$ and all target objects belong to the same category, we take a phrase like *sit comfortably on sofa* as $\mathbf{X}_l$. When $n_{\rm gt} > 0$ and the target objects belong to multiple classes, the language input $\mathbf{X}_l$ is set to the concatenation of multiple phrasal verbs, such as *sit comfortably on chair sit comfortably on bed*.

We only update the proposed memory bank in the latter two cases. In these cases, if a noun is encoded into multiple tokenized features by the text encoder, we use the mean value of the features processed by the transformer encoder as $l^{\rm tr}_{\rm noun}$ for updating. In the last case, we take the average of multiple noun features $l^{\rm tr}_{\rm noun-1}, \ldots, l^{\rm tr}_{\rm noun-n_c}$ as the final $l^{\rm tr}_{\rm noun}$, where n_c is the count of classes. In this way, the privileged knowledge of multiple nouns is more easily distilled into the single pronoun feature of the student model when an image contains multiple classes of objects equally suitable for a task.

Components of TOIST. RoBERTa-base [42] and ResNet-101 [21] are used as the text encoder and the CNN-based backbone (the image encoder). For the logit head and the detect head, two feed forward networks of depth one and three are leveraged, respectively. For the segment head, following the network design of [6], a multi-head attention operator and a FPN [38]-like convolutional neural network are used. After each block in the transformer decoder, TOIST generates auxiliary outputs [1] with the prediction heads.

A.2 Loss Functions

We present the loss terms used for the plain TOIST training in details.

For the ground truth objects $\mathbf{O}_{\rm gt}$ of each training sample, we define $\mathbf{p}^{\rm span}_i = [p^{\rm span}_{i,1}, \ldots, p^{\rm span}_{i,n_{\rm max}}] \in [0,1]^{n_{\rm max}}$ as a uniform distribution over the span positions of the text tokens corresponding to the i-th ground truth object. $p^{\rm span}_{i,n_{\rm max}}$ stands for the probability of "no-object", which is 0 for the ground truth objects. As a reminder, we use the whole verb-pronoun (or verb-noun) description as token span. Assuming that $n_{\rm gt}$ (the count of $\mathbf{O}_{\rm gt}$) is smaller that $n_{\rm pred}$ (the count of the predicted objects $\mathbf{O}_{\rm pred}$), we pad $\mathbf{O}_{\rm gt}$ with $\varnothing$ ("no-object") to be of size $n_{\rm pred}$, denoted as $\mathbf{O}'_{\rm gt}$. For $\varnothing$, $\mathbf{p}^{\rm span}_i$ is assigned to be $p^{\rm span}_{i,j} = \mathbb{1}_{\{j=n_{\rm max}\}}$: if $j = n_{\rm max}$, $p^{\rm span}_{i,j} = 1$; otherwise, $p^{\rm span}_{i,j} = 0$.

We denote the bipartite matching between $\mathbf{O}'_{\rm gt}$ and $\mathbf{O}_{\rm pred}$ as $\hat{\sigma}_0$, which is calculated by minimizing the matching loss with the Hungarian algorithm [6]:

$$\hat{\sigma_0} = \underset{\sigma_0 \in \mathfrak{S}_{n_{\rm pred}}}{\arg\min} \sum_{i}^{n_{\rm pred}} \mathbb{1}_{\{p^{\rm span}_{i,n_{\rm max}}=0\}} \left[\mathcal{L}_{\rm l1}(b_i, \hat{b}_{\sigma_0(i)}) + \mathcal{L}_{\rm giou}(b_i, \hat{b}_{\sigma_0(i)}) + \mathcal{L}_{\rm token-m}(\mathbf{p}^{\rm span}_i, \hat{\mathbf{g}}_{\sigma_0(i)}) \right]. \tag{10}$$

Here, $\mathfrak{S}_{n_{\rm pred}}$ is the set of all permutations of $n_{\rm pred}$ elements. b_i and $\hat{b}_{\sigma_0(i)}$ are the ground truth box and the predicted box, respectively. $\hat{\mathbf{g}}_{\sigma_0(i)}$ is the predicted logit, as detailed in the main paper. The loss terms are defined as follows:

$$\mathcal{L}_{\rm l1}(b_i, \hat{b}_{\sigma_0(i)}) = \left\| b_i - \hat{b}_{\sigma_0(i)} \right\|_1. \tag{11}$$

$$\mathcal{L}_{\rm giou}(b_i, \hat{b}_{\sigma_0(i)}) = 1 - \left(\frac{|b_i \cap \hat{b}_{\sigma_0(i)}|}{|b_i \cup \hat{b}_{\sigma_0(i)}|} - \frac{|B(b_i, \hat{b}_{\sigma_0(i)}) \backslash b_i \cup \hat{b}_{\sigma_0(i)}|}{|B(b_i, \hat{b}_{\sigma_0(i)})|} \right). \tag{12}$$

$$\mathcal{L}_{\rm token-m}(\mathbf{p}^{\rm span}_i, \hat{\mathbf{g}}_{\sigma_0(i)}) = -\sum_{j}^{n_{\rm max}} p^{\rm span}_{i,j} \frac{\exp\left(\hat{g}^{\sigma_0(i)}_j\right)}{\sum_{l=1}^{n_{\rm max}} \exp\left(\hat{g}^{\sigma_0(i)}_l\right)}. \tag{13}$$

$\mathcal{L}_{\text{giou}}$ is the Generalized Intersection over Union (GIoU) loss [54]. $|\cdot|$ calculates the size of an area. $B(b_i, \hat{b}_{\sigma_0(i)})$ stands for the smallest box containing b_i and $\hat{b}_{\sigma_0(i)}$. The calculation of $\mathcal{L}_{\text{giou}}$ is implemented by linear functions, so it is differentiable and can be used for back propagation.

For segmentation, Dice/F-1 loss [47] $\mathcal{L}_{\text{dice}}$ and Focal cross-entropy loss [39] $\mathcal{L}_{\text{cross}}$ are leveraged:

$$\mathcal{L}_{\text{dice}}(m_i, \hat{m}_{\sigma_0(i)}) = 1 - \frac{2 m_i \delta(\hat{m}_{\sigma_0(i)}) + 1}{\delta(\hat{m}_{\sigma_0(i)}) + m_i + 1}. \tag{14}$$

Here, m_i is the ground truth instance mask of the i-th object. $\hat{m}_{\sigma_0(i)}$ is the corresponding predicted mask logits. δ is the sigmoid function.

$$\mathcal{L}_{\text{cross}}(m_i, \hat{m}_{\sigma_0(i)}) = -\alpha_t (1-p_t)^{\gamma} \left[m_i \log \delta(\hat{m}_{\sigma_0(i)}) + (1-m_i) \log(1 - \delta(\hat{m}_{\sigma_0(i)})) \right], \tag{15}$$

where

$$\alpha_t = \alpha m_i + (1-\alpha)(1-m_i), \tag{16}$$

$$p_t = m_i \delta(\hat{m}_{\sigma_0(i)}) + (1-m_i)(1 - \delta(\hat{m}_{\sigma_0(i)})), \tag{17}$$

α and γ are hyper-parameters.

The soft-token prediction loss $\mathcal{L}_{\text{token}}$ is defined as:

$$\mathcal{L}_{\text{token}}(\mathbf{p}_i^{\text{span}}, \hat{\mathbf{g}}_{\sigma_0(i)}) = -\sum_{j}^{n_{\max}} p_{i,j}^{\text{span}} \log \frac{\exp\left(\hat{g}_j^{\sigma_0(i)}\right)}{\sum_{l=1}^{n_{\max}} \exp\left(\hat{g}_l^{\sigma_0(i)}\right)}. \tag{18}$$

The contrastive alignment loss is used to align the embedded features of the predicted objects and the corresponding text tokens. The embedded features are obtained by projecting the processed text features of the transformer encoder and the output features of transformer decoder to the same smaller dimension. We follow the definition of [26]:

$$\begin{aligned} \mathcal{L}_{\text{align}} = & \frac{1}{2} \sum_{i}^{n_{\text{pred}}} \frac{1}{|T_i^+|} \sum_{j \in T_i^+} -\log \frac{\exp\left(o_i^\top t_j / \tau\right)}{\sum_{k=1}^{n_{\max}} \exp\left(o_i^\top t_k / \tau\right)} \\ & + \frac{1}{2} \sum_{i}^{n_{\max}} \frac{1}{|O_i^+|} \sum_{j \in O_i^+} -\log \frac{\exp\left(t_i^\top o_j / \tau\right)}{\sum_{k=1}^{n_{\text{pred}}} \exp\left(t_i^\top o_k / \tau\right)}. \end{aligned} \tag{19}$$

T_i^+ is the set of token features that a predicted object feature o_i should be aligned to. O_i^+ is the set of object features to be aligned with a token feature t_i. Here the predicted objects matched to $\varnothing$ are not included. τ is a hyper-parameter.

Finally, the total loss for the plain TOIST can be written as:

$$\begin{aligned} \mathcal{L}_{\text{TOIST}} = & \mathbb{1}_{\{p_{i,n_{\max}}^{\text{span}}=0\}} [\lambda_1 \mathcal{L}_{\text{l1}}(b_i, \hat{b}_{\hat{\sigma_0}(i)}) + \lambda_2 \mathcal{L}_{\text{giou}}(b_i, \hat{b}_{\hat{\sigma_0}(i)})] \\ & + \mathbb{1}_{\{p_{i,n_{\max}}^{\text{span}}=0\}} [\lambda_3 \mathcal{L}_{\text{dice}}(m_i, \hat{m}_{\hat{\sigma_0}(i)}) + \lambda_4 \mathcal{L}_{\text{cross}}(m_i, \hat{m}_{\hat{\sigma_0}(i)})] \\ & + \lambda_5 \mathcal{L}_{\text{token}}(\mathbf{p}_i^{\text{span}}, \hat{\mathbf{g}}_{\hat{\sigma_0}(i)}) \\ & + \lambda_6 \mathcal{L}_{\text{align}}, \end{aligned} \tag{20}$$

A.3 Hyper-parameters and Training Details

For the proposed architecture, we set $d = 256$, $n_{\max} = 256$, $n_{tr} = 6$, $n_{\text{pred}} = 100$, $n_{\text{mem}} = 1024$ and $\text{K} = 3$. For the loss functions, we set $\lambda_1 = 5$, $\lambda_2 = 2$, $\lambda_3 = 1$, $\lambda_4 = 1$, $\lambda_5 = 1$, $\lambda_6 = 1$, $\lambda_7 = 10^4$, $\lambda_8 = 50$, $\alpha = 0.25$, $\gamma = 2$ and $\tau = 0.07$.

During training, we augment input images with random resize and random crop. Specifically, each image is resized such that the shortest side is between 480 and 800 pixels and the longest side is less than 1333 pixels. With probability 0.5, an image is cropped to a random size, where each side is between 384 and 1333 pixels.

We implement TOIST with PyTorch [50]. Both of the student and teacher TOIST models are initialized with the model pre-trained by [26]. We fine-tune the two models on the COCO-Tasks dataset separately for 30 epochs. Then we use the fine-tuned teacher model to distill knowledge to the student model for 15 epochs. We train the models with an AdamW optimizer. The initial learning rates are set to 10^{-5}, 10^{-5}, 5×10^{-5} for text encoder, backbone and transformer, respectively. The weight decay is 10^{-4}. Our experiments were preformed on 8 NVIDIA A100 GPUs.

Table 8: Comparison of different updating strategies for the memory bank in the distillation.

Method	mAP^{box}	mAP^{mask}
TOIST w/o distillation	41.3	35.2
first-in-first-out	43.9 (+2.6)	38.8 (+3.6)
replacing the closest one	**44.1 (+2.8)**	**39.0 (+3.8)**

Table 9: Comparison of the proposed method to 'MDETR+GGNN' baseline on the COCO-Tasks dataset.

Method	mAP^{box}	mAP^{mask}
MDETR + GGNN w/o pretraining	9.6	8.6
MDETR + GGNN	36.8	30.3
TOIST	41.3 (+4.5)	35.2 (+4.9)
TOIST w/ distillation	**44.1 (+7.3)**	**39.0 (+8.7)**

B Dataset Details

We perform experiments on the COCO-Tasks dataset [55] which re-annotates the COCO dataset [40] with preference-aware affordance labels. The COCO-Tasks dataset contains 14 tasks. For each task, there are 3600 train images and 900 test images. We train the proposed architecture on all the train images and evaluate it on all the test images.

In an image, the most suitable objects (one or more) for solving the task are selected and their bounding boxes are taken as ground truth labels for detection. The number of selected objects in an image varies from zero to a dozen. For each task, the total number of selected objects varies between 1,105 and 9,870 and the number of different object categories varies between 6 and 30. Totally, the COCO-Tasks dataset contains 65797 selected objects spanning 49 of the 80 COCO object class categories. This shows the diversity of the dataset. For each task, the total count of all instances belonging to the selected categories largely varies between 7,172 and 34,160. This shows the task oriented object detection problem on this dataset is a non-trivial problem and solving it with traditional methods is very challenging. The preference between multiple classes and multiple instances of the same class must be taken into account.

The COCO-Tasks dataset is annotated with the available COCO detection boxes. Leveraging the corresponding COCO segmentation masks directly gives the upgraded instance segmentation version. We evaluate our proposed method on the upgraded dataset.

C Quantitative results

Strategies for Updating Memory Bank. To update the text feature memory bank in noun-pronoun distillation, two strategies are compared: (1) First-in-first-out. (2) Replacing the feature closest to the new-coming $l_{\text{noun}}^{\text{tr}}$. As shown in Table 8, the second strategy leads to better performance.

Comparison to the Baseline with the Same Backbone. To investigate whether our TOIST architecture is a standalone technical contribution by marginalizing the benefits brought by pre-trained models, we present another baseline 'MDETR+GGNN'. Specially, to leverage the knowledge in noun referring expression comprehension, we use the official pre-trained model of MDETR [26] and then fine-tune it on the COCO-Task dataset. We use the class names of the ground truth objects in each image as the text input to detect these objects. Then we use the GGNN model [55] to infer which objects are preferred for a task. The results are shown in Table 9. Note that this baseline is also tested with privileged noun ground truth, but our distillation method only use the privileged knowledge during training. Nevertheless, our proposed method still has a significant performance improvement over this strong baseline (+7.3% mAP^{box} and +8.7% mAP^{mask}). This demonstrates that our TOIST architecture is a standalone technical contribution towards task oriented instance segmentation and pretraining is necessary but insufficient to get the performance level of TOIST.

Table 10: Ablations for the soft-token prediction loss and the contrastive alignment loss.

Method	mAP^{box}	mAP^{mask}
TOIST	**41.3**	**35.2**
TOIST w/o $\mathcal{L}_{\text{token}}$	40.1 (-1.2)	34.8 (-0.4)
TOIST w/o $\mathcal{L}_{\text{align}}$	41.1 (-0.2)	35.1 (-0.1)
TOIST w/o $\mathcal{L}_{\text{token}}$ and $\mathcal{L}_{\text{align}}$	23.4 (-17.9)	20.7 (-14.5)

Ablations for Loss Terms. To demonstrate the effectiveness of the used loss terms, we provide ablation studies in which we remove $\mathcal{L}_{\text{align}}$ or $\mathcal{L}_{\text{token}}$ or both. The quantitative results are demonstrated in Table 10. It shows that removing $\mathcal{L}_{\text{token}}$ brings a performance drop of -1.2% mAP^{box} and -0.4% mAP^{mask}, because the association between the matched object predictions and the task descriptions is weakened. Removing $\mathcal{L}_{\text{align}}$ brings a performance drop of -0.2% mAP^{box} and -0.1% mAP^{mask}, because the features of an object and its corresponding text features cannot be explicitly constrained to be closer. Interestingly, removing both of them brings a significant performance drop of -17.9% mAP^{box} and -14.5% mAP^{mask}, implying the two loss terms enhance the effect of each other to make TOIST understand verb reference better.

Table 11: Ablations for distillation settings. CCR, CL and SBTL are short for cluster center replacement, cluster loss and soft binary target loss, respectively.

Index	CCR	CL	SBTL	mAP^{box}	mAP^{mask}
(a)	×	×	×	41.3±0.44	35.0±0.28
(b)	×	×	✓	43.2±0.20	38.0±0.02
(c)	×	✓	×	42.0±0.03	37.1±0.04
(d)	×	✓	✓	43.5±0.30	38.6±0.05
(e)	✓	×	×	42.0±0.03	36.9±0.06
(f)	✓	×	✓	42.3±0.02	37.3±0.02
(g)	✓	✓	×	42.3±0.02	37.5±0.03
(h)	✓	✓	✓	**44.1±0.12**	**39.0±0.07**

Table 12: Ablations for pronoun input.

Method	Pronoun	mAP^{box}	mAP^{mask}
TOIST	something	41.3±0.44	35.0±0.28
	it	41.4±0.26	35.1±0.20
	them	41.5±0.28	34.7±0.28
	abcd	38.9±0.10	33.2±0.15
TOIST w/ distillation	something	44.1±0.12	39.0±0.07
	it	43.8±0.12	38.4±0.02
	them	43.7±0.13	38.1±0.03
	abcd	42.8±0.06	37.4±0.02

Replicates. In Table 11 and 12, we show the mean and standard deviation of the results obtained by running experiments three times under different random seeds.

Per-task Results. In Table 13 and 14, we provide per-task results of the proposed method and existing state-of-the-art baselines on the COCO-Tasks dataset. The results show that our TOIST model with noun-pronoun distillation achieves the best performance in most tasks.

Table 13: Per-task object detection results on COCO-Tasks.

Object Detection, AP@0.5															
Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	mean
Faster-RCNN	28.1	25.8	30.1	22.0	30.5	11.7	30.8	0.0	5.1	33.4	9.7	6.1	24.6	30.9	20.6
Faster-RCNN + pick best	22.9	18.1	19.8	15.0	21.3	5.8	20.4	3.9	3.3	22.0	11.1	5.0	12.5	15.6	14.1
Faster-RCNN + ranker	10.7	10.4	11.5	11.6	11.8	3.3	15.0	2.4	4.6	10.5	5.2	5.0	8.3	17.2	9.1
Faster-RCNN + classifier	33.1	26.7	36.8	32.9	35.4	14.6	40.3	14.4	17.6	38.4	17.1	24.5	33.2	38.1	28.8
Faster-RCNN + GGNN	36.6	29.8	40.5	37.6	41.0	17.2	43.6	17.9	21.0	40.6	22.3	28.4	39.1	40.7	32.6
Yolo + GGNN	36.8	31.9	39.1	38.0	41.6	16.5	44.4	18.7	23.0	39.0	22.3	26.9	44.0	42.0	33.2
Mask-RCNN	30.8	25.0	32.7	20.4	33.1	8.0	28.2	7.9	11.2	43.1	8.1	14.7	32.2	32.0	23.4
Mask-RCNN + pick best	21.9	19.6	22.1	21.6	28.3	12.6	26.2	3.4	3.6	29.1	20.4	3.7	22.4	28.6	18.8
Mask-RCNN + ranker	11.6	11.6	12.2	14.0	15.0	5.0	18.9	2.6	3.3	12.2	7.1	4.7	9.2	21.3	10.6
Mask-RCNN + classifier	36.8	31.1	42.4	39.5	40.8	18.6	48.3	13.9	17.2	48.4	21.4	23.1	43.5	46.4	33.7
Mask-RCNN + GGNN	39.0	33.2	46.4	43.8	47.7	21.4	51.2	16.7	20.3	**51.3**	27.3	26.8	50.2	48.9	37.4
MDETR + GGNN	44.3	36.5	45.2	28.6	44.0	**27.6**	35.9	20.7	**34.7**	46.3	27.8	41.5	46.5	36.2	36.8
TOIST	44.0	39.5	46.7	43.1	**53.6**	23.5	52.8	21.3	32.0	46.3	33.1	41.7	48.1	52.9	41.3
TOIST w/ distillation	**45.8**	**40.0**	**49.4**	**49.6**	53.4	26.9	**58.3**	**22.6**	32.5	50.0	**35.5**	**43.7**	**52.8**	**56.2**	**44.1**

Table 14: Per-task instance segmentation results on COCO-Tasks.

Instance Segmentation, AP@0.5															
Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	mean
Mask-RCNN	23.8	21.0	33.0	12.1	33.0	6.8	19.3	6.4	7.7	43.0	6.2	10.9	31.8	24.8	20.0
Mask-RCNN + pick best	17.9	16.4	21.7	16.8	27.9	10.6	23.3	2.9	2.8	28.7	18.6	2.3	21.9	23.4	16.8
Mask-RCNN + ranker	9.0	10.0	11.9	11.2	15.3	4.4	14.4	1.8	2.3	12.0	6.2	3.1	9.1	19.1	9.3
Mask-RCNN + classifier	30.0	27.3	41.4	30.3	39.9	14.8	35.7	12.5	13.5	46.8	19.4	16.0	43.0	38.1	29.2
Mask-RCNN + GGNN	31.8	28.6	45.4	33.7	46.8	16.6	37.8	15.1	15.0	**49.9**	24.9	18.9	49.8	39.7	32.4
MDETR + GGNN	36.9	31.3	43.6	17.1	42.9	20.1	19.9	18.7	24.5	45.5	23.1	30.9	46.2	24.0	30.3
TOIST	37.0	34.4	44.7	34.2	51.3	18.6	40.5	17.1	23.4	43.8	29.3	29.9	46.6	42.4	35.2
TOIST w/ distillation	**40.8**	**36.5**	**48.9**	**37.8**	**53.4**	**22.1**	**44.4**	**20.3**	**26.9**	48.1	**31.8**	**34.8**	**51.5**	**46.3**	**38.8**

D More Analysis

In Fig.7, we present the percentage of ground truth object categories, in each task. The figures show that the distribution of object categories in each task in the COCO-Tasks dataset is very diverse. In Fig.8, we present the percentage of the images that contain at least one ground truth object or contain no object in each task. These figures show that many of the images do not contain objects that are suitable for the tasks. All these statistical results demonstrate that it is non-trivial to find the objects with affordance and preference for a specific task in a given image.

We present the precision-recall curves of our detection and segmentation results on all the test data or the test data that contain at least one ground truth object (termed nonempty data) in Fig.9-12. It can be seen that the curves of the results with noun input are generally higher than those with pronoun input. The proposed noun-pronoun distillation makes the curves of the results with pronoun input closer to those with noun input, which demonstrates the effectiveness of the distillation method. Meanwhile, on the nonempty data, the noun-pronoun distillation only marginally improves TOIST, indicating that the method makes TOIST more capable of filtering out objects that do not afford the tasks in the empty data.

We further present the precision-recall curves of our results on the test data that contain objects of specified classes in each task. The results show that the distillation method has different performance on different categories. For example, in the task *step on something*, it works for the data containing *chair* but does not work for those containing *dining table* or *couch* (see the first row of Fig.13). However, in the task *place flowers* and *get potatoes out of fire*, the proposed distillation method works for almost every category (see the fifth row to the twelfth row of Fig.13). Combined with the statistics of category proportion in Fig.7, we think one possible reason is that the effect of the distillation on different categories is influenced by the proportion of categories in the tasks. When a few classes take a large portion of selected objects in a certain task, the effect of the distillation on these classes is good, while that on others is poor. If the number of categories in the whole task is distributed more evenly, the distillation can boost performance for most categories.

E More Qualitative results

We present more qualitative results in Fig.15. For each task, we select four diverse scenes for visualization. These results show that our method is robust to complex scenes of different tasks.

F Used and Released Asserts

The license of the assets we used is as follows: (a) MIT License for the COCO-Tasks dataset. (b) Creative Commons Attribution 4.0 License for the Microsoft COCO dataset. (c) Apache License 2.0 for MDETR, DETR and Mask-RCNN implemented by Detectron2. All existing codes and dataset we used are open-source and allowed for research. To avoid the disclosure of personally identifiable information and the presentation of the content that might be considered offensive, we have blurred out some of the figures in this paper. Our code is released under the MIT license.

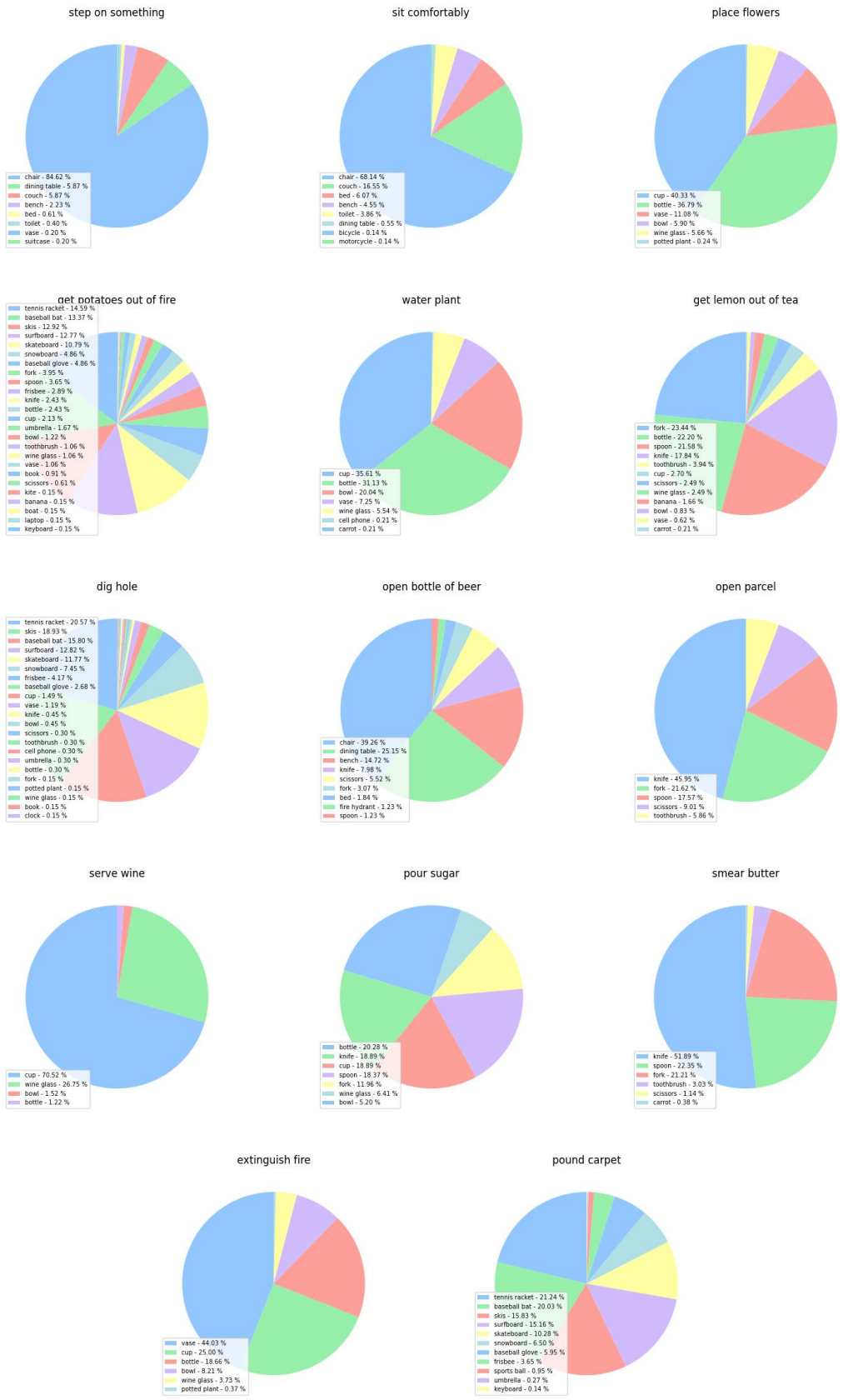


Figure 7: Statistics on the number of images about each ground truth category selected in each task.

Figure 8: Statistics on the number of images that contains selected objects (denoted as 'nonempty') or not (denoted as 'empty') in each task.

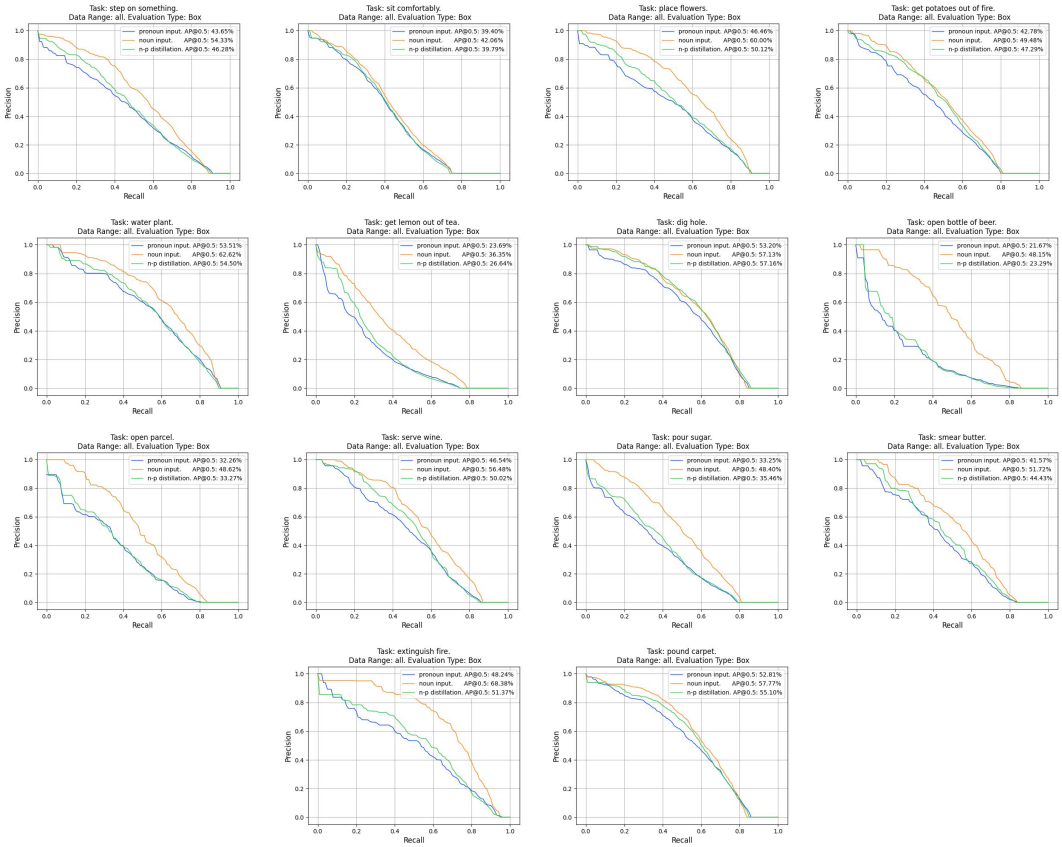


Figure 9: The precision-recall curves for object detection on all the test data in each task. The evaluation type 'Box' means object box instead of object mask.

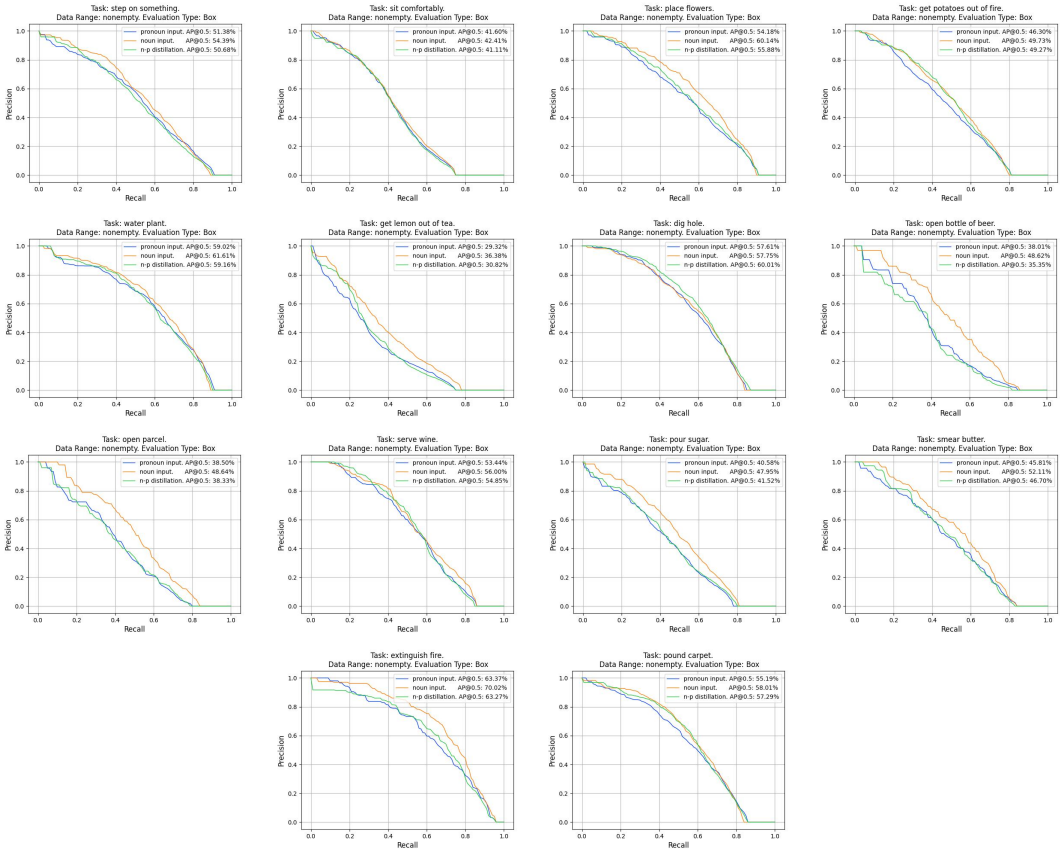


Figure 10: The precision-recall curves for object detection on the test data that contain selected objects in each task. The evaluation type 'Box' means object box instead of object mask.

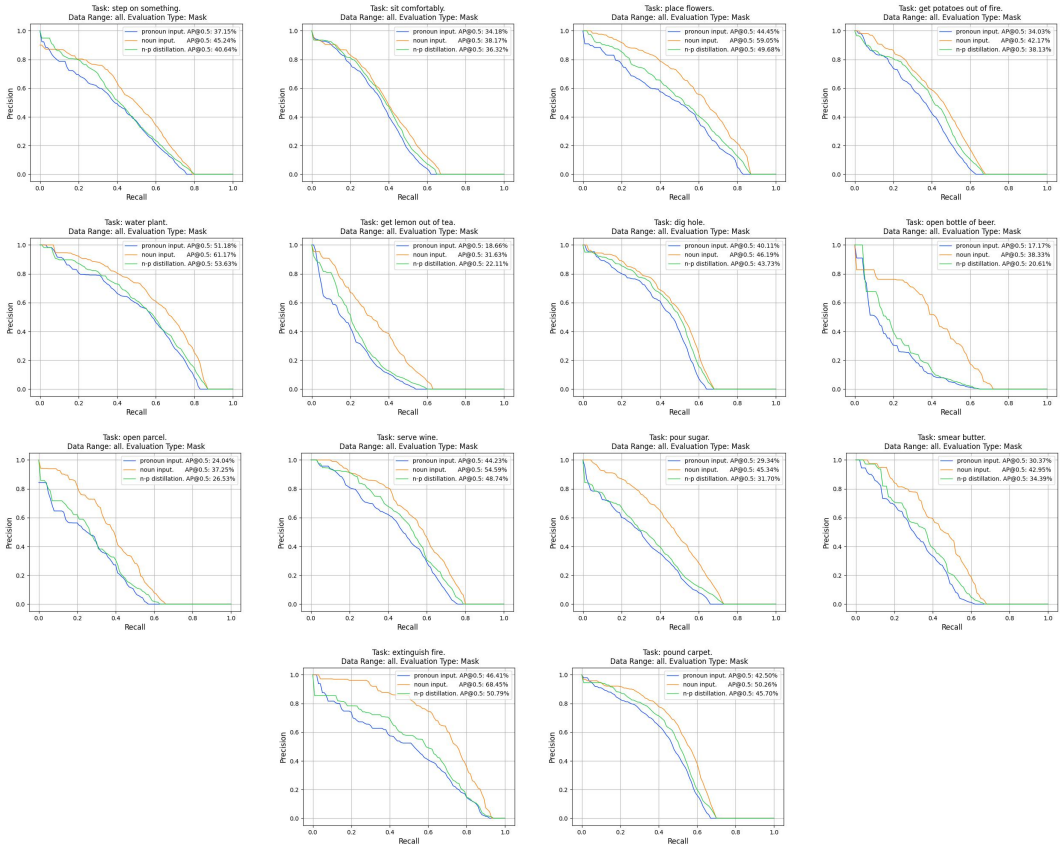


Figure 11: The precision-recall curves for instance segmentation on all the test data in each task. The evaluation type 'Mask' means object mask instead of object box.

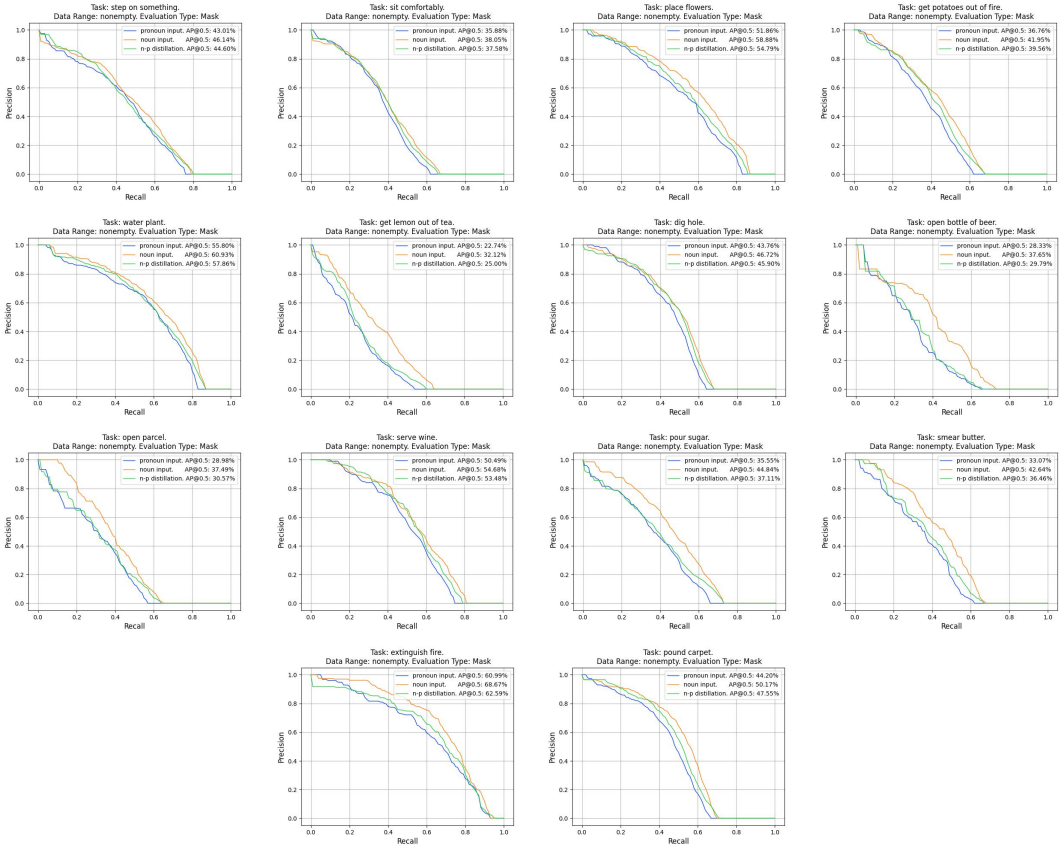


Figure 12: The precision-recall curves for instance segmentation on the test data that contain selected objects in each task. The evaluation type 'Mask' means object mask instead of object box.

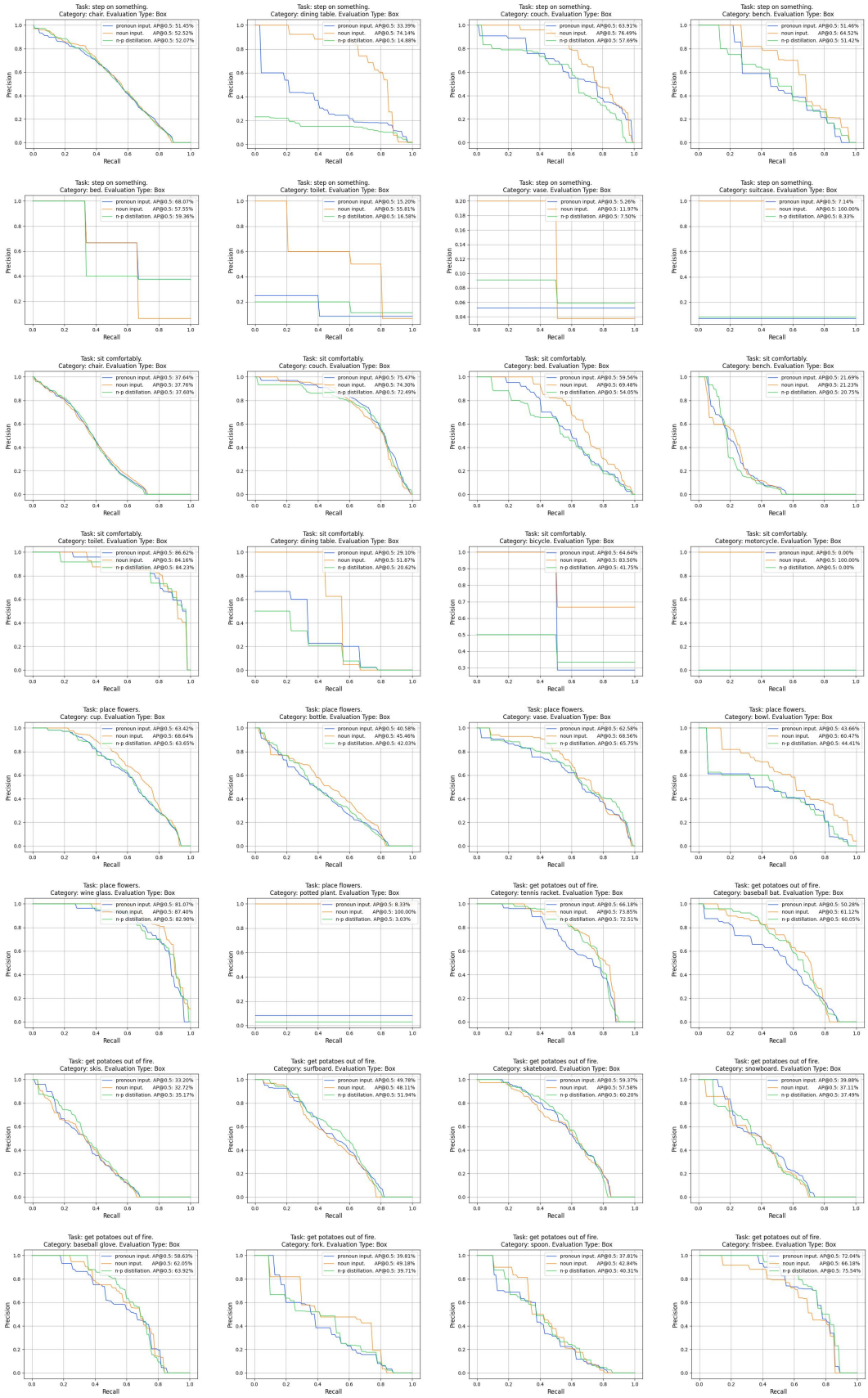


Figure 13: The precision-recall curves for object detection on the test data that contain objects of specified classes in each task.

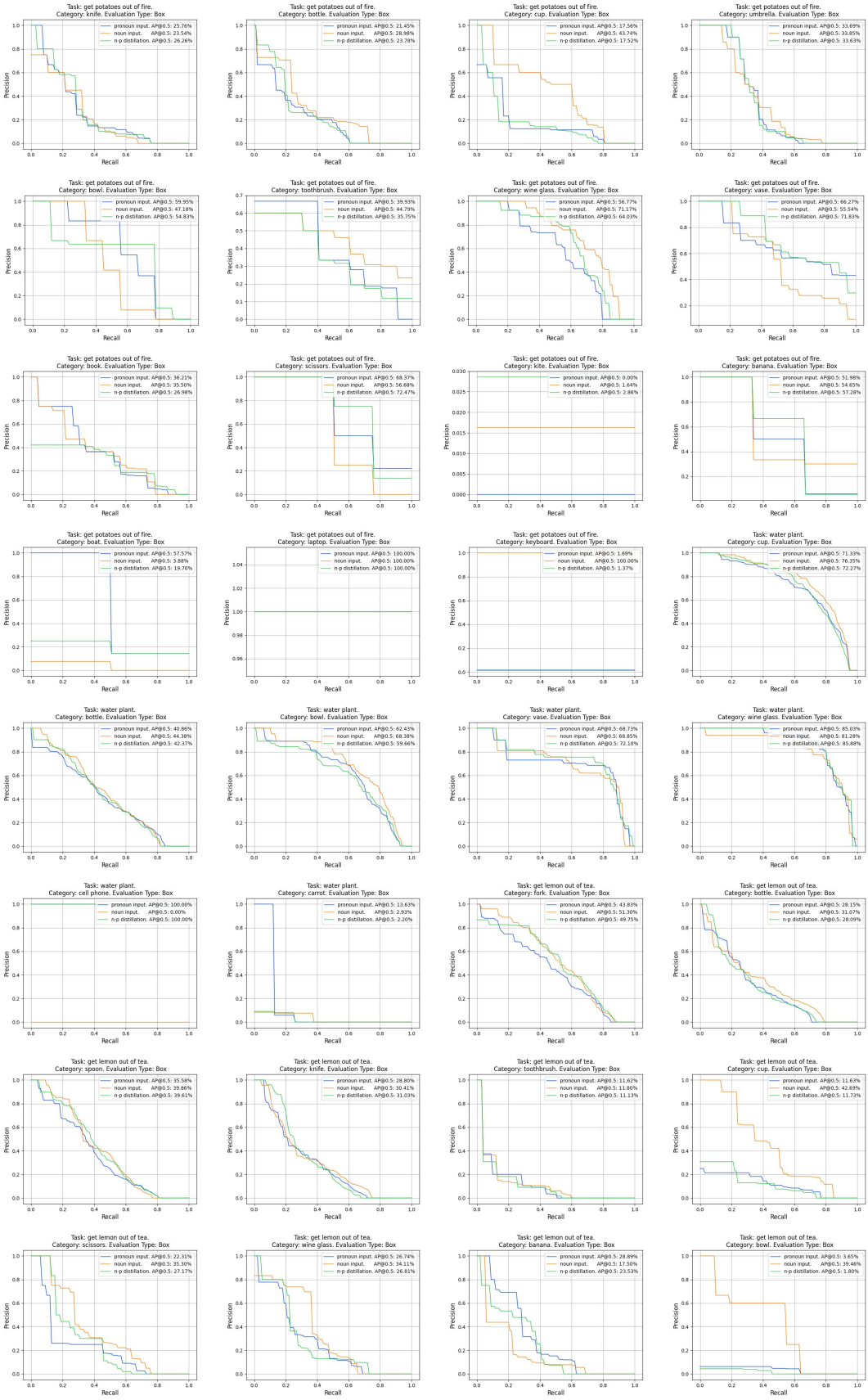


Figure 13: The precision-recall curves for object detection on the test data that contain objects of specified classes in each task (cont.).

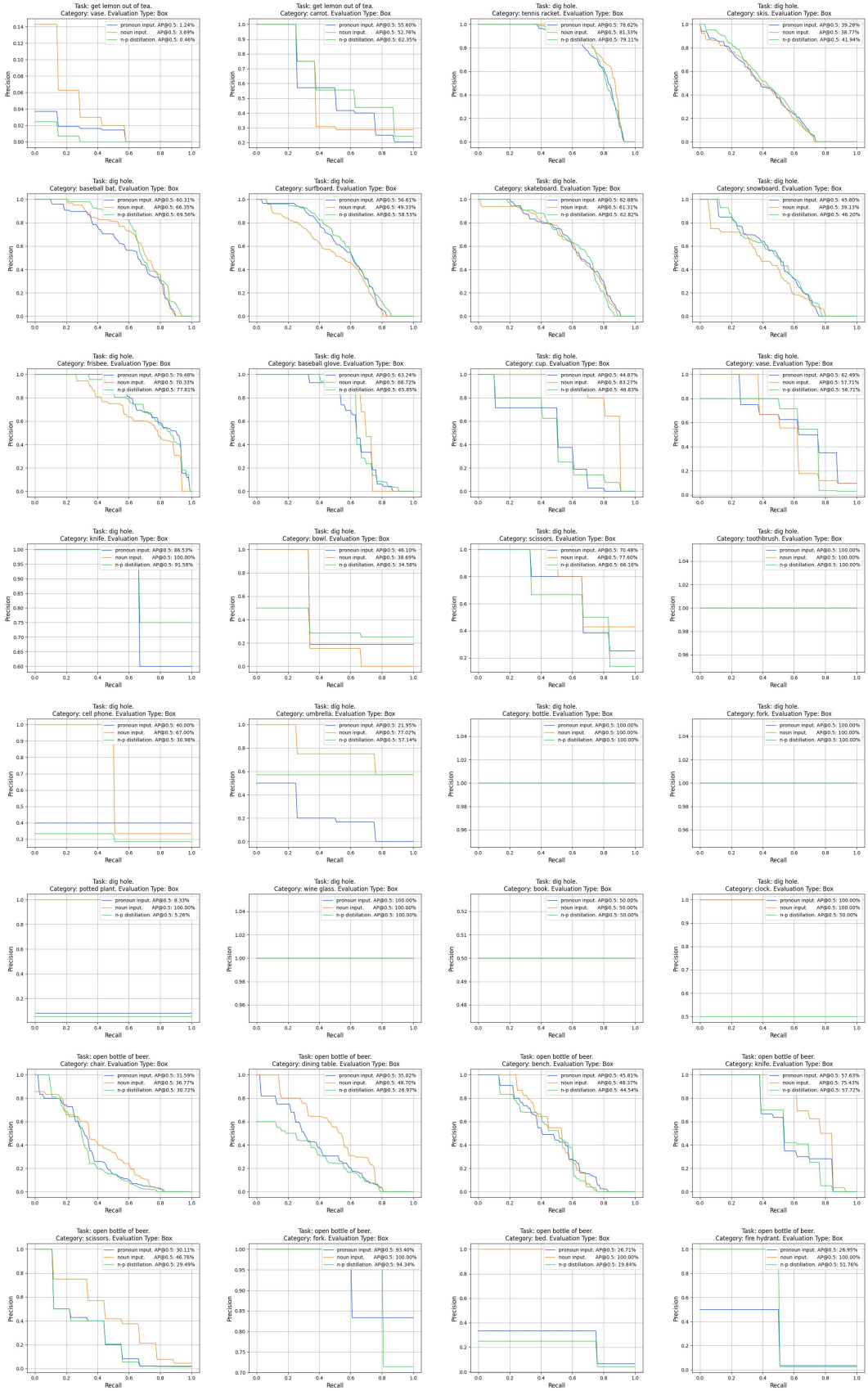


Figure 13: The precision-recall curves for object detection on the test data that contain objects of specified classes in each task (cont.).

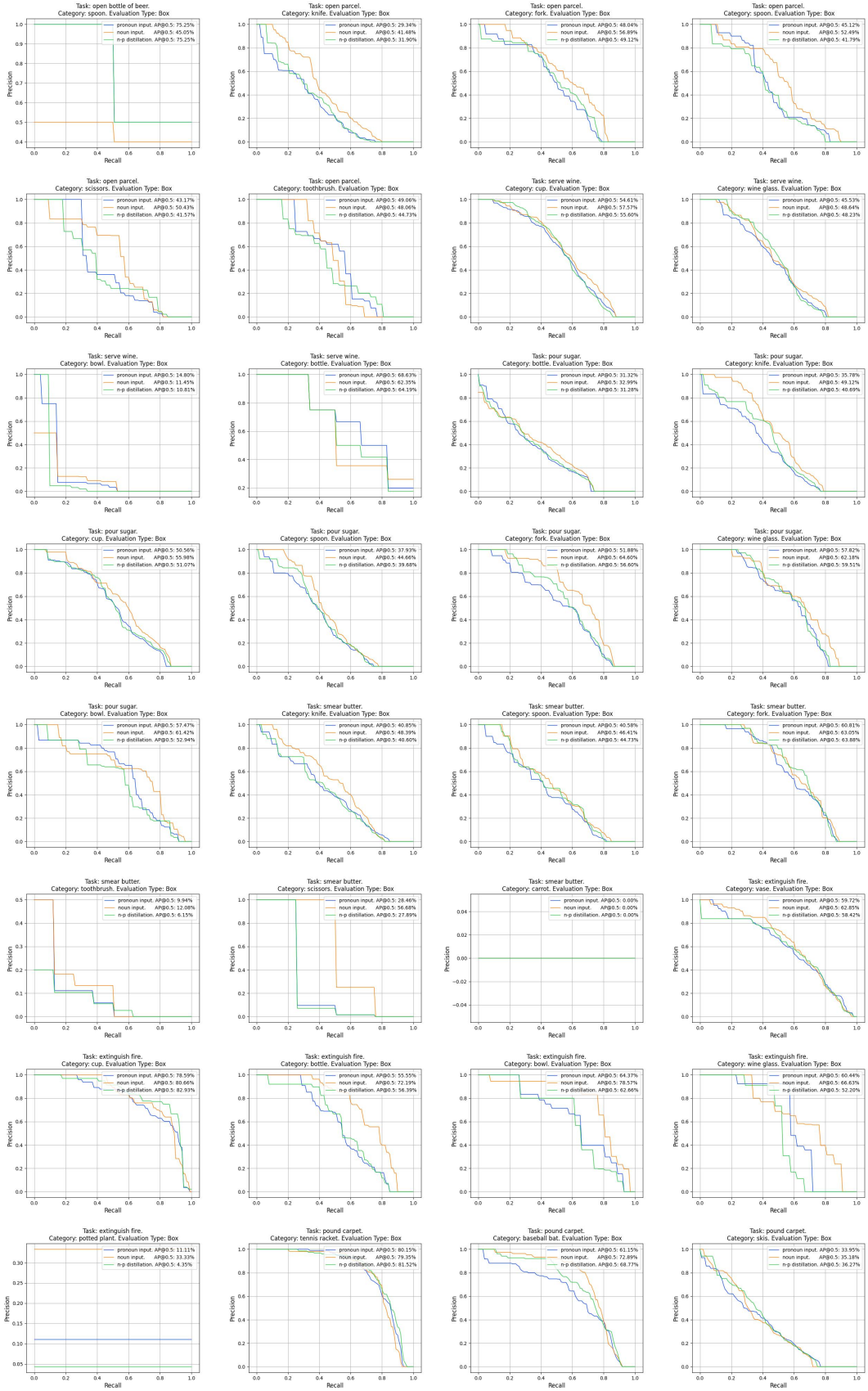


Figure 13: The precision-recall curves for object detection on the test data that contain objects of specified classes in each task (cont.).

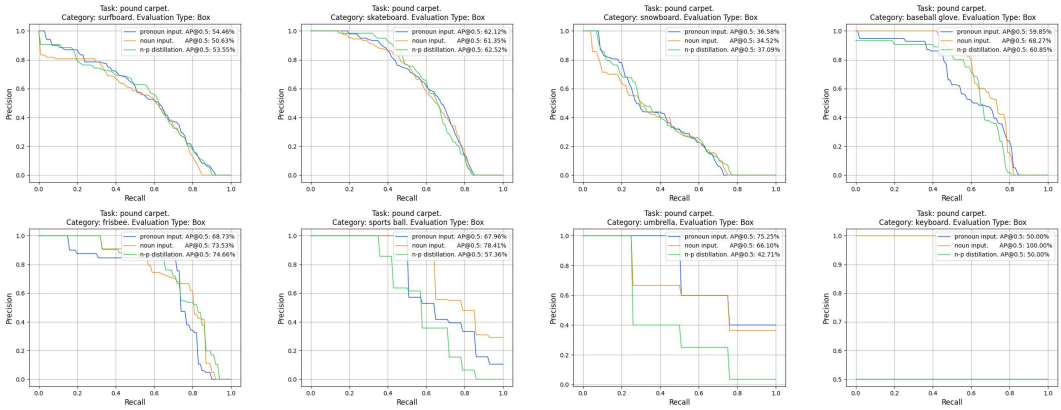


Figure 13: The precision-recall curves for object detection on the test data that contain objects of specified classes in each task (cont.).

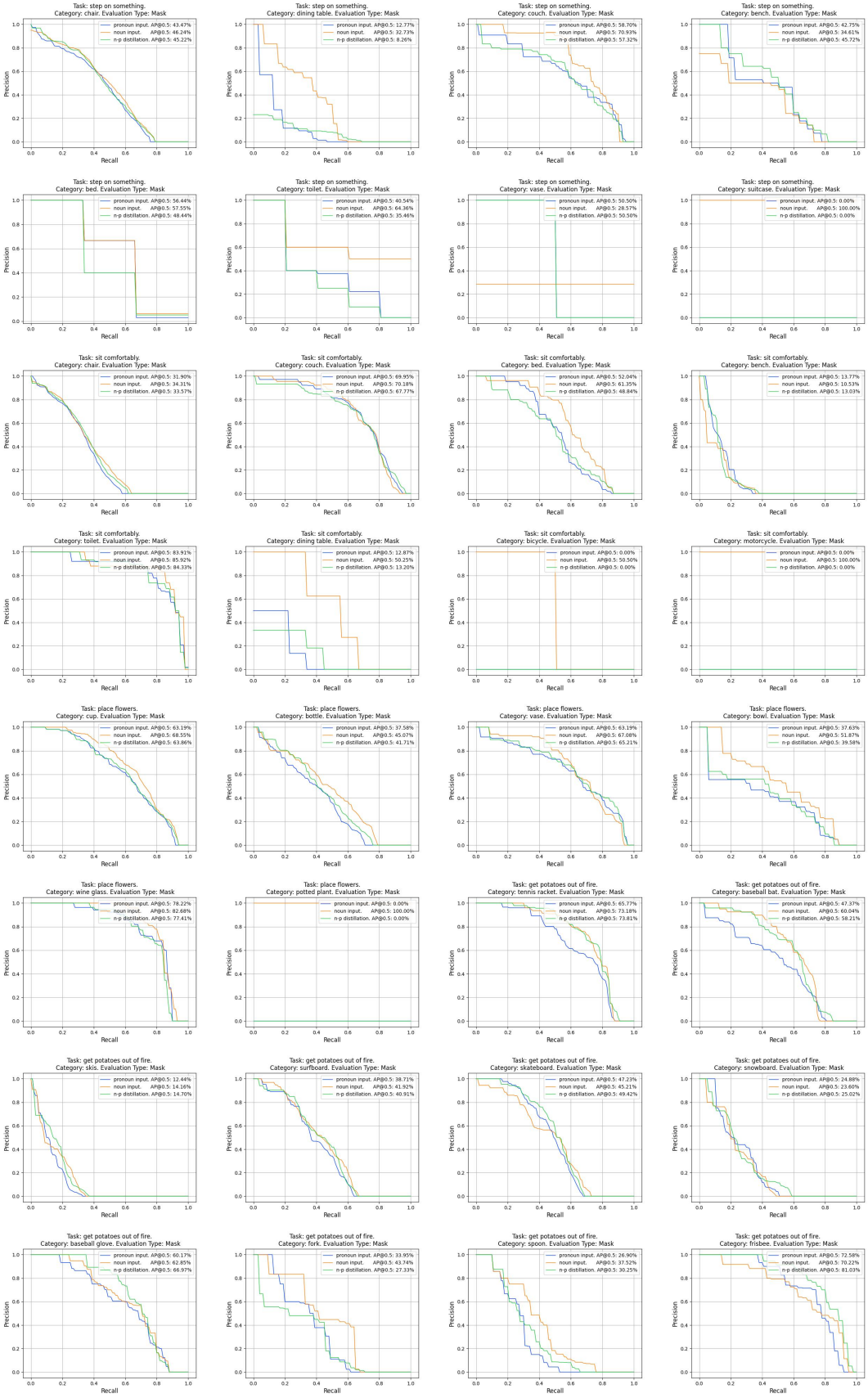


Figure 14: The precision-recall curves for instance segmentation on the test data that contain objects of specified classes in each task.

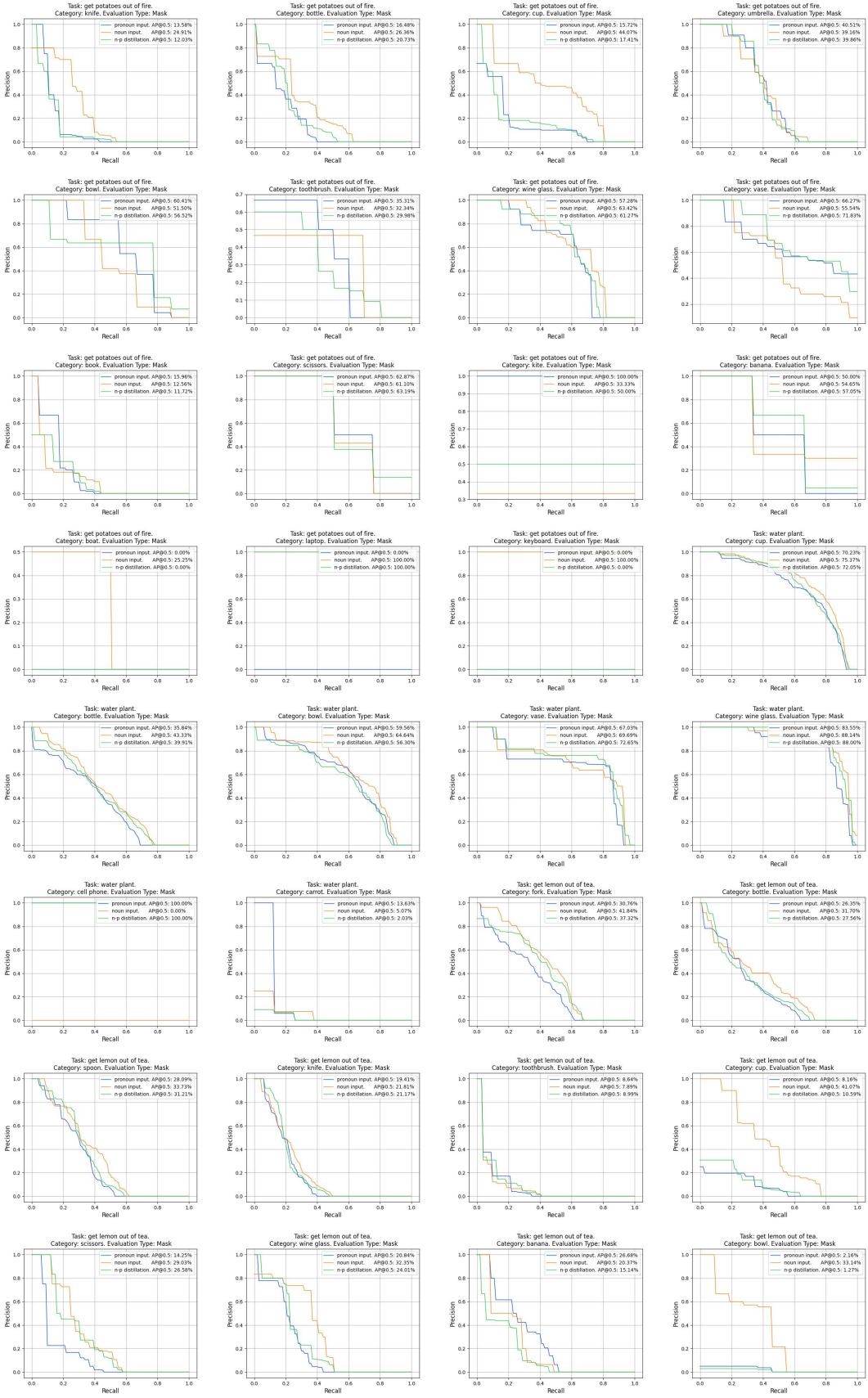


Figure 14: The precision-recall curves for instance segmentation on the test data that contain objects of specified classes in each task (cont.).

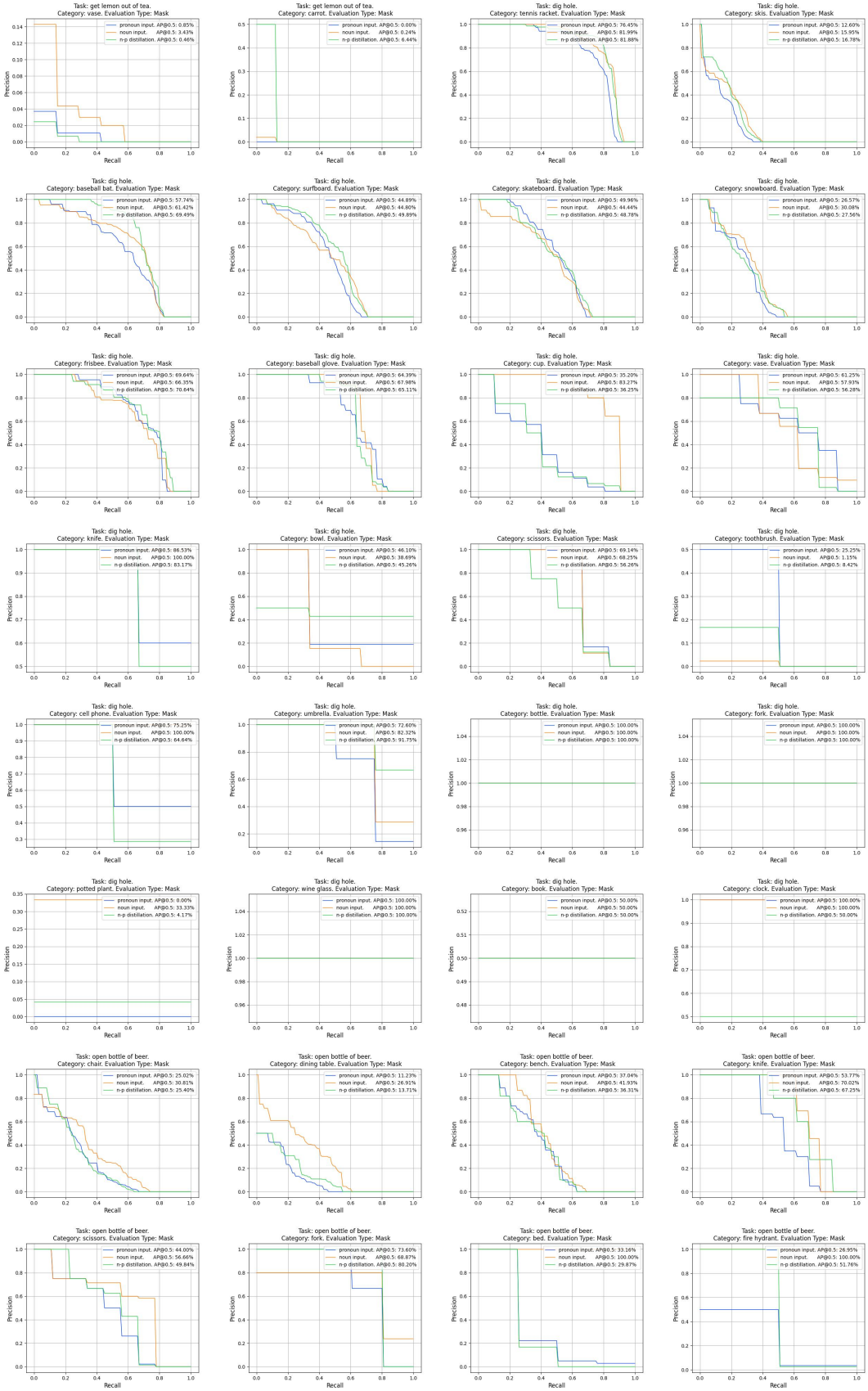


Figure 14: The precision-recall curves for instance segmentation on the test data that contain objects of specified classes in each task (cont.).

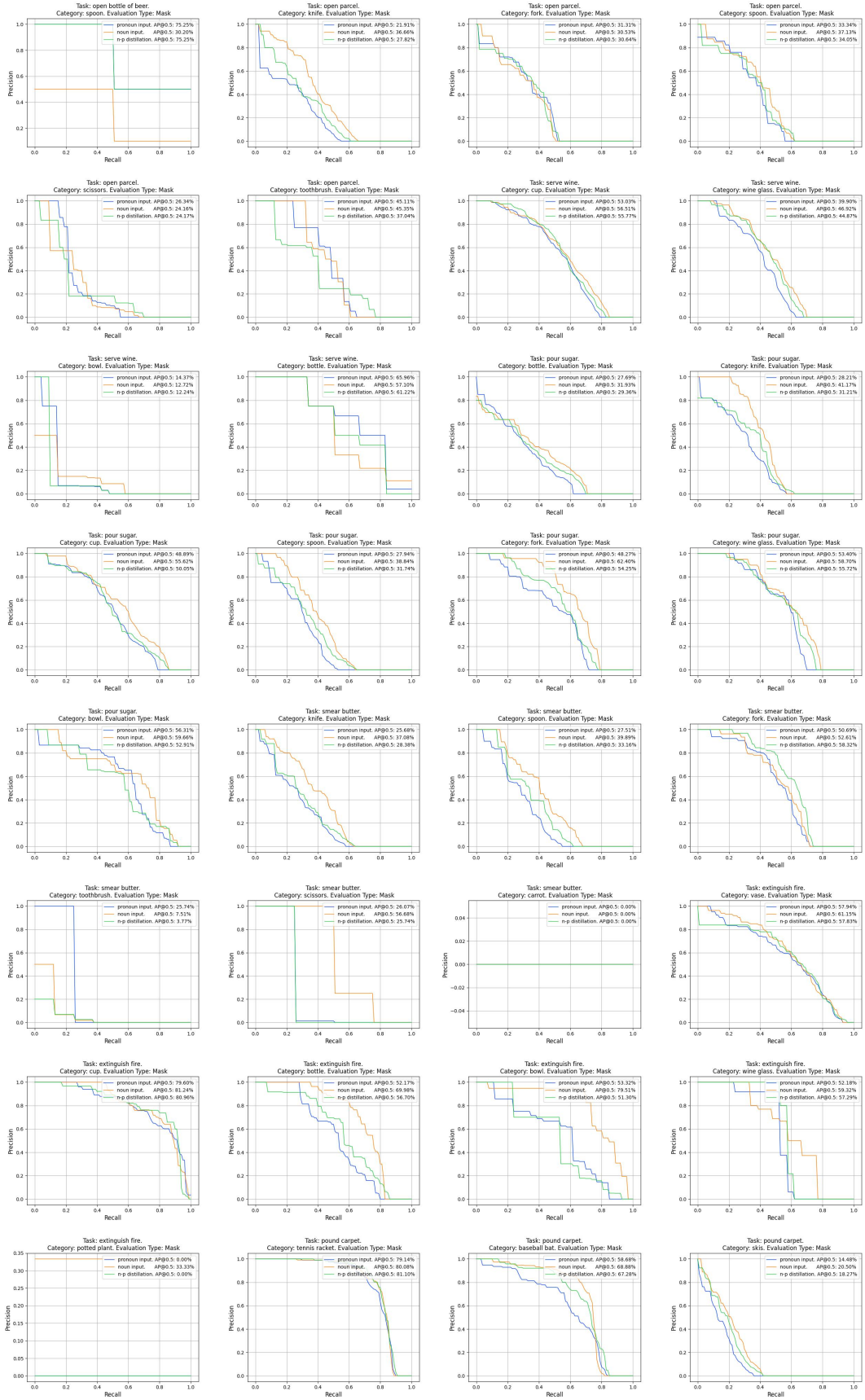


Figure 14: The precision-recall curves for instance segmentation on the test data that contain objects of specified classes in each task (cont.).

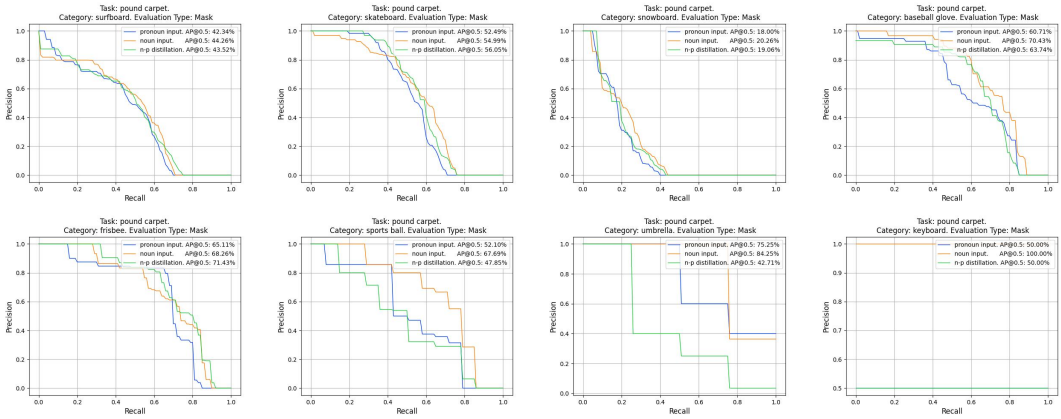


Figure 14: The precision-recall curves for instance segmentation on the test data that contain objects of specified classes in each task (cont.).

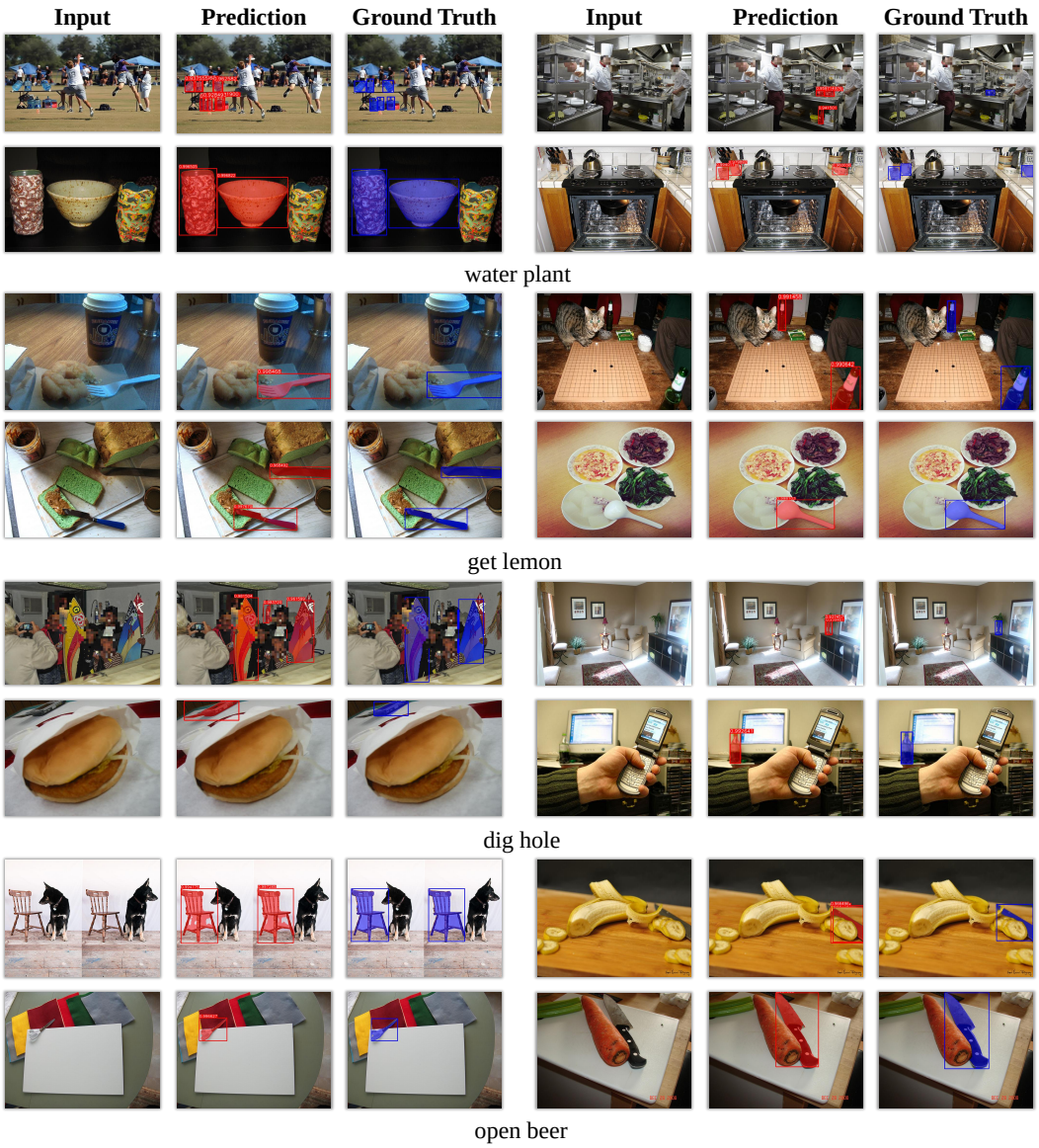


Figure 15: More qualitative results.

Figure 15: More qualitative results (cont.).

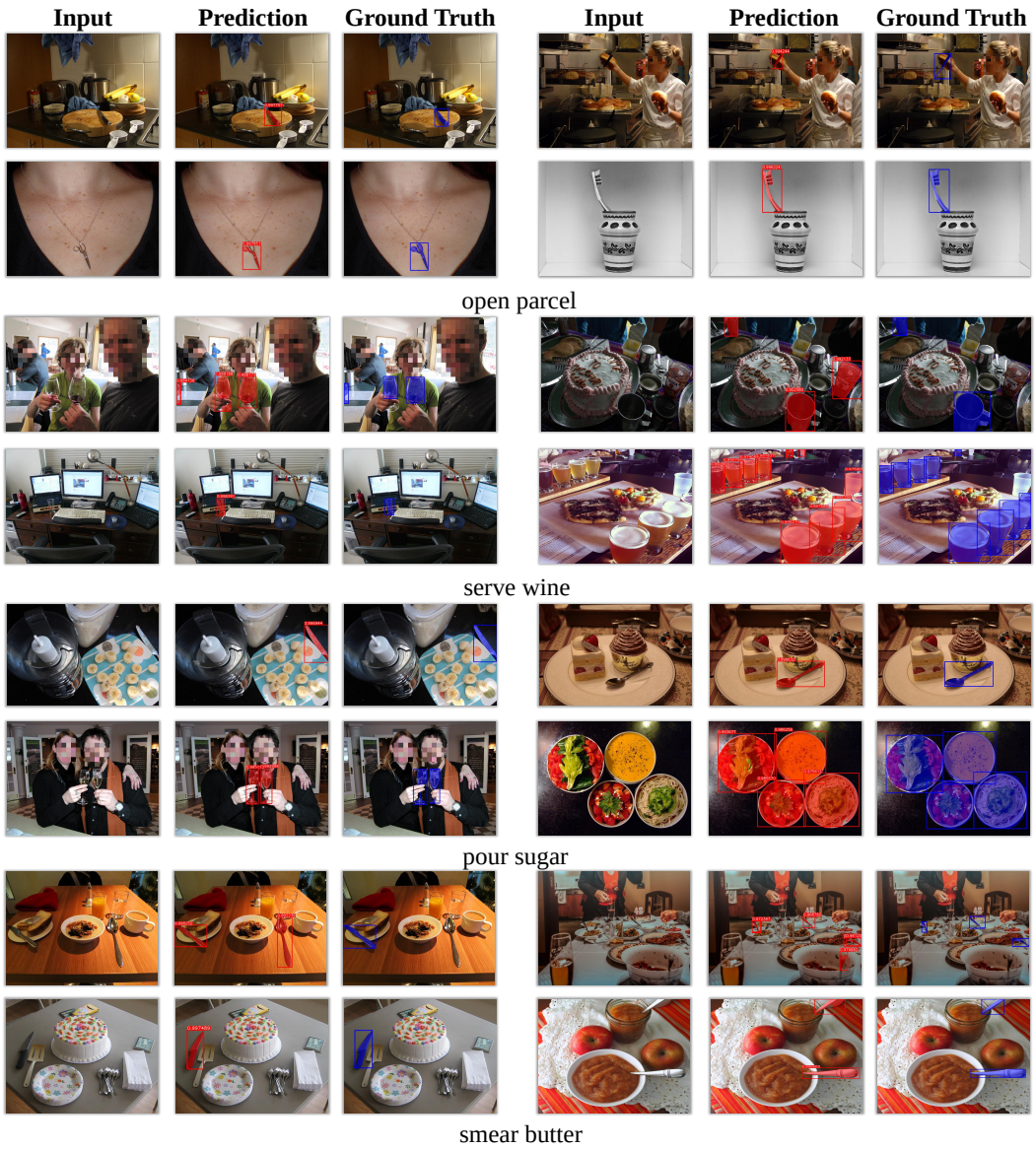


Figure 15: More qualitative results (cont.).

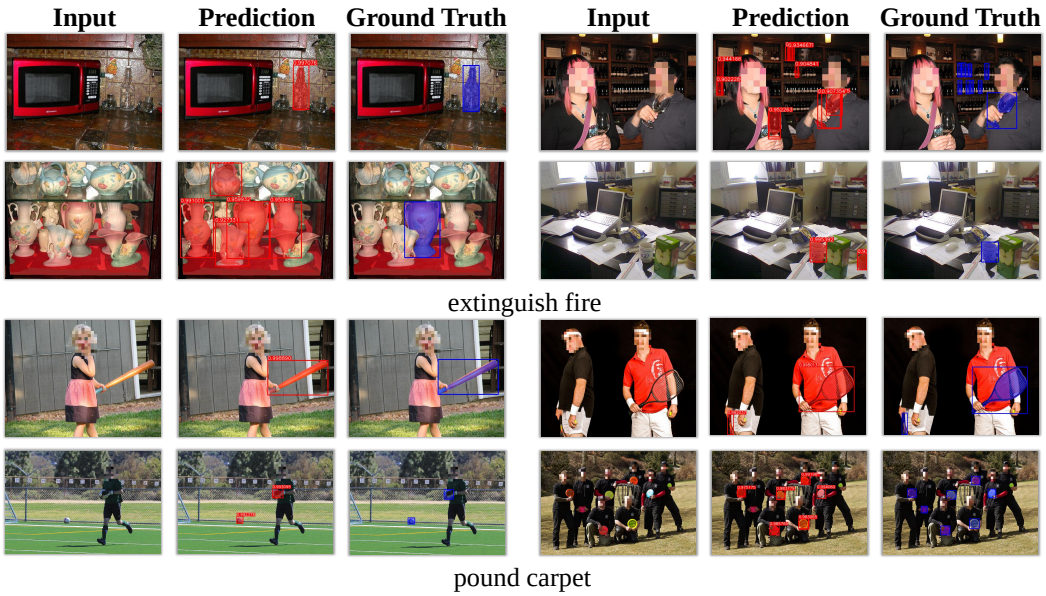


Figure 15: More qualitative results (cont.).