Vision-Language Grounding as Bidirectional Concept Correspondence

Jieyu Zhang[1]* **Ziqi Gao**[1,2]* **Luke Zettlemoyer**[1,3] **Ranjay Krishna**[1]

[1]University of Washington
[2]Allen Institute for AI
[3]FAIR at Meta

Project Page: `https://uwgzq.github.io/papers/ConCor-1`

Abstract

Vision-language grounding connects language to visual content, yet most existing formulations reduce grounding to a unidirectional localization problem; given a prespecified text phrase or category name, identify the corresponding image region. This setup assumes that the relevant linguistic unit is already known, overlooking a more basic challenge in grounded communication: determining which parts of the text are visually referential and how they correspond to entities in the image. We formulate grounding as *bidirectional concept correspondence* over an image-text pair. Given an image and its paired text, the goal is to recover all correspondences between visually referential text spans and instance-level image segments, without assuming that the relevant text spans are provided. This formulation unifies common grounding tasks, including phrase grounding, referring expression grounding, and open-vocabulary detection by treating text segmentation, image segmentation, and cross-modal alignment as a single correspondence prediction problem. To address this task, we introduce **ConCor-1**, a grounding model built on top of a pretrained vision-language model. It uses learnable *bridge tokens* to represent candidate image-text correspondences and predicts, for each token, a text mask, an image mask, and a correspondence presence score. To train and evaluate this task, we convert diverse grounding and segmentation datasets into a unified correspondence format. Experiments show that **ConCor-1** consistently outperforms baselines, improving correspondence F1 by 48% on the long-caption dataset and by 29% on zero-shot LVIS, where the large category list serves as the text input.

1 Introduction

Grounding is a long-standing problem in vision-language learning that asks how linguistic concepts correspond to visual content in an image. Grounding enables multimodal systems to produce outputs that are spatially attributable, making their predictions more interpretable and actionable [1, 2, 3].

Most existing formulations, however, simplify grounding into a unidirectional language-to-image localization task. Early work studied referring expressions [4, 5, 6, 7], where a short phrase such as "squirrel" identifies a target object, as in ReferItGame [8]. Later work expanded grounding to phrase grounding [9, 10], where referential phrases in a caption, such as "a brown dog" or "the wooden bench" are aligned to image regions, and further to open-vocabulary detection and segmentation [11, 12]. These tasks typically assume that the relevant linguistic unit is already specified. Text indicates what to find, and the image provides the region to retrieve.

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

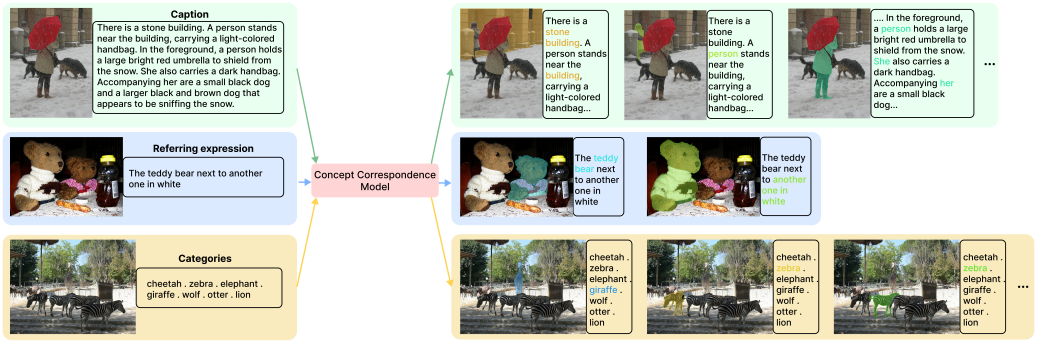


Figure 1: **Bidirectional concept correspondence.** Given an image paired with different forms of text, including a full caption, a referring expression, or a list of category names, the concept correspondence model predicts a complete set of correspondences between text segments and image segments. For each correspondence, the model identifies the visually grounded text span, including repeated or coreferential mentions, and localizes the corresponding instance-level mask in the image.

Grounding is not simply a one-way retrieval process. In human communication, reference is a joint activity between speaker and listener, where utterances are interpreted against shared perceptual context and common ground [13, 14]. Listeners are not handed a pre-segmented list of phrases to localize; rather, they must infer which parts of an utterance are referential, which entities in the scene are relevant, and how the two should be aligned. Likewise, cognitive accounts of spatial language show that people construct structured spatial mental models from descriptions, rather than treating words as isolated category labels [15, 16]. Visual cognition also requires selective attention [17] and object binding: image regions must be grouped into a coherent structure before they can serve as referents for language [18]. Together, these perspectives suggest that visual grounding is more naturally viewed as a problem of recovering correspondences between structured linguistic and visual representations [19].

Motivated by this perspective, we formulate vision-language grounding as *bidirectional concept correspondence* over an image-text pair. In other words, grounding determines both *what in the text is visually referential* and *what in the image it refers to*. Given an image and its paired text, the goal is to recover the complete set of correspondence pairs $\{(t_i, s_i)\}$, where s_i is an instance-level image mask and t_i is a binary mask over text marking all mentions that refer to the same visual entity. The model must jointly identify grounded text segments, segment the corresponding visual entities, and determine whether each proposed text-image pairing is valid. Moreover, t_i may contain multiple non-contiguous spans, allowing the model to group repeated or coreferential mentions.

This formulation subsumes several existing grounding tasks as special cases. Phrase grounding arises when the relevant text spans are already specified in the paired caption. Open-vocabulary detection and segmentation correspond to the restricted case where the text consists only of category names. Referring expression grounding corresponds to the case where a query expression is provided, but our formulation additionally requires the model to ground contextual entities mentioned in the expression. More broadly, bidirectional concept correspondence connects vision-language grounding to visual coreference resolution [20] and Winograd-style referential ambiguity [21, 22, 23], where resolving expressions such as "it" in "The trophy does not fit into the suitcase because it is too large" depends on linking language to the visual or physical situation.

To address this task, we introduce **ConCor-1**, a correspondence prediction model built on top of a pretrained vision-language model (we use Qwen3.5 [24] as our backbone). This design is motivated by two requirements of bidirectional correspondence. First, recovering all grounded entities in an image-text pair requires reasoning over long and compositional text, including modifiers, relations, repeated mentions, and discourse context. Second, because text segmentation is part of the output space, the model must identify visually grounded spans, resolve coreference, and interpret contextual references. Large pretrained vision-language models provide a natural backbone for these capabilities because they jointly process image and text while preserving a flexible language interface.

Built on this shared backbone, **ConCor-1** adds three lightweight prediction heads: one for image segmentation, one for text segmentation, and one for correspondence presence prediction. To represent individual correspondences, we introduce a set of learnable *bridge tokens* that act as queries

for potential correspondences. We append these bridge tokens to the multimodal token sequence, allowing them to be contextualized through self-attention over both image tokens and text tokens in the pretrained backbone. After interacting with both vision and text tokens, the image segment and text mask are predicted by measuring similarity between the bridge token and the vision or text tokens, respectively, while the presence head predicts whether the correspondence is valid. This design keeps the pretrained backbone largely intact while adding a minimal task-specific interface for predicting bidirectional correspondences.

We train and evaluate **ConCor-1** by converting diverse grounding, detection, and segmentation datasets into a unified correspondence format. Experiments show that **ConCor-1** consistently performs best at predicting concept correspondences across two evaluation settings: image-caption grounding, where the model must identify visually referential spans in natural captions, and image-category grounding, where the input text is a large list of candidate categories. The gains are especially substantial on COCONut-PanCap, which contains long, complex captions with dense object references: compared with the strongest baseline, **ConCor-1** improves correspondence F1 (JointF1) by 48% and correspondence IoU (mJS) by 41%. In the zero-shot, large-vocabulary LVIS setting, **ConCor-1** is able to process the full category vocabulary in a single forward pass and outperforms the strongest baseline by 29% in JointF1. These results show that explicit bridge-token correspondence prediction enables a unified model to recover text spans, image masks, and their alignments across both dense captions and large-vocabulary category queries.

2 Related work

Vision-language grounding. Vision-language grounding has been studied under several formulations that differ mainly in how the linguistic query is specified and what visual output is required [3, 25, 26, 27, 28]. Early work focused on referring expression comprehension in natural images, where a short phrase is used to localize a target object or region [8, 4, 5, 6]. Later, phrase grounding extended this setup to align multiple phrases in a caption with image regions [9, 10], while more recent open-vocabulary detection and segmentation methods allow arbitrary category names or free-form text queries to localize boxes or masks [29, 30, 11, 31, 32]. Despite these differences, most prior formulations remain fundamentally *query-driven*: text is treated as an input specification for retrieval, and the objective is to localize the corresponding visual region. In contrast, we study grounding as a *bidirectional concept correspondence* problem over an image-text pair, where the model must recover the full set of aligned text-image concepts without assuming that the grounded text spans are given in advance.

Grounding models. Prior work on grounding and segmentation includes detector-style methods such as MDETR [33], DetCLIP [34, 35], GLIP [29, 30], Grounding DINO [11, 36, 37], and GLEE [38]; generalized and promptable segmentation models such as X-Decoder [39], OpenSeeD [40], SEEM [41], SAM3 [12], and OpenWorldSAM [42]; and autoregressive VLM-based systems including Florence-2 [43], Kosmos-2 [44], Molmo [1, 2], LISA [45, 46], and GLaMM [47]. In contrast, our model combines extensively pretrained VLM backbones and segmentation-based grounding: it builds on a pretrained VLM for rich joint image-text understanding, while adding lightweight segmentation-oriented heads to explicitly predict text spans, image segments, and their correspondences in a non-autoregressive and structured manner.

3 Bidirectional concept correspondence

We start by defining vision-language grounding as a bidirectional correspondence problem between language and vision. Formally, given an image I and a text sequence T, the output is a set of correspondence pairs $\mathcal{C} = \{(t_i, s_i)\}_{i=1}^{K}$, where each t_i denotes a grounded text segment in T and each s_i denotes the corresponding image segment in I. Each t_i is not necessarily restricted to a single contiguous character span; it can be represented as a binary mask over characters in the text, allowing the formulation to capture discontinuous or multi-span grounded expressions.

By requiring the model to recover the complete set of concrete correspondences in an image–text pair, the model must first interpret the text to resolve coreference and identify entity mentions, then examine the image to surface candidate objects, and finally determine which mentions and objects correspond—while recognizing that some mentions may have no visual referent and some objects

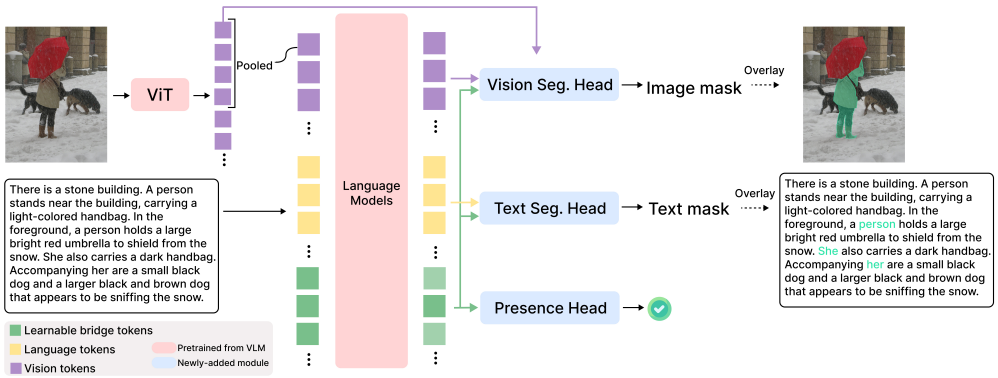


Figure 2: **Model overview. ConCor-1** builds on a pretrained vision-language backbone and introduces learnable bridge tokens to represent candidate text–image correspondences. Given an image and paired text, vision tokens, language tokens, and bridge tokens are jointly processed by the language model. For each bridge token, the model predicts an image mask with a vision segmentation head, a text mask with a text segmentation head, and a presence score indicating whether the token corresponds to a valid grounded concept.

may not be described in the text. This richer structure also enables applications beyond standard grounding benchmarks, including evaluating the groundedness of image captions and the faithfulness of text-to-image generations.

This new task poses key technical challenges. The model must handle long, compositional text with complex referring structures, including modifiers, repeated mentions, and pronouns. It must also produce fine-grained instance image masks with local spatial detail. Finally, a single text–pixel similarity matrix is insufficient because it only provides local affinities, not a structured set of correspondences. The model must decide how to group tokens into referential text masks, how to group pixels into instance-level image masks, how many correspondences exist, and which text and image groups should be paired.

3.1 Concept Correspondence Model

We build the model (**ConCor-1**) on top of a pretrained vision-language backbone (Qwen3.5 [24]). This choice is motivated by its strong long-context modeling, rich text understanding (e.g., coreference and compositionality), and broad multimodal knowledge acquired during pretraining, all of which are essential for identifying and aligning vision–text concepts without predefined text spans. We use the 0.8B variant to keep the model size comparable to existing grounding models. Although Qwen3.5 is originally designed for autoregressive language generation, we instead use it as a contextual image–text encoder and attach lightweight prediction heads to directly predict text masks, image masks, and correspondence presence.

As shown in Fig. 2, given an image–text pair, the backbone first produces a unified sequence of visual and textual tokens that encode both modalities in a shared representation space. We additionally introduce a set of learned *bridge tokens*, where each bridge token represents a candidate image–text correspondence. By attending jointly to visual and textual tokens, each bridge token aggregates the context needed to hypothesize a potential alignment between an image region and a text span. Conditioned on this representation, the model simultaneously predicts (1) an image mask for the corresponding visual segment, (2) a text mask for the corresponding text segment, and (3) a presence score indicating whether the proposed pairing is a valid correspondence.

Bridge token assignment. Because bridge tokens are fixed in number and initially unordered, while each image–text pair contains a variable number of correspondences, the model needs a training-time assignment strategy that encourages different tokens to specialize to different correspondence hypotheses. A simple approach is to follow raster order over image objects [1], but this provides no explicit structural prior. Instead, we introduce a multi-scale spatial assignment strategy: we allocate bridge tokens over grids at multiple resolutions, with token groups corresponding to partitions such as 1, 2×2, 3×3, and so on over the image. Each bridge token is therefore associated with a spatial cell at a particular scale. During training, we compute the IoU between each ground-truth image

mask and each candidate grid cell, and then use Hungarian matching to assign correspondences to bridge tokens. This gives bridge tokens both spatial and scale priors, encouraging them to specialize in objects with different locations and extents.

Presence head. For each bridge token, we predict a scalar *presence score* indicating whether it corresponds to a valid image–text concept pair. This head is implemented as a lightweight MLP applied to the final hidden state of the bridge token.

Text segmentation head. To recover the grounded text segment associated with a bridge token, we predict a binary mask over the text tokens in the input sequence. We first project bridge and text token features into a shared correspondence space, and then score each bridge–text-token pair with a learnable bilinear function, producing a logit for whether the token belongs to the grounded text segment represented by that bridge token.

Image segmentation head. To recover the corresponding image mask, we predict a binary pixel mask for each bridge token. Because the vision tokens in language models are spatially compressed through 2×2 patch merging, we first project them back to the original patch grid using learned projection layers. We then fuse these reconstructed VLM features with patch-level features from the vision encoder to restore local spatial detail. The fused features are processed by a lightweight convolutional decoder to produce a spatial feature map at the mask-prediction resolution. Similar to the text segmentation head, we project the bridge token and spatial features into a shared correspondence space, and use a learnable bilinear scoring function to predict whether each spatial location belongs to the image mask associated with that bridge token.

Training objectives. We train the model to jointly predict text masks, visual masks, and correspondence validity for each bridge token. Text masks are supervised with binary cross-entropy (BCE). For visual masks, we use a PointRend-style [48] point-sampled loss, combining BCE and Dice loss [49] over matched positive bridges. A separate BCE loss supervises the presence head, indicating whether each bridge token corresponds to a valid text–image pair. A more detailed description of the model and training implementation can be found in the appendix.

3.2 Concept correspondence dataset and evaluation

We convert diverse grounding, segmentation, and referring-expression datasets into a unified correspondence format with image masks, text masks, and correspondence labels, with additional dataset details and statistics provided in the appendix.

Repurposing existing grounded-caption datasets. Existing grounded-caption datasets are valuable but not directly compatible with our bidirectional correspondence formulation. Their annotations are often incomplete: co-referring text mentions are typically annotated only at first occurrence, and some mentioned entities lack corresponding image masks [11, 50, 9]. We therefore use an LLM-based rewriting pipeline that groups co-referring mentions into shared text masks and removes mentions without associated image masks. This yields complete correspondence annotations between coreference-aware text masks and instance-level image masks.

Training data. We include the following repurposed or constructed datasets as training data. GoldG [11] and COCONut-PanCap [50] are built from human-written captions and processed with the aforementioned LLM rewriting pipeline. GoldG is derived from Flickr30k [9] and GQA [51] annotations from the Grounding DINO training corpus, with pseudo masks generated from ground-truth boxes using SAM3. COCONut-PanCap uses relabeled COCO panoptic captions with ground-truth masks. We also construct GroundedRef-train from COCO panoptic segmentation by generating referring and compositional captions with an LLM and pairing them with ground-truth masks. Along with these caption-style datasets, we incorporate instance segmentation datasets, including COCO [52], COCONut [53], EntitySeg [54], and ADE20K [55]. We further include the counting subset of PixMo-Point [1], converting points to masks with SAM3 [12]. Overall, our training set contains 653K image–text pairs over 184K unique images for caption-style datasets, annotating 3.1M correspondences over 4.2M instance masks, and 614K images with 7.8M instance masks for instance segmentation data.

Evaluation data. For evaluation on image–caption pairs, we process the Flickr30k [9] and COCONut-PanCap [50] validation sets using the same LLM rewriting pipeline, followed by human filtering to ensure annotation quality. We also construct a separate GroundedRef validation set from the validation

Table 1: Image-caption results on COCONut-PanCap, GroundedRef, and Flickr30k validation sets.

	COCONut-PanCap						GroundedRef						Flickr30k					
Method	JointF1	MaskF1	TextF1	mJS	mMaskIoU	mSpanIoU	JointF1	MaskF1	TextF1	mJS	mMaskIoU	mSpanIoU	JointF1	MaskF1	TextF1	mJS	mMaskIoU	mSpanIoU
GDINO+SAM	34.1	64.8	39.8	48.0	67.9	44.2	45.0	56.4	50.9	37.8	44.2	39.7	81.6	85.2	86.7	81.9	86.9	81.5
MM-GDINO+SAM	32.2	59.5	37.4	51.1	70.8	47.4	39.8	58.1	44.3	31.9	44.2	33.3	84.6	87.7	89.6	82.6	87.3	82.3
LLMDet+SAM	31.5	52.8	37.9	59.1	75.4	55.7	54.6	69.0	62.8	68.1	**77.9**	70.1	76.7	79.6	85.4	75.3	78.7	77.1
GPT-5.4 (medium)+SAM	43.8	47.4	88.9	48.6	43.5	82.7	25.5	37.5	68.2	33.9	34.3	70.8	37.3	43.4	86.6	44.7	41.4	87.0
Florence-2	36.7	53.4	53.6	47.9	55.8	56.7	52.6	61.2	79.2	56.4	55.2	77.7	68.7	70.1	92.8	69.4	61.6	88.9
GLaMM	2.5	39.1	3.0	11.9	27.4	8.0	6.1	47.3	10.0	18.4	32.8	16.9	53.1	69.1	59.5	55.4	60.3	58.2
Qwen3.5-FT	59.9	63.4	93.1	63.3	55.6	87.9	51.5	56.3	**82.1**	49.7	45.8	77.5	70.2	72.6	**95.2**	69.1	60.3	95.1
ConCor-1 (random init.)	51.8	59.4	67.6	63.2	63.6	77.9	32.4	43.4	56.9	44.6	48.1	70.5	55.0	61.1	72.7	64.4	62.3	83.7
ConCor-1	**88.8**	**91.9**	**93.4**	**89.5**	**87.0**	**95.7**	**70.3**	**76.4**	78.7	**69.6**	68.8	**78.7**	**91.4**	**92.4**	95.1	**91.8**	**88.6**	**97.4**

Table 2: Image-category results on COCO validation set, LVIS-minival, and EntitySeg validation set.

	COCO						LVIS-minival						EntitySeg					
Method	JointF1	MaskF1	TextF1	mJS	mMaskIoU	mSpanIoU	JointF1	MaskF1	TextF1	mJS	mMaskIoU	mSpanIoU	JointF1	MaskF1	TextF1	mJS	mMaskIoU	mSpanIoU
GDINO+SAM	63.0	66.5	73.3	76.0	71.3	90.7	15.3	22.4	17.7	33.1	47.8	35.7	21.5	30.6	27.0	36.9	45.5	43.3
MM-GDINO+SAM	65.7	68.7	75.6	77.1	72.0	91.6	9.4	18.4	12.4	**53.6**	75.7	**56.8**	29.4	36.9	37.0	45.5	51.7	55.7
LLMDet+SAM	62.4	65.9	72.1	77.2	72.2	91.9	4.3	8.7	5.8	53.2	**76.0**	56.5	22.2	31.7	29.2	44.5	**56.2**	55.2
GPT-5.4 (medium)+SAM	34.2	35.1	**81.7**	39.7	32.5	80.0	17.2	20.9	**43.0**	25.8	26.3	53.7	20.2	30.1	42.4	30.7	34.4	51.9
Florence-2	9.8	14.8	11.9	7.6	10.1	9.1	6.2	17.7	8.2	17.0	38.0	19.6	11.8	20.9	23.5	21.2	28.8	32.1
GLaMM	38.5	40.8	45.7	33.1	31.5	38.9	23.1	24.0	29.7	19.4	18.4	24.1	22.2	23.1	26.0	18.3	17.4	21.2
Qwen3.5-FT	39.4	41.2	51.4	37.3	37.4	48.3	3.9	26.9	5.2	4.4	28.0	5.2	31.1	38.6	47.6	33.1	38.0	43.6
ConCor-1 (random init.)	15.1	32.8	22.9	25.1	49.5	35.6	0.1	18.5	0.2	0.2	32.5	0.3	2.8	20.4	5.4	4.3	26.6	6.5
ConCor-1	**72.4**	**74.3**	81.0	**79.8**	**74.2**	**92.7**	**29.9**	**41.4**	33.8	39.8	51.4	45.0	**49.8**	**59.9**	**59.1**	**48.9**	53.7	**57.4**

splits of RefCOCOg [5] and gRefCOCO [56]. Because these datasets primarily annotate only the target object referred to by each expression, we additionally annotate masks for contextual objects mentioned in the expression, using SAM3 proposals followed by manual verification. This makes the validation examples compatible with our task. The final evaluation sets contain 2K image–caption pairs for Flickr30k, 2.2K for COCONut-PanCap, and 2K for GroundedRef.

4 Experiments

We train the **ConCor-1** on 8 H100 GPUs for 100K optimization steps with an effective batch size of 96, using image cropping and collage data augmentations. Further implementation details are provided in the appendix. Our experiments evaluate bidirectional concept correspondence across image–caption and image–category settings.

4.1 Main results

Evaluation metrics. We evaluate predictions at three levels: text segmentation, image segmentation, and joint correspondence. For text segmentation, we match predictions to ground truth with Hungarian matching on character span IoU. A match is considered correct if span IoU ≥ 0.5, and we report **TextF1**. To capture quality beyond thresholded correctness, we also report **mSpanIoU**, the average span IoU of matched pairs over all ground-truth groups, with unmatched groups contributing zero. For image segmentation, we apply the same procedure using mask IoU, and report **MaskF1** and **mMaskIoU**. For joint correspondence, we perform Hungarian matching on the geometric mean of text and mask IoU, and count a match as correct only when both exceed 0.5. We report **JointF1** together with **mJS**, a soft joint score defined as the average geometric mean over matched pairs, with unmatched ground-truth instances assigned zero.

Baselines. We compare against open-vocabulary grounding baselines that localize candidate regions and produce grounded mask outputs. GDINO+SAM uses Grounding DINO [11] to detect caption-conditioned boxes and recover text spans from token similarity scores, then refines boxes into masks with SAM3 [12]. We also include its successors, MM-GDINO [36] and LLMDet [37]. Florence-2 [43] predicts grounded boxes and phrase labels from the caption, recovers character spans by phrase matching, and converts boxes to masks with its region-to-segmentation head. Long captions are chunked when needed for these methods. GLaMM [47] natively generates interleaved grounded phrases and masks; we condition its FullScope checkpoint on the target caption and match generated phrases back to caption spans. GPT-5.4 [57] (medium reasoning) predicts grounded caption spans or category names with boxes, which SAM3 [12] converts to masks. Finally, we fine-tune Qwen3.5-

Table 3: Ablation study. We train each variant for 15K steps and report the average score for each metric across tasks. Ablations are performed sequentially; the vision encoder is frozen for the scoring-function, segmentation-head, and attention ablations, and fine-tuned for the subsequent vision-feature and bridge-assignment ablations. Bidirectional attention is used for the vision-feature and bridge-assignment ablations.

Component	Variation	JointF1	MaskF1	TextF1	mJS	mMaskIoU	mSpanIoU
Scoring function	Dot product	48.7	55.4	60.3	**59.9**	**59.7**	**73.3**
	Bilinear	**49.3**	**56.0**	**61.2**	59.4	59.0	72.9
Image segmentation head	Bilinear-up (4x4)	47.8	54.3	60.3	59.1	58.0	**73.3**
	Two Conv-up (4x4)	**49.3**	56.0	**61.2**	**59.4**	59.0	72.9
	Three Conv-up (2x2)	48.8	56.3	60.1	59.0	**59.5**	71.6
	Four Conv-up (1x1)	48.8	**56.4**	60.6	58.5	59.3	70.4
Attention	Causal	49.3	56.0	61.2	59.4	59.0	72.9
	Bidirectional	**51.3**	**57.2**	**62.6**	**61.4**	**60.4**	**74.7**
Vision feature	Vision encoder feature + VLM vision tokens	52.4	59.4	62.5	**62.2**	**62.7**	**73.9**
	Vision encoder feature only	51.8	59.1	61.9	61.7	62.7	73.4
	VLM vision tokens only	**52.5**	**59.5**	**62.8**	61.0	61.4	72.8
Bridge token assignment	Raster order (64 tokens)	43.4	53.3	58.4	48.3	51.6	60.6
	Spatial assignment (8x8 tokens)	51.7	59.2	62.5	60.8	61.7	72.4
	Spatial assignment (11x11 tokens)	51.3	59.1	61.4	61.3	62.5	72.6
	Multi-scale spatial assignment	**52.4**	**59.4**	**62.5**	**62.2**	**62.7**	**73.9**

0.8B [24] (Qwen3.5-FT) with autoregressive supervision to output text spans and polygon masks for each correspondence.

Results overview. Table 1 reports results on three image-caption benchmarks: COCONut-PanCap, GroundedRef, and Flickr30k. Table 2 reports results on three image-category benchmarks: COCO, LVIS-minival, and EntitySeg, where category names are concatenated into a single text input; because the LVIS vocabulary is large, all baselines except Qwen3.5-FT require chunking the category list and running multiple forward passes. Across both settings, **ConCor-1** consistently achieves the best F1-based metrics, including JointF1, MaskF1, and TextF1, showing that it can recover text spans, image masks, and their alignments across both natural captions and category-list queries.

Findings. We list several findings from the results. (1) *Explicit correspondence prediction outperforms language-only generation.* Qwen3.5-FT formulates the task as autoregressive language generation of text spans and polygon masks, and performs strongly on text metrics, but **ConCor-1** improves JointF1 from 59.9 to 88.8 and mJS from 63.3 to 89.5 on COCONut-PanCap, showing that bridge tokens better bind text spans to image regions. (2) *The largest gains appear on long, dense captions.* The gains of **ConCor-1** over baselines are strongest on COCONut-PanCap, which contains longer captions with dense object references, showing that explicit bridge-token prediction is more effective for challenging many-to-many text–image alignment. (3) *Pretrained VLM representations are essential for robust correspondence learning.* The random-initialization gap highlights the value of pretrained visual and language representations for object recognition, referential understanding, and cross-modal alignment, especially in long, dense inputs. (4) *Detection-based grounding is not sufficient for full correspondence.* Detector-based baselines remain competitive on mask-only or IoU metrics, but lag on joint correspondence, showing that the key challenge is identifying visually referential text spans and matching them to the image. (5) ***ConCor-1** handles large-vocabulary category lists in a single forward pass.* On LVIS-minival, **ConCor-1** improves JointF1 from 23.1 to 29.9 over the strongest baseline, a 29% relative gain. While detector-based baselines require chunking the LVIS vocabulary into multiple inference passes, **ConCor-1** processes the full category list at once. (6) *Thresholded F1 and soft IoU reveal different strengths.* Some detector-based baselines achieve strong soft IoU metrics, reflecting good partial localization, while **ConCor-1** achieves the best thresholded F1 metrics, showing better text–image correspondence decisions.

4.2 Ablations

We ablate the main architectural choices of **ConCor-1**. Each variant is trained for 15K steps, and we report the average performance across evaluation tasks in Tab. 3.

Scoring function. We compare dot-product and bilinear scoring for predicting image and text masks from bridge-token features. Bilinear scoring improves the thresholded F1 metrics, suggesting that the learned metric compatibility separates foreground from background tokens. Dot-product scoring performs slightly better on IoU-based metrics, indicating that simpler similarity may produce smoother or better-calibrated continuous masks.

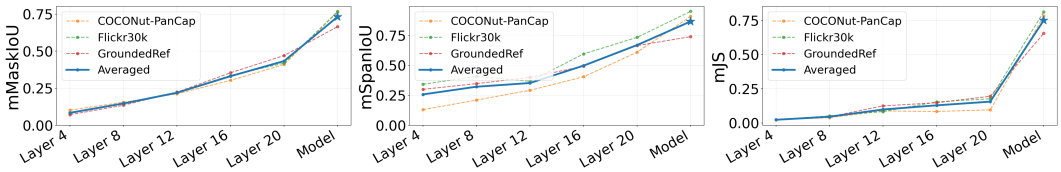


Figure 3: **Grounding emerges progressively across language model layers.** We probe bridge-to-text and bridge-to-vision attention across language-model layers and evaluate the induced masks with mSpanIoU, mMaskIoU, and mJS. Grounding quality improves progressively across layers, and the final prediction heads provide a further substantial boost.

Image segmentation head. We ablate the image segmentation decoder design. Bilinear upsampling performs worse than convolutional upsampling. Among convolutional variants, the two-layer decoder gives the best JointF1, TextF1, and mJS, while deeper decoders slightly improve MaskF1 or mMaskIoU but reduce text and joint correspondence quality. This suggests that increasing decoder depth can improve mask-only localization, but may not improve the alignment between text and image segments. We therefore use the two-convolution upsampling head by default.

Attention pattern. We ablate the attention pattern in the LLM full-attention layers. Replacing causal attention with bidirectional attention consistently improves performance across metrics. This suggests that correspondence prediction benefits from full image–text–bridge context, since the task requires resolving all alignments jointly rather than generating tokens autoregressively.

Vision features. We compare two sources of vision features for the image segmentation head: vision encoder features and VLM vision tokens. Using only VLM vision tokens yields the best F1 metrics, suggesting that the multimodal tokens provide stronger semantic context for deciding correspondences. In contrast, combining vision encoder features with VLM vision tokens gives the best soft localization metrics, indicating that the original vision encoder features preserve fine-grained spatial detail useful for mask quality. We therefore use the fused representation by default, as it improves IoU-based correspondence and localization metrics while remaining competitive on thresholded F1 metrics.

Bridge token assignment. We ablate how ground-truth correspondences are assigned to bridge tokens during training. A raster-order assignment performs substantially worse, while spatial grid-based assignment yields clear gains, indicating that localized priors are important for learning stable correspondence queries. Our multi-scale spatial assignment performs best overall, showing that bridge tokens benefit from covering objects at different sizes and locations.

4.3 Analysis and visualizations

Layer-wise grounding analysis. We further analyze where grounding information appears inside the VLM by directly using the attention between bridge tokens and text or vision tokens as a prediction signal. Specifically, for each selected language-model layer, we treat bridge-to-text attention as a text-mask prediction and bridge-to-vision attention as an image-mask prediction, and evaluate the resulting correspondences on COCONut-PanCap, Flickr30k, and GroundedRef. As shown in Fig. 3, lower layers contain little usable grounding information, while later layers yield consistently higher mMaskIoU, mSpanIoU, and mJS. This trend suggests that bridge tokens progressively accumulate cross-modal correspondence information as they pass through the language model. Nevertheless, the final model substantially outperforms all attention-only probes, showing that the learned segmentation heads are crucial for producing precise text spans and image masks.

Qualitative comparison. Fig. 4 shows qualitative examples on COCONut-PanCap. GDINO+SAM often detects plausible regions but struggles to use the caption context to determine which instance is being referred to, especially when multiple people or similar objects appear in the image. By jointly predicting text masks, image masks, and their correspondences, **ConCor-1** better disambiguates referred instances and produces more complete text–image alignments.

Text-to-image attention analysis and visualization.

We analyze text-to-image attention at full-attention Layers 4, 8, 12, 16, and 20 on the image-caption evaluation datasets. For each phrase, we average attention over heads and constituent text tokens, and compare the resulting visual-token map with its ground-truth mask. As shown in Tab. 4, **ConCor-1**

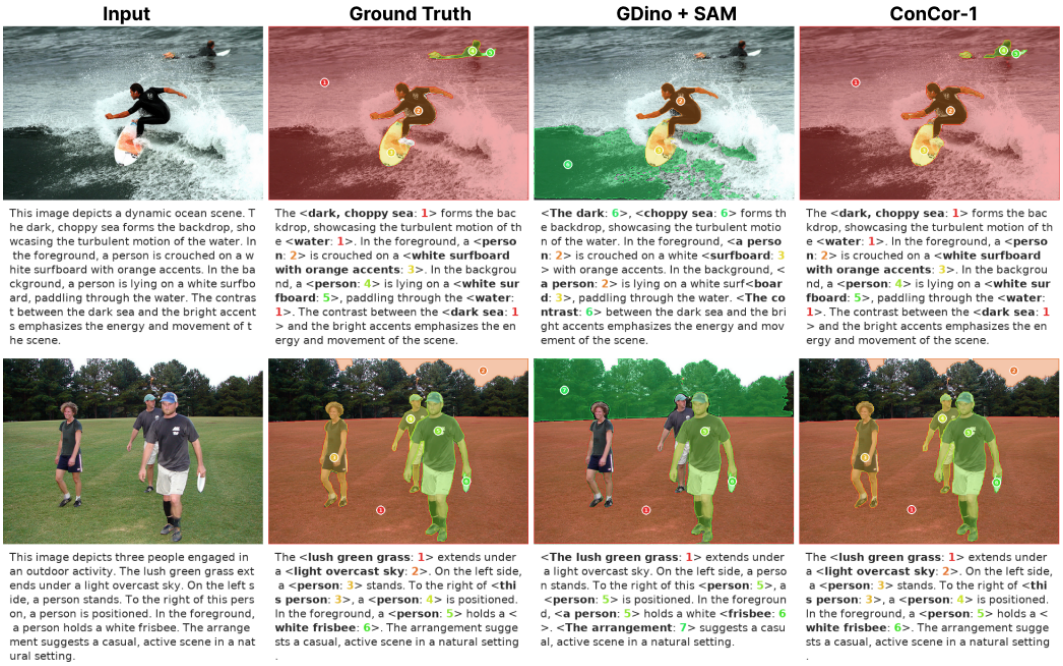


Figure 4: **Qualitative comparison on COCONut-PanCap.** We visualize ground truth, GDINO+SAM, and **ConCor-1** predictions, using matching colors for grounded text spans and image masks. **ConCor-1** better disambiguates referred instances and recovers more complete text–image correspondences, while GDINO+SAM can confuse different people or objects mentioned in the caption and may miss some grounded entities. Additional examples are provided in Appendix 8.

consistently achieves higher attention IoU and top-1 hit rate and lower spatial entropy than Qwen3.5 and Qwen3.5-FT across all analyzed layers, indicating more spatially aligned and concentrated phrase-level attention.

Table 4: Quantitative text-to-image attention analysis across full-attention layers, averaged over the image-caption evaluation datasets. Attn-IoU measures the overlap between the highest-attention visual tokens and the phrase-specific ground-truth mask, using the ground-truth mask area to determine the number of selected tokens. Top-1 Hit measures whether the visual token receiving the maximum attention lies inside the ground-truth mask. Entropy measures the normalized spatial dispersion of attention, where lower values indicate more concentrated attention.

Metric	Model	L4	L8	L12	L16	L20
Attn-IoU ↑	Qwen3.5	0.16	0.20	0.22	0.22	0.21
	Qwen3.5-FT	0.16	0.27	0.23	0.19	0.15
	ConCor-1	**0.29**	**0.42**	**0.48**	**0.33**	**0.57**
Top-1 Hit ↑	Qwen3.5	0.32	0.43	0.43	0.43	0.40
	Qwen3.5-FT	0.41	0.56	0.47	0.36	0.27
	ConCor-1	**0.56**	**0.70**	**0.70**	**0.55**	**0.75**
Entropy ↓	Qwen3.5	0.82	0.69	0.73	0.74	0.73
	Qwen3.5-FT	0.78	0.73	0.71	0.70	0.63
	ConCor-1	**0.73**	**0.66**	**0.65**	**0.68**	**0.54**

We visualize Layer 20 in Fig. 5 because it provides the strongest alignment for **ConCor-1**, achieving the highest attention IoU and top-1 hit rate and the lowest entropy among the analyzed layers. The original Qwen3.5 exhibits scattered attention patterns, with high responses on unrelated background regions and objects. Fine-tuning Qwen3.5 on our data makes the attention more localized, but the attended regions are still noisy and do not consistently cover the correct entity. In contrast, our model produces more compact and semantically aligned attention maps: the token "surfboard" attends to the surfboard region, while the token "man" attends to the person. This indicates that our

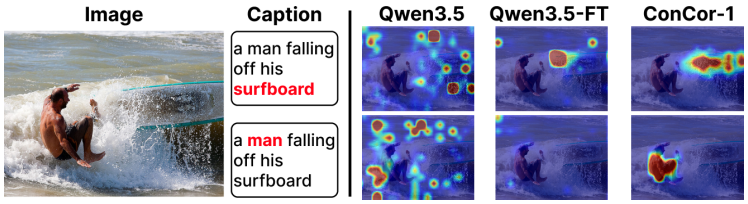


Figure 5: **Text-to-image attention visualization.** We visualize layer-20 attention from selected text tokens (highlighted in red) to image tokens for the original Qwen3.5, Qwen3.5 fine-tuned on our dataset, and our model. Compared with the diffuse or noisy baselines, **ConCor-1** produces more localized and semantically aligned attention. Additional visualizations are provided in Appendix 9.

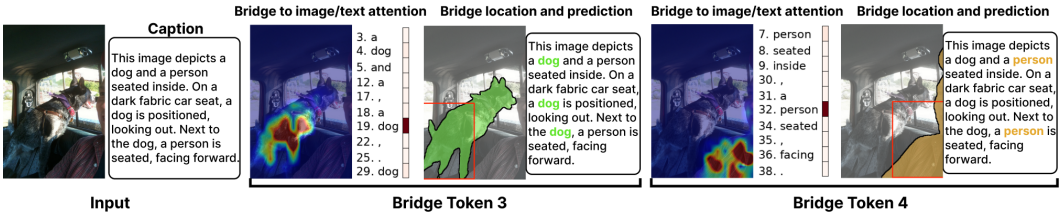


Figure 6: **Bridge-token attention and prediction.** We visualize two bridge tokens at layer 20 for one test sample, including bridge-to-image attention, bridge-to-text attention, spatial prior (red bounding box), and final prediction. Different bridge tokens specialize to distinct object-level correspondence hypotheses, such as the dog and the seated person. Additional examples are provided in Appendix 10.

explicit bridge-token correspondence objective not only improves final grounding predictions, but also induces cleaner cross-modal alignment in the underlying VLM representations.

Bridge-token attention visualization. We visualize bridge-to-image and bridge-to-text attention at layer 20, together with the spatial prior and final prediction associated with each bridge token. As shown in Fig. 6, different bridge tokens attend to different visual regions and text mentions, forming distinct correspondence hypotheses. For example, one bridge token focuses on the dog region and attends to repeated mentions of "dog", while another focuses on the seated person and attends to the corresponding person mentions. The final predictions show that these bridge-specific attention patterns are converted into coherent text–image correspondences, supporting the role of bridge tokens as object-level hypotheses that bind language and vision.

5 Discussion and Future Work

We present bidirectional concept correspondence as a structured formulation for grounding text segments, image segments, and their alignments from an image–text pair. This formulation unifies caption grounding, referring-expression grounding, and open-vocabulary grounding under a single output space. Our model **ConCor-1** provides a first step toward this goal by explicitly predicting text masks, image masks, and their correspondences. Future work includes extending correspondence to video, where alignments must persist over time, and using it as a training objective for better-aligned vision-language models. This task can also support the evaluation and improvement of vision-language models, such as checking whether generated captions are visually grounded or whether text-to-image outputs faithfully realize prompt entities and relations.

Acknowledgement

This project was partially funded by Toyota Motor Inc.

References

[1] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna

Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *CVPR*, 2025.

[2] Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, Vincent Shao, Yue Yang, Weikai Huang, Ziqi Gao, Taira Anderson, Jianrui Zhang, Jitesh Jain, George Stoica, Winson Han, Ali Farhadi, and Ranjay Krishna. Molmo2: Open weights and data for vision-language models with video understanding and grounding, 2026.

[3] Christopher Clark, Yue Yang, Jae Sung Park, Zixian Ma, Jieyu Zhang, Rohun Tripathi, Mohammadreza Salehi, Sangho Lee, Taira Anderson, Winson Han, et al. Molmopoint: Better pointing for vlms with grounding tokens. *arXiv preprint arXiv:2603.28069*, 2026.

[4] 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.

[5] 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.

[6] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *European Conference on Computer Vision*, pages 792–807. Springer, 2016.

[7] Ranjay Krishna, Ines Chami, Michael Bernstein, and Li Fei-Fei. Referring relationships. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6867–6876, 2018.

[8] 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.

[9] 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.

[10] Chenyun Wu, Zhe Lin, Scott Cohen, Trung Bui, and Subhransu Maji. Phrasecut: Language-based image segmentation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10216–10225, 2020.

[11] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[12] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025.

[13] Herbert H. Clark. *Using Language*. Cambridge University Press, Cambridge, 1996.

[14] Herbert H. Clark and Susan E. Brennan. Grounding in communication. In Lauren B. Resnick, John M. Levine, and Stephanie D. Teasley, editors, *Perspectives on Socially Shared Cognition*, pages 127–149, Washington, DC, 1991. American Psychological Association.

[15] Barbara Tversky. Spatial mental models. In Gordon H. Bower, editor, *The Psychology of Learning and Motivation*, volume 27, pages 109–145. Academic Press, 1991.

[16] Holly A. Taylor and Barbara Tversky. Spatial mental models derived from survey and route descriptions. *Journal of Memory and Language*, 31(2):261–292, 1992.

[17] Ainaz Eftekhar, Kuo-Hao Zeng, Jiafei Duan, Ali Farhadi, Ani Kembhavi, and Ranjay Krishna. Selective visual representations improve convergence and generalization for embodied ai. *arXiv preprint arXiv:2311.04193*, 2023.

[18] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[19] Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170, 1983.

[20] Arushi Goel, Basura Fernando, Frank Keller, and Hakan Bilen. Who are you referring to? coreference resolution in image narrations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15247–15258, 2023.

[21] Terry Winograd. Understanding natural language. *Cognitive Psychology*, 3(1):1–191, 1972.

[22] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 552–561, 2012.

[23] Brendan Park, Madeline Janecek, Naser Ezzati-Jivan, Yifeng Li, and Ali Emami. Picturing ambiguity: A visual twist on the winograd schema challenge. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 355–374, 2024.

[24] Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026.

[25] Yuan Yao, Qiushi Yang, Humen Zhong, Jiangning Wei, Yifang Men, Shuai Bai, Miaomiao Cui, and Zhibo Yang. Qwen3-vl-seg: Unlocking open-world referring segmentation with vision-language grounding. *arXiv preprint arXiv:2605.07141*, 2026.

[26] Chenhao Zheng, Jieyu Zhang, Mohammadreza Salehi, Ziqi Gao, Vishnu Iyengar, Norimasa Kobori, Quan Kong, and Ranjay Krishna. One trajectory, one token: Grounded video tokenization via panoptic sub-object trajectory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23156–23166, 2025.

[27] Chenhao Zheng, Jieyu Zhang, Jianing Zhang, Weikai Huang, Ashutosh Kumar, Quan Kong, Oncel Tuzel, Chun-Liang Li, and Ranjay Krishna. Trajtok: Learning trajectory tokens enables better video understanding. *arXiv preprint arXiv:2602.22779*, 2026.

[28] Zhixiang Wei, Yi Li, Zhehan Kan, Xinghua Jiang, Zuwei Long, Shifeng Liu, Hongze Shen, Wei Liu, Xiaoyu Tan, Haojia Lin, et al. Youtu-vl: Unleashing visual potential via unified vision-language supervision. *arXiv preprint arXiv:2601.19798*, 2026.

[29] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022.

[30] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *Advances in Neural Information Processing Systems*, 35:36067–36080, 2022.

[31] Jae Sung Park, Zixian Ma, Linjie Li, Chenhao Zheng, Cheng-Yu Hsieh, Ximing Lu, Khyathi Chandu, Quan Kong, Norimasa Kobori, Ali Farhadi, et al. Synthetic visual genome. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9073–9086, 2025.

[32] Ziqi Gao, Jieyu Zhang, Wisdom Oluchi Ikezogwo, Jae Sung Park, Tario G. You, Daniel Ogbu, Chenhao Zheng, Weikai Huang, Yinuo Yang, Winson Han, Quan Kong, Rajat Saini, and Ranjay Krishna. Synthetic visual genome 2: Extracting large-scale spatio-temporal scene graphs from videos, 2026.

[33] 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.

[34] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. *Advances in Neural Information Processing Systems*, 35:9125–9138, 2022.

[35] Lewei Yao, Jianhua Han, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, and Hang Xu. Detclipv2: Scalable open-vocabulary object detection pre-training via word-region alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23497–23506, 2023.

[36] Xiangyu Zhao, Yicheng Chen, Shilin Xu, Xiangtai Li, Xinjiang Wang, Yining Li, and Haian Huang. An open and comprehensive pipeline for unified object grounding and detection. *arXiv preprint arXiv:2401.02361*, 2024.

[37] Shenghao Fu, Qize Yang, Qijie Mo, Junkai Yan, Xihan Wei, Jingke Meng, Xiaohua Xie, and Wei-Shi Zheng. Llmdet: Learning strong open-vocabulary object detectors under the supervision of large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14987–14997, 2025.

[38] Junfeng Wu, Yi Jiang, Qihao Liu, Zehuan Yuan, Xiang Bai, and Song Bai. General object foundation model for images and videos at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3783–3795, 2024.

[39] Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15116–15127, 2023.

[40] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1020–1031, 2023.

[41] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. In *NeurIPS*, 2023.

[42] Shiting Xiao, Rishabh Kabra, Yuhang Li, Donghyun Lee, Joao Carreira, and Priyadarshini Panda. Open-worldsam: Extending sam2 for universal image segmentation with language prompts. *arXiv preprint arXiv:2507.05427*, 2025.

[43] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, 2024.

[44] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. In *arXiv preprint arXiv:2306.14824*, 2023.

[45] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023.

[46] Senqiao Yang, Tianyuan Qu, Xin Lai, Zhuotao Tian, Bohao Peng, Shu Liu, and Jiaya Jia. Lisa++: An improved baseline for reasoning segmentation with large language model. *arXiv preprint arXiv:2312.17240*, 2023.

[47] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13009–13018, 2024.

[48] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9799–9808, 2020.

[49] 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.

[50] Xueqing Deng, Qihang Yu, Ali Athar, Chenglin Yang, Linjie Yang, Xiaojie Jin, Xiaohui Shen, and Liang-Chieh Chen. Coconut-pancap: Joint panoptic segmentation and grounded captions for fine-grained understanding and generation. *arXiv preprint arXiv:2502.02589*, 2025.

[51] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[52] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[53] Xueqing Deng, Qihang Yu, Peng Wang, Xiaohui Shen, and Liang-Chieh Chen. Coconut: Modernizing coco segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21863–21873, June 2024.

[54] Lu Qi, Jason Kuen, Weidong Guo, Jiuxiang Gu, Zhe Lin, Bo Du, Yu Xu, and Ming-Hsuan Yang. High-quality entity segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4047–4056, 2023.

[55] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International journal of computer vision*, 127(3):302–321, 2019.

[56] Shuting He, Henghui Ding, Chang Liu, and Xudong Jiang. Grec: Generalized referring expression comprehension. *arXiv preprint arXiv:2308.16182*, 2023.

[57] OpenAI. Introducing GPT-5.4. `https://openai.com/index/introducing-gpt-5-4/`, March 2026. Accessed: 2026-07-26.

[58] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[59] Peter Robicheaux, Matvei Popov, Anish Madan, Isaac Robinson, Joseph Nelson, Deva Ramanan, and Neehar Peri. Roboflow100-vl: A multi-domain object detection benchmark for vision-language models. *arXiv preprint arXiv:2505.20612*, 2025.

A Limitations and Societal Impact

Limitations. First, **ConCor-1**'s mask quality remains behind that of specialized detection and segmentation models. This is partly because our image-mask head is intentionally lightweight and designed for joint text-image correspondence prediction rather than optimizing segmentation quality alone. Second, our approach depends on correspondence-style supervision, where grounded text spans and image masks are jointly annotated. Since most existing datasets are not natively annotated in this format, we convert several grounded-caption, detection, segmentation, and visual-coreference datasets into a unified representation, which may introduce noise and leave some forms of supervision underused. Future work could study whether large-scale single-modality datasets, such as segmentation-only or text-only coreference data, can further improve the visual or linguistic side of correspondence prediction. Third, while our evaluation covers captions, referring expressions, and category-name inputs, future benchmarks should include longer and more compositionally complex captions with more entities, repeated mentions, nested references, attributes, relations, and discourse-level coreference. These settings would better test whether models can recover complete correspondence structures under realistic language complexity.

Societal Impact. The societal impact of this work depends on how grounding models are used in downstream systems. More precise image-text correspondence can improve interpretability by making vision-language predictions spatially attributable, which may benefit applications such as visual search, dataset annotation, assistive perception, and embodied agents. However, grounding errors may also propagate to downstream decisions, especially in human-facing or safety-critical settings. For example, incorrect correspondences could cause an assistive system to refer to the wrong object, a robotic system to act on an unintended target, or an annotation pipeline to amplify dataset bias. Since grounding models reflect the distributions and biases of their training data, they may also perform unevenly across object categories, visual domains, languages, and cultural contexts. We therefore view this work as a step toward more transparent and structured vision-language understanding, but recommend careful validation before deployment in high-stakes applications.

B Implementation Details

The implementation of **ConCor-1** follows a grounding-only design: the model does not autoregressively decode an output string, but instead appends learnable bridge tokens to the multimodal input sequence and directly predicts image masks, text masks, and correspondence presence from the final token representations.

B.1 Backbone and Input Sequence

We instantiate the backbone from the pretrained Qwen3.5-0.8B vision-language model. **ConCor-1** uses the VLM backbone as a feature extractor to obtain the final hidden state sequence, rather than as a conditional language generator. We therefore apply bidirectional attention to the full-attention layers of the Qwen3.5 language backbone, allowing the bridge tokens to access the complete multimodal context and to differentiate from one another, while preserving the original attention behavior in the linear-attention layers. This preserves the pretrained hybrid-attention structure of Qwen3.5, but allows the full-attention layers to jointly contextualize visual tokens, text tokens, and bridge tokens.

For each image–text pair, we construct a flat multimodal sequence:

$$[\texttt{<vision_start>},\ \texttt{<image_pad>}_{1:N_v},\ \texttt{<vision_end>},\ x^{\text{text}}_{1:N_t},\ b_{1:Q}],$$

where N_v is the number of merged visual tokens, N_t is the number of text tokens, and Q is the number of bridge tokens.

For all main-table experiments, inspired by detection-based grounding methods that operate under a controlled image scale, we fix the image pixel budget at 1,003,520. We retain Qwen3.5's native aspect-ratio-preserving dynamic resizing, such that the image height and width adapt to the original aspect ratio rather than being forced to a fixed square resolution. This provides an approximately fixed visual-token budget across images while preserving their original geometry.

B.2 Bridge Tokens and Multi-scale Assignment

The bridge tokens are learnable embeddings appended after the image and text tokens. Each bridge token represents one candidate text–image correspondence. We organize bridge tokens into multi-scale spatial groups. Specifically, we use grid levels from 1×1 to 10×10. A level s contributes s^2 bridge tokens, so the total number of bridges is

$$Q = \sum_{s=1}^{10} s^2 = 385.$$

The bridge tokens are ordered by scale, and within each scale they are ordered in raster order over the corresponding grid. This gives the bridge set an implicit spatial and scale structure: coarse levels can represent large instances, while fine levels can represent smaller localized instances.

During training, each ground-truth correspondence is assigned to one bridge token. For each bridge token, we define its associated grid cell according to its scale and spatial position. We then perform Hungarian matching between masks of each correspondence and bridge tokens, using the mask–cell overlap as the primary criterion and the distance between the mask centroid and the cell center as a tie-breaker.

B.3 Prediction Heads

Given the final hidden states from the backbone, **ConCor-1** extracts three groups of token features: bridge-token features, text-token features, and visual-token features. The bridge-token features are shared by all three prediction heads. Each head first produces logits and then applies a sigmoid function to obtain probabilities for binary prediction.

Presence head. The presence head predicts whether a bridge token corresponds to a valid grounded correspondence. It takes the final hidden state of each bridge token as input and outputs a scalar logit:

$$z_j^{\mathrm{pres}} = f_{\mathrm{pres}}(\mathbf{b}_j),$$

where $\mathbf{b}_j \in \mathbb{R}^{1024}$ denotes the representation of the j-th bridge token. We implement f_{pres} as a SwiGLU MLP following the gated feed-forward style of the backbone: a gated projection and an up projection are combined with a SiLU nonlinearity, normalized, and projected to a scalar. The logit is then converted into a presence probability by a sigmoid function. The resulting presence score is used to determine whether the bridge should produce a valid text–image correspondence.

Text segmentation head. The text segmentation head predicts the text span associated with each bridge token. Rather than predicting explicit start and end indices, we formulate text grounding as token-level binary segmentation. For a bridge representation $\mathbf{b}_j$ and a text-token representation $\mathbf{x}_\ell^{\mathrm{text}}$, we project both into a shared correspondence space using independent SwiGLU MLPs:

$$\phi_b(\mathbf{b}_j),\ \phi_t(\mathbf{x}_\ell^{\mathrm{text}}) \in \mathbb{R}^{256}.$$

We then compute a learned bilinear compatibility score:

$$z_{j,\ell}^{\mathrm{text}} = \phi_t(\mathbf{x}_\ell^{\mathrm{text}})^\top W_{\mathrm{text}} \phi_b(\mathbf{b}_j).$$

Applying this scorer to all bridge–text-token pairs yields

$$\mathbf{Z}^{\mathrm{text}} \in \mathbb{R}^{B \times Q \times N_t}.$$

Vision segmentation head. The vision segmentation head predicts the image segment associated with each bridge token. It contains two components: a feature decoder that reconstructs dense visual features from the compressed VLM visual tokens, and a mask predictor that scores each bridge against the decoded visual features.

The Qwen3.5 vision encoder uses 16×16 image patches followed by 2×2 spatial merging. Thus, each merged visual token corresponds to a 32×32 image region. Directly predicting masks over these merged tokens is spatially coarse, so our model uses a fine decoding path. First, each merged visual token is expanded into four patch-level features by four individual MLPs, corresponding to the top-left, top-right, bottom-left, and bottom-right 16×16 patches before spatial merging. This

step recovers the local 2×2 patch layout inside each merged token. To preserve low-level visual detail, we additionally use a shortcut from the vision encoder to fuse the decoded patch-level features with the corresponding pre-merger ViT patch features. This fusion combines the language-model contextual representation with local visual features that are closer to the original image grid.

After this patch-level reconstruction, the features are rearranged into a spatial feature map and refined by a lightweight convolutional upsampling decoder. The decoder contains two successive upsampling blocks. Each block applies a stride-2 transposed convolution, followed by a SiLU activation, a depthwise 3×3 convolution for local refinement, and a two-dimensional layer normalization. Since each block upsamples the feature map by a factor of two, the two-block decoder refines the representation from the 16×16 patch level to a 4×4 pixel-block grid.

Given a decoded visual feature $\mathbf{v}_n$ at spatial location n and a bridge representation $\mathbf{b}_j$, the mask predictor projects both into a shared 256-dimensional correspondence space and applies a learned bilinear scorer:

$$z_{j,n}^{\text{vis}} = \phi_v(\mathbf{v}_n)^\top W_{\text{vis}} \phi_b(\mathbf{b}_j).$$

This produces a fine-resolution mask logit map for every bridge token:

$$\mathbf{Z}^{\text{vis}} \in \mathbb{R}^{B \times Q \times N_{\text{fine}}}.$$

We supervise this fine mask prediction directly during training. At inference time, the predicted fine mask grid is reshaped into its spatial layout and bilinearly interpolated to the original image resolution before thresholding and evaluation.

B.4 Training Loss

We supervise the three outputs of each bridge token with a presence loss, a text segmentation loss, and a visual segmentation loss. Expanding the visual term into its sampled-point BCE and Dice components, the total objective is a weighted sum of four terms:

$$\mathcal{L} = \lambda_{\text{pres}} \mathcal{L}_{\text{pres}} + \lambda_{\text{text}} \mathcal{L}_{\text{text}} + \underbrace{\lambda_{\text{bce}} \mathcal{L}_{\text{bce}}^{\text{pt}} + \lambda_{\text{dice}} \mathcal{L}_{\text{dice}}^{\text{pt}}}_{\mathcal{L}_{\text{vis}}}.$$

We use $\lambda_{\text{text}} = 1.0$, $\lambda_{\text{bce}} = 2.0$, $\lambda_{\text{dice}} = 0.5$, and $\lambda_{\text{pres}} = 2.0$ for all main experiments; this setting is selected by the sweep reported in Tab. 5.

Presence loss. The bridge assignment defines the binary presence target. A bridge token is labeled positive if it is matched to a ground-truth correspondence and negative otherwise. We train the presence head with binary cross-entropy to select which candidate bridge tokens correspond to valid grounded concepts. This term is scaled by λ_{pres} and, unlike the two segmentation terms, is applied to every bridge token rather than only the matched positives.

Text segmentation loss. The text target for each positive bridge is a binary mask over text tokens. We derive this mask from the ground-truth character spans: a text token is labeled positive if its offset overlaps the character span of the assigned correspondence. The text segmentation loss is binary cross-entropy over valid text-token positions, scaled by λ_{text}.

Visual segmentation loss. The visual target for each positive bridge is a binary ground-truth mask at the processed image resolution. We supervise visual masks with a PointRend-style [48] point-sampled loss. For each matched positive bridge, we reshape its finest visual logits into a spatial logit map, over-sample candidate points uniformly, and select a fixed fraction of points with the highest uncertainty, measured by the negative absolute logit magnitude. The remaining points are sampled uniformly for spatial coverage. The same normalized coordinates are used to sample both predicted logits and high-resolution binary ground-truth masks.

As written above, the visual loss $\mathcal{L}_{\text{vis}}$ is the weighted sum of a sampled-point BCE term and a sampled-point Dice term, with weights λ_{bce} and λ_{dice}. The BCE term provides local point-wise supervision, while the Dice term encourages mask-level overlap. Visual supervision is applied only to matched positive bridge tokens; unmatched bridges are trained only through the presence loss.

Choice of loss weights. The four terms are not naturally on the same scale, so their relative weighting matters. We sweep $(\lambda_{\text{text}}, \lambda_{\text{bce}}, \lambda_{\text{dice}}, \lambda_{\text{pres}})$ over five settings and report the results in Tab. 5. All runs in this sweep share the same configuration: a jointly fine-tuned vision encoder, bidirectional attention, and 30K training steps. Each entry is averaged over our evaluation benchmarks, and we report the six headline metrics. $(1, 2, 0.5, 2)$ is best on all six metrics, so we adopt it for the main experiments.

Table 5: Effect of the loss weights $(\lambda_{\text{text}}, \lambda_{\text{bce}}, \lambda_{\text{dice}}, \lambda_{\text{pres}})$.

λ_{text}:λ_{bce}:λ_{dice}:λ_{pres}	JointF1	MaskF1	TextF1	mJS	mMaskIoU	mSpanIoU
1 : 0.5 : 0.5 : 1	60.5	66.8	70.7	66.8	66.5	78.4
1 : 2 : 0.5 : 1	61.3	67.2	71.3	67.3	66.8	78.9
1 : 5 : 5 : 1	59.9	66.1	69.9	66.0	66.0	77.4
1 : 2.5 : 0 : 2	59.7	65.5	70.8	66.1	65.6	79.3
1 : 2 : 0.5 : 2	**62.0**	**67.7**	**72.1**	**68.0**	**67.3**	**79.5**

B.5 Backbone and Input Sequence

Tab. 6 compares the model size and computational profile of **ConCor-1** with its Qwen3.5-0.8B backbone. We profile one forward pass using a 512×512 image, 512 text tokens, batch size 1, bf16 precision, and an A100-80GB GPU. **ConCor-1** introduces 13.80M additional parameters, increasing the model size by 1.62%. Its 385 bridge tokens increase the sequence length from 770 to 1,155, resulting in a 46.0% increase in FLOPs. The measured peak GPU memory and forward latency increase by 9.5% and 2.6%, respectively.

Tab. 7 further decomposes the parameter and computational overhead. Most of the additional computation comes from processing the bridge tokens through the 24-layer language backbone, which contributes an additional 439.9 GFLOPs. The three prediction heads together add 47.8 GFLOPs, with the visual segmentation head accounting for most of this cost.

B.6 Data Augmentation and Data Mixture

We apply two image-level augmentations to improve the coverage of object layout and scale. The collage augmentation combines multiple training images into a single canvas using either a grid or horizontal/vertical concatenation. The corresponding instance masks are transformed into the collage coordinate system and re-encoded, and the text supervision is rebuilt according to the categories present in the augmented image. This increases the number of grounded instances per sample and distributes instances across different spatial cells, yielding denser supervision for the multi-scale bridge layout. The crop augmentation samples a random crop with constrained scale and aspect ratio, transforms the masks into the cropped image, and keeps only instances that remain sufficiently visible. The category text and character-span annotations are then rebuilt for the remaining instances. This augmentation improves robustness to partial objects, scale variation, and local context changes. It also changes the relative position and extent of objects within the image, which helps reduce center and layout biases in the training data.

We report the training data mixture in Tab. 8.

Table 6: Model size and computational profile. FLOPs, peak GPU memory, and latency are measured for one forward pass with a 512×512 image, 512 text tokens, batch size 1, bf16 precision, and an A100-80GB GPU.

Model	Seq. Len.	FLOPs	Peak Mem.	Latency	Params
Qwen3.5-0.8B backbone	770	1.061 T	1.687 GiB	54.0 ms	852.99 M
ConCor-1	1,155	1.549 T	1.846 GiB	55.4 ms	866.79 M
Δ	+385	+46.0%	+9.5%	+2.6%	+1.62%

Table 7: Component-level parameter and FLOPs breakdown. The computation induced by the bridge tokens is included in the language-model row.

Component	Added Params	GFLOPs		
	(M)	Backbone	ConCor-1	Δ
Vision tower	–	221.5	221.5	–
Language model	–	839.3	1,279.2	+439.9
New bridge embeddings	0.145	–	–	–
Visual segmentation head	12.01	–	46.3	+46.3
Text segmentation head	1.11	–	1.1	+1.1
Presence head	0.52	–	0.4	+0.4
Total	**13.80**	**1,060.8**	**1,548.5**	**+487.7**

Table 8: Training data mixture used for **ConCor-1**. We report the source-level mixture weight in the final training setup. The image augmentation column indicates the fraction of each source group sampled with collage or crop augmentation; the remaining fraction is sampled without image augmentation.

Source dataset	Mix weight	Image aug.
Instance Segmentation Data		
COCO [52]	15%	80% collage
EntitySeg [54]	10%	10% crop
PixMo Points [1]	10%	10% crop
COCONut [50]	5%	70% collage, 10% crop
SA-1B [58]	5%	10% crop
ADE20K [55]	5%	50% collage, 20% crop
Roboflow-VL-100 [59]	5%	—
Caption Grounding Data		
GoldG [11]	18%	—
COCONut-PanCap [50]	15%	—
GroundedRef	12%	—
Total	100%	—

B.7 Inference

During inference, **ConCor-1** produces a set of candidate correspondences from the bridge tokens. We keep bridge tokens whose presence scores exceed the inference threshold. For the main results, we use a presence threshold of 0.1.

For each retained bridge token, the text segmentation head outputs token-level logits over the input text. We apply sigmoid activation and threshold the resulting probabilities to obtain a binary text-token mask. We use a text threshold of 0.45 in the experiments.

For the visual prediction, the vision segmentation head outputs mask logits on the fine 4×4 pixel-block grid. We reshape these logits into their spatial layout and bilinearly upsample them to the original image resolution. The upsampled logits are then passed through a sigmoid function to obtain pixel-level mask probabilities, which are thresholded to produce the final binary mask. We use a visual threshold of 0.45 for the main results.

Because overlapping bridge tokens may predict highly similar correspondences, we apply non-maximum suppression to the retained predictions. Predictions are ranked by their presence scores, and duplicate lower-scoring predictions are removed using an IoU threshold of 0.5. The remaining predictions are used as the final set of text–image correspondences.

B.8 Ablation Details

For the scoring-function and image-segmentation-head ablations, we use causal attention and keep the vision encoder frozen. We then ablate causal versus bidirectional attention while continuing to freeze the vision encoder. After selecting bidirectional attention, we jointly fine-tune the vision encoder for the vision-feature and bridge-token-assignment ablations. For all ablations, we use bridge levels 1×1 to 5×5 and 8×8. The training mixture excludes EntitySeg, Roboflow-VL-100, and

GoldG, while adding LVIS. Each model is trained for 15K steps, with an NMS IoU threshold of 0.9 at inference. All other settings follow the main experiments.

C Evaluation metrics details

Our evaluation framework assesses grounded caption predictions along three independent axes: *text segmentation* (whether the model correctly groups character spans), *image segmentation* (whether the predicted masks align with ground truth), and *joint correspondence* (whether the model correctly associates text spans with masks). Each axis follows the same general procedure: (1) compute a pairwise similarity matrix between all predicted and ground-truth groups for a given image, (2) solve an optimal one-to-one assignment via Hungarian matching, and (3) derive hard and soft scores from the matched pairs. All metrics are computed per image and then macro-averaged across images.

C.1 Notation

Let $\mathcal{G} = \{G_1, \ldots, G_M\}$ denote the set of ground-truth groups for an image and $\hat{\mathcal{G}} = \{\hat{G}_1, \ldots, \hat{G}_K\}$ the predicted groups. Each group G_i consists of a set of character spans $T_i \subseteq \mathbb{Z}$ (the union of all $[\text{start}, \text{end})$ intervals) and a binary segmentation mask $S_i \in \{0,1\}^{H \times W}$. We write $M = |\mathcal{G}|$ for the number of GT groups and $K = |\hat{\mathcal{G}}|$ for the number of predictions.

C.2 Text Segmentation

Text segmentation evaluates whether the model correctly identifies which character spans in the caption correspond to each visual entity, without considering the predicted masks.

Span IoU. For each pair $(G_i, \hat{G}_j)$, we compute the character-level Intersection over Union between their span sets. All $[\text{start}, \text{end})$ intervals within a group are first unioned into a single set of character indices:

$$\text{spanIoU}(G_i, \hat{G}_j) = \frac{|T_i \cap \hat{T}_j|}{|T_i \cup \hat{T}_j|}$$

This formulation correctly handles adjacent spans (e.g., $[0,5) + [5,10)$ versus $[0,10)$ yields IoU $= 1.0$) and overlapping spans within a group.

Matching. We construct an $M \times K$ similarity matrix $\mathbf{A}^{\text{text}}$ where $A^{\text{text}}_{ij} = \text{spanIoU}(G_i, \hat{G}_j)$. Hungarian matching is applied by minimizing $-\mathbf{A}^{\text{text}}$, producing an optimal one-to-one assignment π between GT and predicted groups. No similarity threshold is applied during matching; all assigned pairs are retained regardless of their score.

Hard metrics. Given a threshold τ, a matched pair $(G_i, \hat{G}_{\pi(i)})$ is counted as a true positive if $\text{spanIoU}(G_i, \hat{G}_{\pi(i)}) \geq \tau$. Let TP_τ denote the number of such pairs. We define:

$$\text{TextP@}\tau = \frac{\text{TP}_\tau}{K}, \quad \text{TextR@}\tau = \frac{\text{TP}_\tau}{M}, \quad \text{TextF1@}\tau = \frac{2 \cdot \text{TextP@}\tau \cdot \text{TextR@}\tau}{\text{TextP@}\tau + \text{TextR@}\tau} \tag{1}$$

We report these at $\tau = 0.5$ and $\tau = 0.75$.

Soft metric. The mean span IoU averages the similarity of each GT group's best-matched prediction, assigning zero to unmatched GT groups:

$$\text{mSpanIoU} = \frac{1}{M} \sum_{i=1}^{M} \begin{cases} A^{\text{text}}_{i,\pi(i)} & \text{if } G_i \text{ is matched} \\ 0 & \text{otherwise} \end{cases}$$

C.3 Image Segmentation

Image segmentation evaluates the spatial quality of predicted masks, without considering character spans. The structure is identical to Case 1 with mask IoU replacing span IoU.

Mask IoU. For each pair $(G_i, \hat{G}_j)$, the standard pixel-level IoU is computed between their binary segmentation masks:

$$\text{MaskIoU}(G_i, \hat{G}_j) = \frac{|S_i \cap \hat{S}_j|}{|S_i \cup \hat{S}_j|}$$

Matching. An independent $M \times K$ similarity matrix $\mathbf{A}^{\text{mask}}$ is constructed with $A^{\text{mask}}_{ij} = \text{MaskIoU}(G_i, \hat{G}_j)$, and a separate Hungarian assignment σ is computed. This matching is independent of the text matching in Case 1.

Hard metrics.

$$\text{MaskP@}\tau = \frac{\text{TP}_\tau}{K}, \quad \text{MaskR@}\tau = \frac{\text{TP}_\tau}{M}, \quad \text{MaskF1@}\tau = \frac{2 \cdot \text{MaskP@}\tau \cdot \text{MaskR@}\tau}{\text{MaskP@}\tau + \text{MaskR@}\tau} \tag{2}$$

where

$$\text{TP}_\tau = \left| \left\{ i : \text{MaskIoU}(G_i, \hat{G}_{\sigma(i)}) \geq \tau \right\} \right|.$$

We report these metrics at $\tau = 0.5$ and $\tau = 0.75$.

Soft metric.

$$\text{mMaskIoU} = \frac{1}{M} \sum_{i=1}^{M} \begin{cases} A^{\text{mask}}_{i,\sigma(i)} & \text{if } G_i \text{ is matched} \\ 0 & \text{otherwise} \end{cases}$$

C.4 Joint Correspondence

Joint evaluation measures whether the model correctly *associates* the right text spans with the right masks. Unlike Cases 1 and 2, which evaluate each modality independently, this case requires both the text grouping and the segmentation to be correct for the same prediction.

Joint similarity. The joint score matrix is computed from the element-wise combination of the text and mask matrices from Cases 1 and 2. We use the geometric mean:

$$A^{\text{joint}}_{ij} = \sqrt{A^{\text{text}}_{ij} \cdot A^{\text{mask}}_{ij}}$$

This penalizes imbalanced predictions: if either the text or mask component is poor, the joint score drops sharply. For instance, a pair with spanIoU $= 0.9$ and MaskIoU $= 0.1$ receives a joint score of only 0.3.

Matching. Hungarian matching is applied to $\mathbf{A}^{\text{joint}}$, producing a third assignment ρ that is generally different from the text-only (π) and mask-only (σ) assignments. This matching jointly optimizes for both modalities.

Correctness criterion. A matched pair $(G_i, \hat{G}_{\rho(i)})$ is counted as a true positive only if *both* individual scores exceed the threshold:

$$\text{TP}_\tau = \left| \left\{ i : \text{spanIoU}(G_i, \hat{G}_{\rho(i)}) \geq \tau \text{ and } \text{MaskIoU}(G_i, \hat{G}_{\rho(i)}) \geq \tau \right\} \right|.$$

This is strict: a prediction with perfect text but poor segmentation (or vice versa) is counted as incorrect.

Hard metrics.

$$\text{JointP@}\tau = \frac{\text{TP}_\tau}{K}, \quad \text{JointR@}\tau = \frac{\text{TP}_\tau}{M}, \quad \text{JointF1@}\tau = \frac{2 \cdot \text{JointP@}\tau \cdot \text{JointR@}\tau}{\text{JointP@}\tau + \text{JointR@}\tau} \tag{3}$$

We report these metrics at $\tau = 0.5$ and $\tau = 0.75$.

Soft metric. The mean joint score averages the geometric-mean similarity over all GT groups:

$$\text{mJS} = \frac{1}{M} \sum_{i=1}^{M} \begin{cases} A^{\text{joint}}_{i,\rho(i)} & \text{if } G_i \text{ is matched} \\ 0 & \text{otherwise} \end{cases}$$

D Dataset curation

We train on a unified mixture of caption-grounding and instance segmentation data that combines human-annotated benchmarks, converted existing datasets, and automatically constructed supervision (Tab. 9). Our caption-grounding data includes GoldG, COCONut-PanCap, and GroundedRef, which provide human-written or synthetic captions paired with object-level masks, while our instance segmentation data includes standard benchmarks such as COCO, ADE20K, COCONut, EntitySeg, converted Roboflow-VL-100 datasets, and two automatically curated sources from PixMo Points and SA-1B. This mixture balances precise mask supervision from established datasets with the dense scenes, long-tail object coverage, and diverse text queries needed to train a grounding model that generalizes across both segmentation-style and caption-grounding settings.

Table 9: Statistics of training data. Image counts are unique images per source and the total is their sum, so COCO images shared by COCO, COCONut-PanCap and GroundedRef are counted once per source.

Source dataset	Num. of images	Num. of instances	Annotation Type	Role
Instance Segmentation Data				
COCO [52]	117.3K	860.0K	Human annotated	Standard benchmark
COCONut (Objects365 subset) [53]	242.6K	3.6M	Human annotated	Standard benchmark
EntitySeg [54]	8.1K	79.1K	Human annotated	Standard benchmark
ADE20K [55]	25.6K	639.5K	Human annotated	Scene-centric objects
Roboflow-VL-100 (81 datasets) [59]	92.9K	812.3K	Converted benchmark data	Diverse domains
PixMo Points [1]	56.8K	1.7M	Auto-constructed	Dense scenes + diverse text queries
SA-1B [58]	71.1K	249.5K	Auto-constructed	Dense scenes + diverse text queries
Caption Grounding Data				
GoldG [11]	75.2K	2.6M	Rewritten + pseudo masks	Human-caption grounding
COCONut-PanCap [50]	100.1K	1.3M	Rewritten + GT masks	Human-caption grounding
GroundedRef	60.2K	230.5K	Synthetic captions + GT masks	Referring and compositional grounding
Total	849.8K	12.0M	–	–

D.1 Caption data

We repurpose three existing datasets into a unified caption-grounding format. GoldG [11] and COCONut-PanCap [50] are built from human-written captions and processed with a shared two-stage LLM rewrite pipeline to clean coreference annotations and rewrite captions for consistent grounding. GoldG is derived from Flickr30k and GQA annotations from the Grounding DINO training corpus, with pseudo masks generated from ground-truth boxes using SAM3 [12], while COCONut-PanCap uses relabeled COCO panoptic captions with ground-truth masks decoded from panoptic annotations. We also repurpose COCO panoptic segmentation to construct GroundedRef, a synthetic dataset of LLM-generated referring and compositional captions paired with ground-truth masks.

Two-stage caption rewrite. GoldG and COCONut-PanCap are both processed with the same LLM-based two-stage rewrite pipeline. In the first stage, *Resolve*, we validate and clean the raw coreference groups. This step removes count-mismatched groups, excludes part mentions from whole-object groups, filters mixed-entity groups, expands groups to include all coreferent mentions, and trims spans to avoid embedded-entity prepositional phrases. In the second stage, *Rewrite*, we minimally rewrite the caption so that only objects represented by annotation groups remain, while preserving actions, spatial relations, and scene context. After rewriting, character spans are reassigned on the rewritten caption. Examples that cannot be rewritten coherently are marked as unrewritable and excluded.

GoldG. Our first source is GoldG, the grounding corpus used to train Grounding DINO. We use the Flickr30k and GQA subsets from the Grounding DINO training corpus. Starting from the original GoldG entity annotations, we first group annotations into coreference sets, then apply the shared two-stage rewrite pipeline to clean the groups and rewrite the caption. Because GoldG provides box-level rather than mask-level supervision, we convert each ground-truth box into a pseudo instance mask using SAM3 [12], and then serialize the result into our unified format. GoldG therefore provides human-written captions and entity spans, refined through LLM-based cleanup, together with pseudo segmentation masks derived from ground-truth boxes rather than human-drawn masks.

Table 10: Statistics of the three image-caption benchmarks we curated or repurposed. Caption length is measured by whitespace-tokenized words per caption. Span multiplicity denotes the number of distinct character spans associated with each mask instance.

Dataset	Images	Captions	Masks	Caption length words/caption	Span multiplicity spans/mask
Flickr30k	779	2,002	6,717	11.01 ± 4.36	1.06 ± 0.26
GroundedRef	1,318	2,025	6,193	10.34 ± 4.20	1.04 ± 0.19
COCONut-PanCap	2,213	2,213	10,821	50.84 ± 16.89	1.68 ± 0.90

COCONut-PanCap. Our second source is COCONut-PanCap, a relabeled version of COCO 2017 validation images with panoptic captions and coreference annotations. Each example contains a caption together with raw coreference groups and panoptic segment metadata. We again apply the shared two-stage rewrite pipeline to clean the groups and rewrite the caption. Unlike GoldG, COCONut-PanCap already provides panoptic segment annotations, so we decode binary masks directly from the RGB-encoded panoptic PNGs and store them as compressed RLE masks. We then convert the processed annotations into the same unified output format.

GroundedRef. Our third source, GroundedRef, is a synthetic caption-grounded dataset constructed from COCO 2017 panoptic segmentation using LLM-generated captions. We generate two styles of captions. The first consists of short referring expressions that target a single instance in images containing multiple objects from the same category, requiring the caption to disambiguate the target from similar distractors. The second consists of compositional queries, in which two grounded sub-queries are joined by "and," each describing one object relative to others in the scene. In both cases, the underlying masks come directly from COCO panoptic segmentation, providing ground-truth segment supervision. The text spans are produced by the LLM and then post-processed with automated validation, including word-boundary correction and deduplication. Compared with GoldG and COCONut-PanCap, GroundedRef introduces more controlled and compositional language while retaining precise mask supervision.

D.2 Instance segmentation data

We train on a heterogeneous collection of instance segmentation data. This includes standard instance segmentation datasets such as COCO [52], ADE20K [55], COCONut [53], and EntitySeg [54], as well as 81 instance-oriented datasets from Roboflow-VL-100 [59], which we convert into a unified instance segmentation format by prompting SAM3 [12] with the bounding boxes. While these datasets provide strong mask supervision, they do not fully cover the diversity of long-tail categories, dense scenes with high object counts, and grounding-style supervision needed for our setting. To complement them, we additionally construct instance segmentation data from PixMo Points [1] and SA-1B [58], whose point annotations, object discovery signals, and dense visual scenes provide scalable supervision for object-centric grounding.

Pixmo. We derive an instance segmentation dataset from the counting subset of PixMo Points, where each image is annotated with object labels and corresponding point locations. Because the original labels include both concrete objects (e.g., car, person) and abstract or non-segmentable concepts (e.g., shadow, reflection), we first filter them with GPT-4o-mini, retaining only labels that refer to physical objects with clear spatial extent. We then normalize verbose labels by removing attribute modifiers, so that, for example, a large red fire truck is mapped to fire truck, yielding canonical category names for downstream processing. To obtain instance masks, we use SAM3 with two complementary prompting strategies. For each image, we generate candidate masks from the normalized category label (concept-prompted) and from each annotated point (point-prompted). We then assign masks to annotations in two passes. In the first pass, each point is matched to the highest-scoring concept-prompted mask that contains it. Any point left unmatched is assigned in a second pass to the highest-scoring point-prompted mask that contains it. Throughout this process, we discard any candidate mask that covers two or more annotated points, since such masks are likely to merge multiple object instances.

SA-1B images. We construct an instance segmentation dataset from a subset of SA-1B (Tars 0–99) by combining vision-language object discovery with SAM3 mask generation under strict point-mask

consistency constraints. We first filter SA-1B images to retain those with high object density (200–500 segments), focusing on scenes with many countable instances. For each selected image, we use GPT-4o-mini in a vision call to enumerate the distinct object categories present in the scene (e.g., dog, car, tree). We then run two independent pipelines on these categories. First, MolmoPoint [3] localizes instances of each category by predicting one point per visible object. Second, SAM3 [12] generates candidate instance masks from text-only prompts, where category names are normalized by removing quantifier prefixes and converting nouns to singular form (e.g., a group of people becomes person). We merge the two outputs using a strict validation procedure. For each object category in an image, we require that (1) the number of predicted points matches the number of generated masks, and (2) there is an exact one-to-one correspondence between points and masks, such that each point lies inside exactly one mask and each mask contains exactly one point. Categories that fail either condition are discarded. This agreement between independently produced point localizations and segmentation masks serves as a strong quality filter, retaining only instances that are spatially well-defined and unambiguous.

E Baseline implementation details

Grounding DINO + SAM. Our first baseline is a two-stage pipeline that combines an open-vocabulary detector with a prompted segmentation model. In the first stage, we use Grounding DINO [11] (`IDEA-Research/grounding-dino-base`) or its successors MM-Grounding DINO [36] and LLMDet [37] to process the input image together with the full caption. Grounding DINO produces a set of detection queries, each associated with a bounding box and a similarity score over caption tokens. We retain detections whose box confidence exceeds 0.25. For each retained detection, we threshold the token-level similarity logits at 0.20 and map the activated tokens back to character offsets in the original caption using the tokenizer offset mapping. Adjacent or overlapping character spans are merged, yielding a set of grounded text spans paired with each detected box. This procedure allows the baseline to recover text-region correspondences directly from the detector outputs without using any external language model or post-hoc span generator. Because Grounding DINO's text encoder has a limited context window, we split long captions exceeding roughly 800 characters (approximately 256 text tokens) into smaller chunks. We split at phrase boundaries using period separators when possible, run detection independently on each chunk, and then remap the predicted character spans back to coordinates in the full caption. The detections from all chunks are merged to form the final set of box–text predictions for the example.

In the second stage, each detected box is passed to SAM3 [12] for instance segmentation. Unless otherwise noted, we use the bounding box alone as the SAM3 prompt. We also support variants that use the recovered grounded text alone or both box and text jointly as prompts, but the default configuration uses box prompts only because it is the most stable and directly follows the standard detector-to-segmentation pipeline.

Florence-2. We also use Florence-2 [43] (`florence-community/Florence-2-large-ft`) as a unified model for both localization and segmentation. For caption-based evaluation, we use the `<CAPTION_TO_PHRASE_GROUNDING>` task, which takes the image and caption as input and predicts a set of bounding boxes paired with phrase labels. Unlike Grounding DINO, Florence-2 does not directly return exact character offsets in the source caption. To obtain text spans comparable to our formulation, we recover character offsets by matching each predicted phrase label back to the original caption. We first perform exact substring matching. If exact matching fails, we apply a normalized token-level matching fallback that ignores minor punctuation and whitespace differences. When multiple detections share the same phrase label, we sort detections by horizontal box center and assign them left-to-right to successive matching occurrences in the caption. To reduce duplicate detections, we apply per-label non-maximum suppression with an IoU threshold of 0.3.

For segmentation, each detected box is passed to Florence-2's `<REGION_TO_SEGMENTATION>` task, which returns polygon coordinates for the segmented region. We convert these polygons into binary masks and encode them as COCO compressed RLE for evaluation. Because Florence-2 represents locations on a discrete 0–999 coordinate grid, bounding box coordinates are quantized to this grid before prompting the region-to-segmentation head. For category-list style inputs, we instead use Florence-2's `<OPEN_VOCABULARY_DETECTION>` task and query one category at a time. When the

query string becomes too long, we split it into smaller category groups; in practice, we chunk queries longer than 2000 characters.

GLaMM. GLaMM is a native grounded-segmentation multimodal model that generates a description interleaved with `<p>phrase</p>[SEG]` markers and emits one segmentation mask per `[SEG]` token, without a separate detector or mask model. Since GLaMM produces its own grounded description rather than grounding a provided caption, we adapt it to our setting by conditioning it on the target caption—prompting it to segment the objects the caption mentions—and matching each generated phrase back to a character span in that caption (exact match, then token-overlap, then the full caption as a fallback) to recover the text grounding; masks are taken directly from GLaMM. For the image-category setting, generated phrases are matched to the provided category list. We use the GLaMM-FullScope checkpoint.

GPT-5.4 + SAM. As a strong proprietary MLLM baseline, we prompt GPT-5.4 to ground each mentioned concept: for the image-caption setting it returns, per concept, the grounded character span(s) in the caption together with a bounding box; for the image-category setting it returns the category name (which we map to its span in the category list) with a bounding box. Each predicted box is then converted to a segmentation mask by SAM3, mirroring the GDINO+SAM pipeline. We run GPT-5.4 with medium reasoning effort.

Qwen3.5-FT. For the Qwen3.5 fine-tuning baseline, we use the same Qwen3.5-0.8B backbone and fine-tune it on our correspondence training data with standard autoregressive supervised fine-tuning. Given an image-text pair, the model is trained to generate a serialized list of correspondences, where each entry contains the grounded character span, and a polygon representation of the corresponding image mask. Specifically, for each ground-truth instance, we obtain its mask boundary as a polygon and pair it with the character spans of the corresponding phrase in the input text. Masks are first converted into polygon contours. To keep the target representation compact, each polygon component is uniformly subsampled to at most 16 points, and all coordinates are normalized to a fixed integer coordinate range $\{0, \ldots, 999\}$. Each target line is formatted as: `<x1 y1 x2 y2 ... x16 y16 | x1 y1 x2 y2 ... x16 y16 ; s e>` where polygon components belonging to the same instance are separated by `<|>`, the semicolon separates visual geometry from textual grounding, and `<s e>` denotes the character span of the grounded mention. Multiple instances are serialized as separate lines following the ground-truth instance order. We use a maximum training sequence length of 8,192 tokens and a 16K-token generation budget during evaluation to accommodate long multi-instance outputs.

This baseline tests whether a pretrained VLM can learn bidirectional correspondence purely through language-generation supervision, without our bridge-token correspondence hypotheses or dedicated text and image segmentation heads. Its strong text scores show that autoregressive fine-tuning can recover grounded mentions reasonably well, but its weaker mask and joint metrics indicate that generating structured masks as text is less effective for precise instance-level correspondence prediction.

F Effects of Annotation Standardization and Model Architecture

We further analyze the effects of annotation standardization and model architecture by training **ConCor-1** and MM-GDINO with either the original source annotations or our standardized annotations. Both models are fine-tuned for 15K steps. Within each model comparison, all other training and evaluation settings are kept unchanged. **ConCor-1** uses its native dynamic-resolution input.

Effect of annotation standardization. Our standardized image-caption annotations recover omitted grounded entities, consolidate coreferential mentions, and align each grounded text span with its corresponding instance mask. This improves image-caption JointF1 by 9.4 points for MM-GDINO-FT and 9.8 points for **ConCor-1**.

Effect of model architecture. With the original annotations, **ConCor-1** outperforms MM-GDINO-FT by 3.8 points on image-caption benchmarks and 11.6 points on image-category benchmarks. With the standardized annotations, the corresponding gains are 4.2 and 11.6 points. The consistent

Table 11: Effects of annotation standardization and model architecture. Image-caption JointF1 is averaged over the three caption benchmarks, while image-category JointF1 is averaged over COCO and LVIS-minival.

Training annotations	Method	Image-caption JointF1	Image-category JointF1
Original source	MM-GDINO-FT + SAM	57.0	33.3
Original source	**ConCor-1**	**60.8**	**44.9**
Standardized	MM-GDINO-FT + SAM	66.4	33.6
Standardized	**ConCor-1**	**70.6**	**45.2**

improvements under both annotation settings show that the architectural advantage is complementary to the gains from annotation standardization.

G Visualizations

Bridge token spatial specialization. We next visualize the activation patterns of bridge tokens from different token groups to understand the effect of our multi-scale spatial assignment. As shown in Fig. 7, bridge tokens in the coarse groups (e.g., 1×1 and 2×2) produce broad activations that cover large regions of the image, whereas tokens in finer groups become progressively more localized and align with smaller spatial cells. This behavior is consistent with the intended design of our bridge tokens: different groups encode different spatial scales, and individual tokens specialize to particular regions within those scales. The visualization suggests that the model learns a structured set of correspondence queries that jointly cover the image from global to local resolution, which helps it handle objects with diverse sizes and positions.

Additional qualitative comparisons. Fig. 8 provides additional qualitative comparisons on COCONut-PanCap. Across diverse scenes with multiple objects and complex captions, GDINO+SAM often detects visually plausible regions but fails to associate them with the correct caption mentions. This is especially common when the image contains multiple instances of the same category or when the referred object is described through contextual relations. In contrast, **ConCor-1** more consistently recovers both the correct text spans and the corresponding image masks, producing more complete and instance-specific text–image alignments.

Additional text-to-image attention visualizations. Fig. 9 shows additional layer-20 text-to-image attention maps for the original Qwen3.5, Qwen3.5 fine-tuned on our dataset, and **ConCor-1**. The original model often attends broadly to background regions or nearby distractor objects, while fine-tuning on our data improves localization but still produces noisy attention in many cases. **ConCor-1** yields more compact and semantically meaningful attention maps, where object words attend to the corresponding visual entities. These examples further support that the bridge-token correspondence objective improves not only the final grounding outputs, but also the internal cross-modal alignment of the VLM.

Additional bridge-token attention visualizations. Fig. 10 presents additional examples of bridge-to-image and bridge-to-text attention, together with the spatial prior and final prediction for each selected bridge token. The visualizations show that different bridge tokens specialize to different object-level hypotheses: each token tends to focus on a localized image region and the corresponding text mentions. The final predictions demonstrate that these bridge-specific attention patterns are effectively transformed into coherent text–mask and image–mask correspondences. This provides further evidence that bridge tokens serve as structured intermediate variables for binding language spans with visual instances.

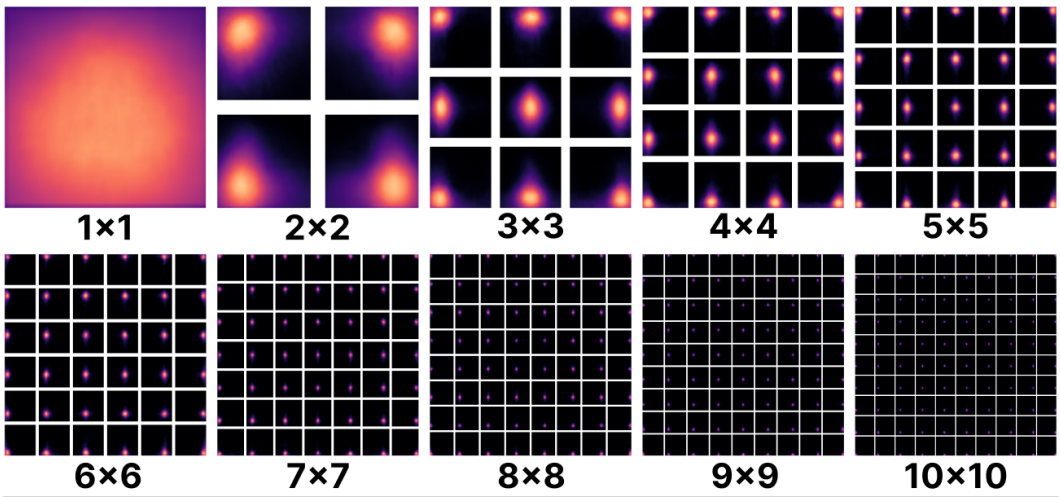


Figure 7: **Spatial specialization of bridge tokens across token groups.** We visualize the spatial activation patterns of bridge tokens from different scale groups under our multi-scale bridge-token design. Each group corresponds to a spatial partition of the image, ranging from a global 1×1 group to increasingly fine grids. Bridge tokens in coarse groups respond to broad image regions, while tokens in finer groups activate on increasingly localized areas aligned with their assigned spatial cells. This shows that bridge tokens learn structured spatial priors and progressively cover the image at multiple resolutions, supporting correspondence prediction for objects of different sizes and locations.



Figure 8: Qualitative visualization examples.

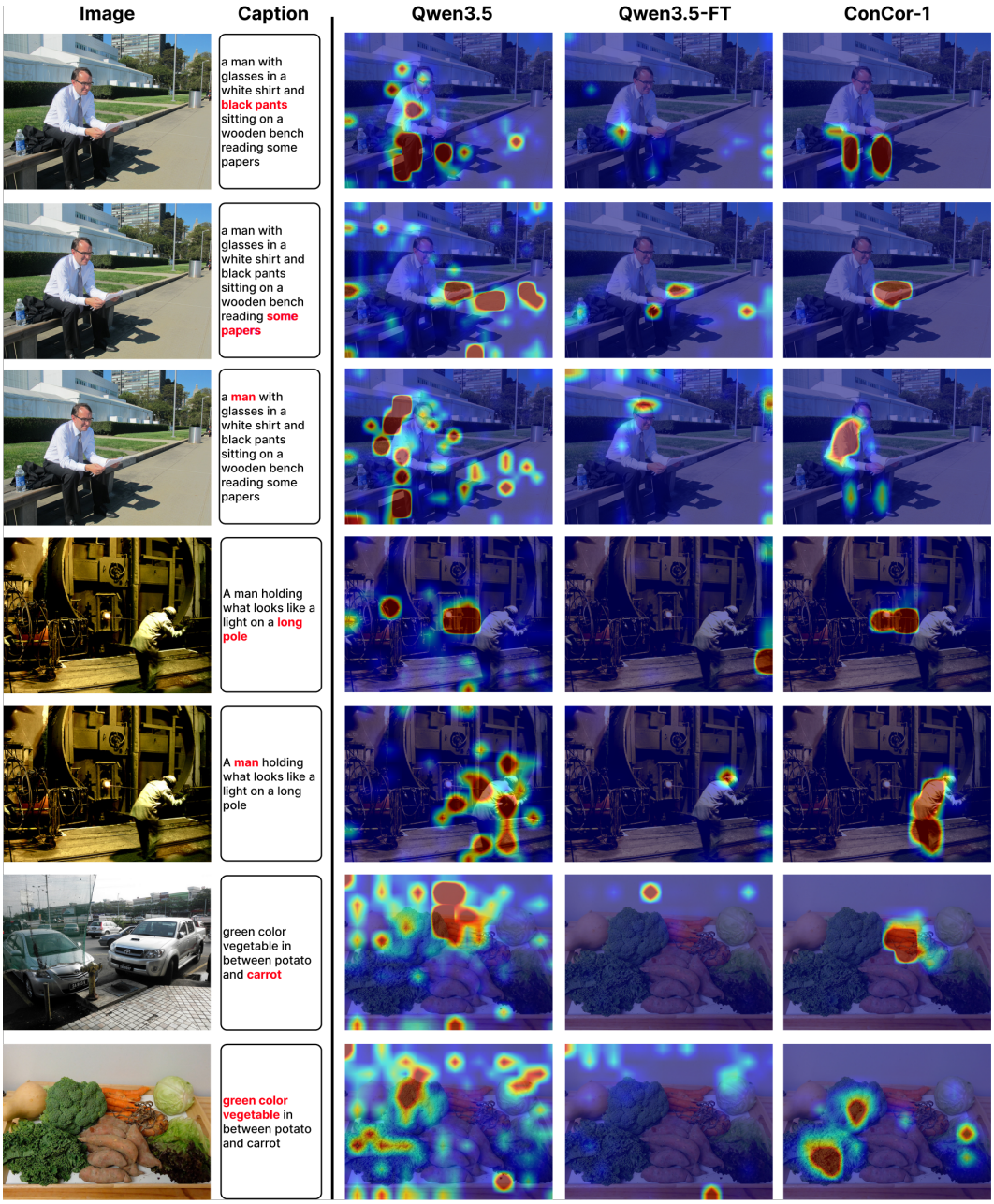
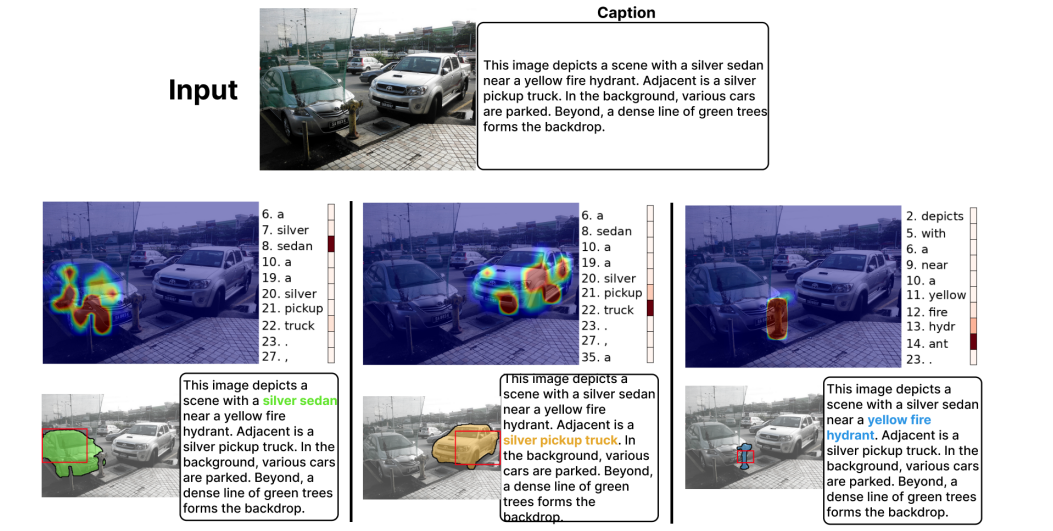
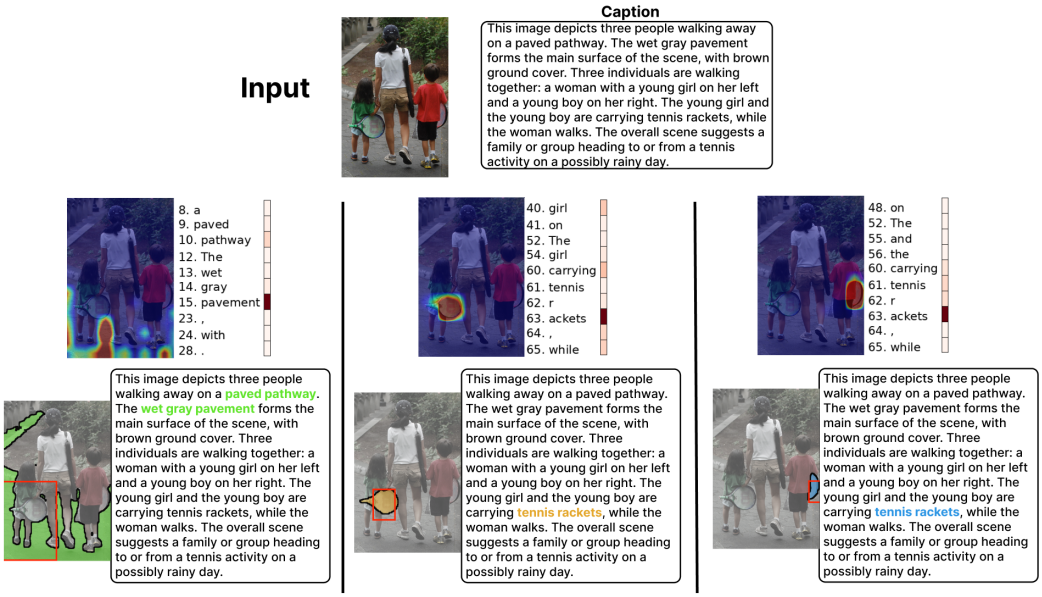


Figure 9: Qualitative visualization examples.





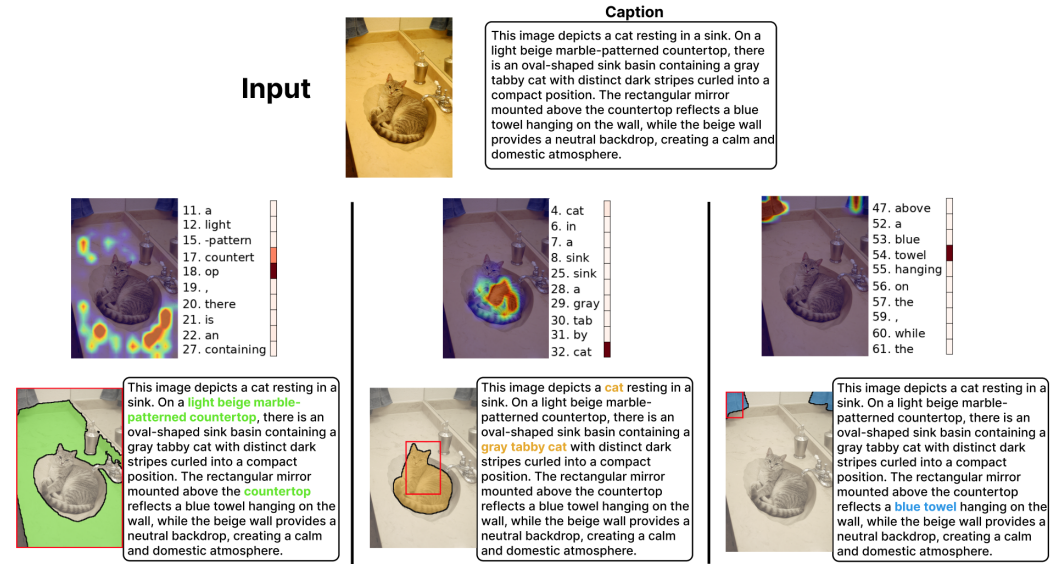


Figure 10: Qualitative visualization examples.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the claims made, including the contributions made in the paper and important assumptions.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper does not include theoretical results.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper reports all the model details, datasets and experimental settings. (Sec.3, Sec.4, Appendix.)

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We will subsequently organize and open-source the model code, as well as the data, and release in a later date.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the experimental details in Sec.4 and Appendix.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not applicable in our experiment, as the model parameters are fixed and we do not perform any sampling.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Appendix. We use 8-GPU A100/H100 to train our model. The workers number is 4.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: N/A

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We only used open-source models for comparative experiments, and used open-source datasets for training and testing. We will provide credits when open-sourcing the code and data.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have detailed the specifics in our paper. In the future, we will provide comprehensive documentation alongside the open-source code.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.