ContourVLA: A Closed-Loop Perception-Action Contour Policy for Generalized Referring Expression Segmentation

Anonymous authors
Paper under double-blind review



Abstract

Generalized referring expression segmentation (GRES) requires dynamically balancing high-level semantics for identifying a variable number of language-specified referents with fine-grained visual evidence for precise boundary delineation. This requirement challenges existing cascaded vision-language architectures, which typically rely on static feature interfaces and single-pass mask prediction, limiting adaptive perception and geometric correction. We introduce ContourVLA, a vision-language-action policy that recasts GRES as a closed-loop visuomotor process, in which editable contours serve as explicit policy states that condition multimodal perception and are updated by geometric action chunks. Evolution-Aware Semantic Scheduling (EASS) couples contour-guided bidirectional boundary sampling with state-conditioned routing of multilevel multimodal features, adapting perception to each contour state. Following supervised initialization, Dustbin-Augmented Entropic Credit Transport GRPO (DECT-GRPO) jointly optimizes discrete grounding and continuous contour actions with instance-level credits. Its rollout rewards and credits are derived from soft prediction-target correspondences that account for false positives and missed targets. ContourVLA improves gIoU over the strongest evaluated baselines by 8.7, 2.8, and 2.7 points on gRefCOCO val, testA, and testB, respectively, and achieves the highest mIoU across all eight RefCOCO, RefCOCO+, and RefCOCOg splits.

1 Introduction

Generalized referring expression segmentation (GRES) delineates a variable number of objects specified by free-form expressions involving attribute, relational, spatial, and numerical constraints (Liu et al., 2023a; Kim et al., 2026). Recent GRES methods typically adopt a cascaded architecture that connects preselected layers of pretrained vision-language models (VLMs) to segmentation heads through static feature interfaces (Lai et al., 2024; Xia et al., 2024). Beyond the demands of conventional fixed-category segmentation, the GRES segmentation head needs to dynamically balance high-level semantics for identifying language-specified referents and fine-grained visual evidence for recovering precise boundaries. However, static cascaded interfaces limit such adaptive feature acquisition (Luo et al., 2025). Moreover, single-pass mask prediction lacks an updatable intermediate geometric state and an explicit feedback path for revisiting local evidence, hindering the refinement of fine structures and the separation of adjacent instances (Peng et al., 2020).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

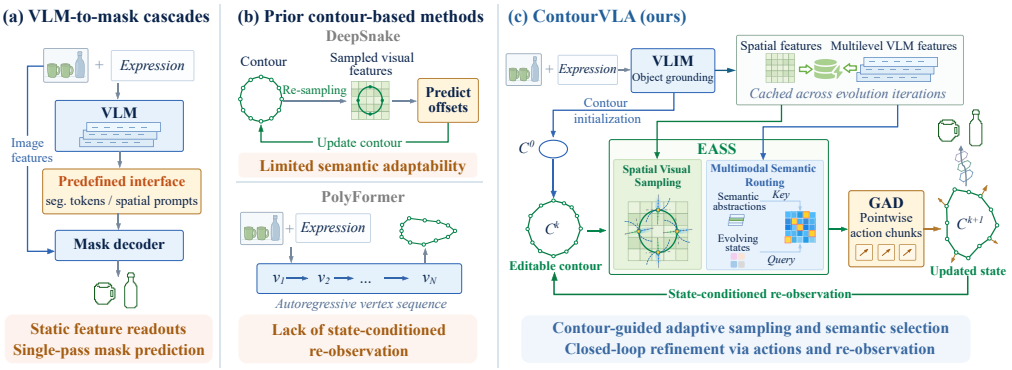

Figure 1: **Paradigm-level comparison of ContourVLA with prior methods.**

These limitations motivate a shift from static feedforward prediction toward an active vision paradigm, in which the current state guides evidence acquisition and action-induced state updates condition subsequent observations (Bajcsy, 1988). Inspired by this view, we formulate GRES as a text-conditioned visuomotor decision process, where an editable contour serves as an explicit geometric state that regulates multimodal perception and evolves through pointwise geometric actions. Figure 1 contrasts this formulation with VLM-to-mask cascades and prior contour-based methods.

We introduce ContourVLA, a vision-language-action (VLA) policy (Kim et al., 2024) that formulates GRES as a closed-loop perception-action process, where multimodal observations are iteratively transformed into geometric actions for contour evolution. ContourVLA consists of a V-L Interaction Module (VLIM) and a Geometric Action Decoder (GAD). VLIM acts as a perceptual expert that grounds text-specified referents and provides multimodal observations for state-conditioned perception, while GAD serves as an action expert that predicts geometric action chunks from these observations (Zhao et al., 2023). Executing each chunk updates the contour state and triggers the next perception–action cycle, progressively refining the contour from coarse localization to precise boundaries. ContourVLA adopts a two-stage training strategy: SFT establishes grounding and contour-trajectory priors, followed by joint GRPO post-training with self-rollout feedback (Shao et al., 2024). This formulation raises two key challenges: adaptive multimodal perception under evolving contour states and effective optimization of the joint policy.

Geometric refinement requires adaptive integration of spatial and semantic evidence throughout contour evolution, a dynamic demand that static feature interfaces cannot accommodate. We therefore introduce Evolution-Aware Semantic Scheduling (EASS), which adapts dual-axis multimodal perception through the evolving contour state. On the spatial axis, Spatial Visual Sampling uses contour geometry as a prior to sample bidirectional boundary cues over an adaptive receptive field. On the semantic axis, Multimodal Semantic Routing encodes instance geometry, evolution progress, contour motion, and decoding depth as queries to select the most relevant VLIM representations.

Effective joint-policy optimization in GRES requires multi-instance credit attribution under uneven prediction quality, false positives, and missed targets. Standard GRPO instead broadcasts a single rollout-level advantage across all instances, conflating their contributions and preventing targeted optimization (Shao et al., 2024). We therefore propose Dustbin-Augmented Entropic Credit Transport GRPO (DECT-GRPO), which assigns credit via soft correspondences established by entropic transport, while dustbin entries absorb false-positive predictions and missed targets (Cuturi, 2013; Sarlin et al., 2020). DECT-GRPO preserves module-specific likelihoods for discrete grounding and continuous contour actions, while converting transport-weighted segmentation quality into branch-wise centered credit maps assigned to corresponding grounding tokens and contour actions.

Our main contributions are as follows:

- We introduce ContourVLA, the first to recast GRES as an active perception-action contour policy in which geometric states bridge multimodal perception and contour actions.
- We propose EASS, a state-conditioned perceptual scheduler that replaces static readouts with spatial visual sampling for adaptive boundary-cue acquisition and multimodal semantic routing for evolution-aware selection of multilevel multimodal features.
- We propose DECT-GRPO, which couples module-specific likelihood optimization with dustbin-augmented entropic transport to derive instance-level credits for grounding tokens and contour actions while accounting for false-positive predictions and missed targets.

2 Related Work

Generalized Referring Expression Segmentation. Generalized referring expression segmentation (GRES) aims to identify and segment a variable number of objects specified by free-form expressions (Liu et al., 2023a). Early approaches rely on cross-modal fusion between separate visual and language encoders (Yang et al., 2022; Liu et al., 2023a; Luo et al., 2025), while recent methods typically connect pretrained VLMs with segmentation decoders through segmentation tokens or spatial prompts (Lai et al., 2024; Xia et al., 2024; Rasheed et al., 2024; Liu et al., 2025a; Huang et al., 2025; Lu et al., 2026). Despite varied interfaces, these cascades rely on fixed feature readouts and single-pass prediction, limiting adaptive semantic-spatial evidence selection and iterative segmentation correction. In contrast, ContourVLA formulates GRES as a closed-loop visuomotor process, where the evolving contour state regulates visual evidence acquisition and semantic abstraction selection, while iterative geometric actions enable continuous boundary refinement.

Contour Evolution and Vision-Language-Action Policies. Contour-based methods represent boundaries as ordered vertices and refine them through geometric updates. DeepSnake and its variants (Peng et al., 2020; Zhang et al., 2025a;b) iteratively sample local visual features for contour refinement, while PolyFormer (Liu et al., 2023b) autoregressively predicts language-conditioned contour vertices. These methods do not adapt multimodal evidence to evolving contour states in GRES. Inspired by vision-language-action models (Kim et al., 2024), ContourVLA instead recasts contour evolution as a 2D geometric action prediction problem conditioned on vision and language, and solves it through a closed-loop perception-action process.

Reinforcement Learning for Vision-Language Reasoning and Grounding. Reinforcement Learning (RL) has emerged as a potent methodology for enhancing the capabilities of VLMs on complex visual tasks. Seg-Zero (Liu et al., 2025a) and SAM-R1 (Huang et al., 2025) apply GRPO (Shao et al., 2024) with localization or mask-quality rewards to multimodal segmentation and grounding. Standard GRPO, however, shares a rollout-level advantage across all outputs without explicitly distinguishing instance quality in the advantage assigned to each prediction. MCR-GRPO (Han et al., 2026) estimates instance importance through marginal contribution analysis. DECT-GRPO instead derives instance-level credit assignments via dustbin-augmented entropy-regularized transport, with dustbin assignments absorbing unmatched predictions and targets.

3 Method

As shown in Figure 2, ContourVLA formulates GRES as a text-conditioned visuomotor decision process, where editable contours serve as geometric policy states that condition perception and action. Executing action chunks iteratively updates the contour state and closes the perception-action loop (Section 3.1). Within this closed loop, EASS constructs state-conditioned multimodal observations by sampling around the contour and scheduling semantic features (Section 3.2). GAD uses these observations to predict geometric action chunks that update the contour state (Section 3.3). Training follows a two-stage strategy: supervised initialization establishes a geometric prior, followed by DECT-GRPO post-training with instance-level credit assignment (Section 3.4).

3.1 State-Conditioned Perception-Action Loop

Multimodal perception and contour initialization. Given an image-expression pair (I, x), VLIM autoregressively generates a structured grounding sequence $y_{1:L_y} \sim \pi_\theta^y(\cdot \mid I, x)$, which contains reasoning tokens followed by either M box records or a no-target response:

$$y_{1:L_y} = \begin{cases} r_{1:L_r}[\langle \text{box} \rangle \langle x_1 \rangle \langle y_1 \rangle \langle x_2 \rangle \langle y_2 \rangle]_{i=1}^{M} \langle \text{eos} \rangle, & M > 0, \\ r_{1:L_r} \langle \text{no} - \text{target} \rangle \langle \text{eos} \rangle, & M = 0. \end{cases} \tag{1}$$

The coordinate tokens in each box record are decoded into a bounding box b_i, which initializes an inscribed elliptical contour $C_i^0 = (c_{i,1}^0, \ldots, c_{i,P_i}^0) \in \mathbb{R}^{P_i \times 2}$. The point budget P_i is determined by the box perimeter to adapt contour resolution to object scale. Meanwhile, VLIM caches a visual feature grid F and multilevel multimodal representations $\{H^\ell\}_{\ell \in \mathcal{L}}$ for subsequent state-conditioned perception. Region features pooled from F are further contextualized with box geometry to obtain the instance representation V_i.

Perception-action loop. At evolution iteration k, EASS constructs state-conditioned observations along spatial and semantic axes. The spatial branch extracts contour-conditioned boundary evidence

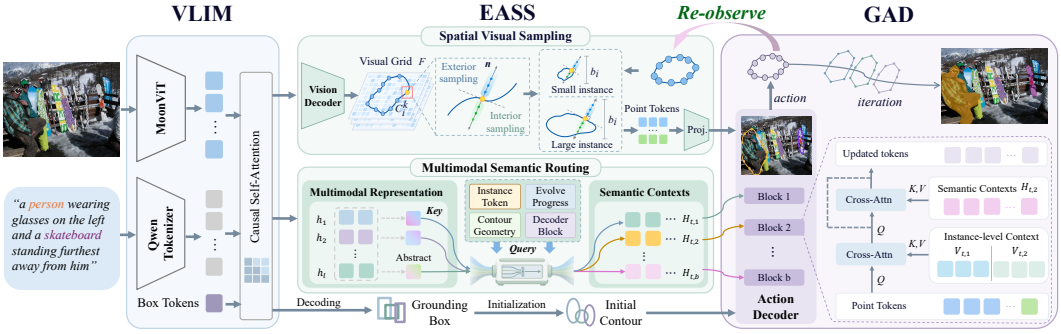

Figure 2: **Overview of the ContourVLA framework.** VLIM predicts grounding boxes for contour initialization and provides visual and multilevel multimodal features. EASS performs adaptive bidirectional boundary sampling and routes semantic representations according to the evolving state. GAD combines contour point tokens with instance context and routed semantics to predict pointwise geometric action chunks. Executing these actions updates the contours and conditions subsequent observations, forming a closed perception-action loop. The dashed inset illustrates a GAD block.

$\Phi(F, C_i^k, b_i)$ from the visual grid, while the semantic branch selects block-specific multimodal features $\mathcal{U}_i^k$ from $\{H^\ell\}_{\ell\in\mathcal{L}}$. Together with the instance representation V_i, these readouts constitute the multimodal observation O_i^k for geometric action chunk prediction A_i^k in GAD:

$$O_i^k = \left(\Phi(F, C_i^k, b_i), V_i, \mathcal{U}_i^k\right), \qquad A_i^k = \pi_\phi(C_i^k, O_i^k), \qquad C_i^{k+1} = \mathcal{T}(C_i^k, A_i^k). \tag{2}$$

Here $\mathcal{T}$ denotes the geometric state transition function. The updated contour C_i^{k+1} initializes the next perception step. Thus, the contour serves as an explicit policy state connecting iterative observation, geometric refinement, and state evolution.

3.2 Evolution-Aware Semantic Scheduling

The spatial evidence and semantic abstractions required for geometric refinement vary across evolution stages and contour states. We therefore introduce Evolution-Aware Semantic Scheduling (EASS), a dual-axis scheduler that adapts visual evidence acquisition and multimodal feature selection throughout evolution.

3.2.1 Spatial Visual Sampling

At each evolution iteration, the current contour provides a geometric prior for acquiring local visual evidence. Beyond vertex-level feature sampling (Peng et al., 2020), EASS dynamically expands the receptive field along each vertex normal by sampling bidirectional interior and exterior features from visual grid F. Let $b_i^{\text{size}} = (w_i, h_i)$ denote the size of box b_i, and let $n_{i,p}^k$ be the unit normal computed after mapping the contour into box-normalized coordinates. Using nonnegative offsets $\{o_j\}_{j=1}^J$, the contour-aligned feature profile is

$$\phi_{i,p}^k = \text{Concat}_{j=1}^J \left[F\left(c_{i,p}^k - o_j(n_{i,p}^k \odot b_i^{\text{size}})\right), F\left(c_{i,p}^k + o_j(n_{i,p}^k \odot b_i^{\text{size}})\right)\right]. \tag{3}$$

Bidirectional sampling captures complementary interior and exterior boundary cues, while box-relative scaling adapts the receptive field to object scale. Details are provided in Appendix A.5.

3.2.2 Multimodal Semantic Routing

Semantic abstractions as keys. The multilevel multimodal features of VLIM constitute a candidate semantic space. For each layer ℓ, let H^ℓ denote its multimodal token representations. EASS compresses H^ℓ into a key that characterizes the semantic abstraction provided by each layer:

$$\kappa_\ell = \text{Norm}_2\left(f_{\text{key}}\left(\text{Pool}(H^\ell) + e_\ell\right)\right), \tag{4}$$

where $\text{Norm}_2(z) = z/\|z\|_2$ and e_ℓ is a learned layer identity.

Evolving states as queries. Instance characteristics g_i, contour states c_i^k, evolution stages r^k, and decoder depths d jointly determine the semantic requirements of each refinement step. EASS encodes these factors into queries to select the VLIM layer that provides the most relevant semantic cues. For object i at evolution iteration k and GAD block d, the query is constructed as

$$q_i^{k,d} = \text{Norm}_2\left(f_q(g_i, c_i^k, r^k, d)\right). \tag{5}$$

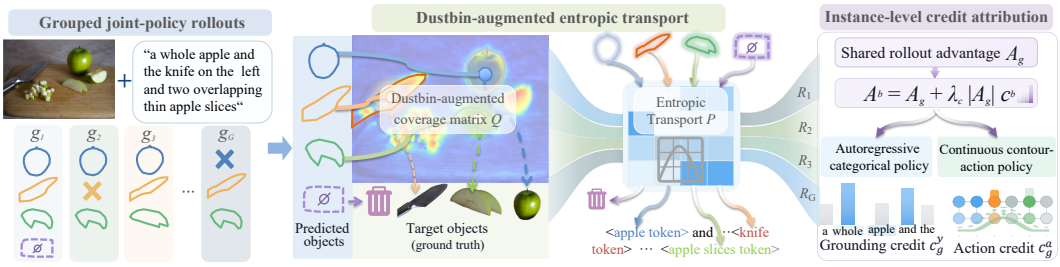

Figure 3: **Overview of DECT-GRPO.** Dustbin-augmented entropic transport establishes soft prediction–target correspondences and assigns instance-level credits to grounding tokens and contour actions for joint policy optimization.

Here $g_i = (b_i, P_i)$ denotes the instance characteristics, where $b_i = (x_i^1, y_i^1, x_i^2, y_i^2)$ is the normalized grounding box and P_i is the object-adaptive contour point budget.

Evolution-aware key-query routing. EASS computes the compatibility between each evolution-conditioned query and VLIM semantic keys, yielding a probability distribution over candidate layers:

$$p_{i,\ell}^{k,d} = \mathrm{Softmax}_{\ell \in \mathcal{L}} \left(\langle q_i^{k,d}, \kappa_\ell \rangle + \alpha_u B_{k,d,\ell} \right), \tag{6}$$

Here $p_{i,\ell}^{k,d}$ denotes the layer-selection probability for VLIM layer ℓ of object i at evolution iteration k and GAD block d. To stabilize early routing, the prior B initializes an ordered correspondence between evolution-depth slots and VLIM layers, while its weight gradually decays to allow contour-conditioned query-key compatibility to dominate. To retain discrete layer selection in the forward pass while enabling differentiable router optimization, EASS applies hard routing with a straight-through estimator:

$$\widetilde{p}_i^{k,d} = \mathrm{sg}\big(\mathrm{onehot}(\arg\max_\ell p_{i,\ell}^{k,d}) - p_i^{k,d}\big) + p_i^{k,d}, \qquad U_i^{k,d} = \mathrm{Proj}\big(\sum_{\ell \in \mathcal{L}} \widetilde{p}_{i,\ell}^{k,d} H^\ell\big), \tag{7}$$

where sg denotes stop-gradient and $U_i^{k,d}$ is the routed multimodal feature delivered to the d-th GAD block. Proj projects the selected VLIM representation into the corresponding GAD feature space. Recomputing the query after each contour update adapts the routed semantic abstraction throughout multistep contour evolution. Further details are provided in Appendix A.4.

3.3 Geometric Action Chunking

GAD decodes EASS observations into continuous pointwise displacement actions. Each contour vertex is represented as a geometric token based on the contour-conditioned visual evidence $\phi_{i,p}^k$ from Spatial Visual Sampling and contour geometry:

$$Z_{i,p}^{k,0} = \mathrm{Proj}\left(\mathrm{Concat}[\phi_{i,p}^k, c_{i,p}^k, \mathrm{PE}_{\mathrm{box}}(c_{i,p}^k; b_i), \mathrm{PE}_{\mathrm{cyc}}(p/P_i)]\right). \tag{8}$$

where $\mathrm{PE}_{\mathrm{box}}(c_{i,p}^k; b_i)$ encodes box-relative vertex coordinates, and $\mathrm{PE}_{\mathrm{cyc}}(p/P_i)$ preserves the cyclic positions of vertices along the closed contour. Stacking the P_i point tokens yields the contour representation $Z_i^{k,0} = [Z_{i,1}^{k,0}, \ldots, Z_{i,P_i}^{k,0}]^\top \in \mathbb{R}^{P_i \times h}$, which is further refined with instance-level context V_i. At each GAD block, contour tokens interact with the semantic representation $U_i^{k,d}$:

$$\overline{Z}_i^{k,d} = \mathrm{CrossAttn}_d(Z_i^{k,d-1}, V_i), \qquad Z_i^{k,d} = \overline{Z}_i^{k,d} + \mathrm{CrossAttn}_d(\overline{Z}_i^{k,d}, U_i^{k,d}). \tag{9}$$

The action decoder then predicts two-dimensional displacement means for each vertex in parallel (Zhao et al., 2023), which updates the contour state and triggers the next iteration.

3.4 Policy Optimization with DECT-GRPO

We initialize VLIM with teacher-forced grounding and GAD with positional supervision along reference contour trajectories, regularized by auxiliary geometric and routing objectives (Appendix A.6). Building on this initialization, DECT-GRPO (Fig. 3) leverages dustbin-augmented entropic transport to derive instance-level credit attribution and transport-based rewards for jointly optimizing grounding and geometric actions.

Grouped joint-policy rollouts. For each input (I, x), VLIM samples G grounding outputs that define the initial contours, after which GAD generates continuous geometric action chunks. The grounding branch follows an autoregressive categorical policy, while the action branch follows a bounded continuous policy based on a $\tanh$-transformed Gaussian. Their likelihood definitions and rollout-replay details are provided in Appendix A.7.

Dustbin-augmented entropic transport. Instances within a GRES rollout may differ substantially in segmentation quality, yet standard GRPO assigns all outputs the same rollout-level advantage without explicitly distinguishing instance quality in the assigned advantage (Shao et al., 2024). DECT-GRPO therefore uses dustbin-augmented entropic transport to establish soft prediction-target correspondences for reward construction and instance-level credit attribution, with virtual null targets and predictions absorbing false positives and missed targets.

For rollout g among the G samples of an input, let M_g and N denote the predicted and target instance counts, respectively. With $Q_{g,ij}$ denoting the mask IoU between prediction i and target j, the pairwise coverage matrix $Q_g = [Q_{g,ij}] \in \mathbb{R}^{M_g \times N}$ is augmented as

$$\overline{Q}_g = \begin{bmatrix} Q_g & b_\varnothing \mathbf{1}_{M_g} \\ b_\varnothing \mathbf{1}_N^\top & b_\varnothing \end{bmatrix} \in \mathbb{R}^{(M_g+1)\times(N+1)}, \tag{10}$$

where $b_\varnothing$ is the null-match score. The last column absorbs unmatched predictions, while the last row accounts for missed targets. Let $P \in \mathbb{R}^{(M_g+1)\times(N+1)}$ denote the soft transport plan, where P_{ij} is the matching mass assigned between prediction i and target j, including dustbin matches. DECT-GRPO obtains the globally consistent soft assignment through entropy-regularized optimal transport:

$$P_g^\star = \arg\max_{P \in \mathcal{U}_g} \left[\sum_{i,j} P_{ij} \overline{Q}_{g,ij} - \varepsilon \sum_{i,j} P_{ij} \log P_{ij} \right]. \tag{11}$$

Here $\mathcal{U}_g$ is the set of nonnegative transport plans satisfying the prescribed prediction and target marginals. The entropy term prevents ambiguous overlaps from collapsing prematurely to hard correspondences, preserving uncertainty among plausible matches and yielding smoother instance-level credit. The marginal constraints enforce global consistency across predictions and targets, so $P^\star_{g,ij}$ represents a globally normalized soft correspondence rather than an independent pairwise score. Mass assigned to the null target denotes unmatched predictions and lowers instance credit, whereas mass from the null prediction denotes missed targets and reduces target coverage. Its Sinkhorn solution is derived in Appendix A.8.

Transport-based reward. A reliable rollout reward should jointly capture contour accuracy, target coverage, and prediction-target correspondence. DECT-GRPO therefore evaluates segmentation quality from complementary prediction and target perspectives under the soft transport plan $P_g^\star$:

$$\mathrm{Prec}_g = \frac{1}{M_g} \sum_{i=1}^{M_g} \frac{\sum_{j=1}^{N} P^\star_{g,ij} Q_{g,ij}}{\sum_{j=1}^{N+1} P^\star_{g,ij}}, \quad \mathrm{Rec}_g = \frac{1}{N} \sum_{j=1}^{N} \frac{\sum_{i=1}^{M_g} P^\star_{g,ij} Q_{g,ij}}{\sum_{i=1}^{M_g+1} P^\star_{g,ij}}, \quad R_g = \frac{2\,\mathrm{Prec}_g\,\mathrm{Rec}_g}{\mathrm{Prec}_g + \mathrm{Rec}_g} \tag{12}$$

Here, Prec_g and Rec_g denote transport-aware precision and recall under $P_g^\star$. We set $R_g = 0$ when $\mathrm{Prec}_g + \mathrm{Rec}_g = 0$; empty-set cases are handled separately in Appendix A.9. Soft transport stabilizes the reward under ambiguous correspondences by avoiding discontinuities from hard matching. Dustbin mass remains in the normalization but receives no overlap credit; therefore, unmatched predictions reduce Prec_g, while missed targets reduce Rec_g. The resulting harmonic mean jointly captures contour quality, correspondence uncertainty, and target coverage.

Instance-level advantage modulation. Predictions within a rollout may differ in segmentation quality, whereas a shared rollout advantage applies the same scalar feedback to all outputs. DECT-GRPO uses transport-derived instance credits to modulate this shared advantage at the corresponding grounding-token spans and contour-action positions, introducing instance-dependent feedback while retaining the rollout-level signal.

For each predicted instance i, DECT-GRPO computes transport-weighted credit and broadcasts its normalized value to the corresponding grounding-token span $S_{g,i}$ and contour-action indices $\mathcal{D}_{g,i}$.

Branch-wise centering and scaling produce the credit maps:

$$e_{g,i} = \frac{\sum_{j=1}^{N} P^{\star}_{g,ij} Q_{g,ij}}{\sum_{j=1}^{N+1} P^{\star}_{g,ij}}, \qquad \mathbf{c}^{y}_{g}[S_{g,i}] = \mathrm{Norm}_{y}(e_{g,i}), \qquad \mathbf{c}^{a}_{g}[\mathcal{D}_{g,i}] = \mathrm{Norm}_{a}(e_{g,i}). \tag{13}$$

The normalization is performed over branch outputs, yielding zero-centered credit maps that preserve the branch-wise mean advantage. The resulting output-specific advantages are

$$A^{b}_{g,\nu} = A_g + \lambda_c |A_g| \, \mathbf{c}^{b}_{g}[\nu], \qquad b \in \{y, a\}, \tag{14}$$

Answer tokens outside instance spans retain the shared rollout advantage A_g. The output-count-weighted normalization preserves the branch-wise mean advantage. Since $\mathbf{c}^{b}_{g}[\nu] \in [-1, 1]$ and we use $\lambda_c = 0.5$, the instance-modulated advantages retain the sign of A_g. Higher-credit instances receive larger positive advantages when $A_g > 0$ and smaller-magnitude negative advantages when $A_g < 0$. Thus, DECT-GRPO provides bounded, mean-preserving modulation of shared rollout feedback rather than independent instance-level reward or penalty decisions.

Joint policy optimization. Let $\rho^{y}_{g,n}$ and $\rho^{a}_{g,\xi}$ denote the current-to-behavior likelihood ratios for grounding tokens and contour actions, respectively. The joint objective is

$$\mathcal{L}_{\mathrm{DECT}} = \frac{1}{G} \sum_{g=1}^{G} \left[-\frac{1}{L_g} \sum_{n=1}^{L_g} \mathcal{S}_{\mathrm{clip}}(\rho^{y}_{g,n}, A^{y}_{g,n}) - \frac{\lambda_a}{|\mathcal{D}_g|} \sum_{\xi \in \mathcal{D}_g} \mathcal{S}_{\mathrm{clip}}(\rho^{a}_{g,\xi}, A^{a}_{g,\xi}) + \beta_{\mathrm{KL}} \mathcal{K}_g \right] \tag{15}$$

Here λ_a balances the two branches, and $\mathcal{K}_g$ regularizes the grounding policy toward the reference model. Likelihood ratios, the clipped surrogate, and replay details are given in Appendix A.9.

4 Experiments

4.1 Experimental Setup

Datasets and metrics. We evaluate GRES on gRefCOCO (Liu et al., 2023a) and conventional referring expression segmentation (RES) on RefCOCO, RefCOCO+ (Yu et al., 2016), and RefCOCOg (Mao et al., 2016). We report gIoU and cIoU for GRES, and mIoU and cIoU for RES.

Training setup. ContourVLA initializes its visual and language backbones from MoonViT and Qwen2.5-3B-Instruct, respectively. Grounding training on LocateAnything-Data (Wang et al., 2026) is followed by SFT and DECT-GRPO jointly on the training sets of gRefCOCO, RefCOCO, RefCOCO+, and RefCOCOg. A single resulting checkpoint is evaluated across all benchmark splits. Training and evaluation details are provided in Appendix A.1.

4.2 Comparison with Existing Methods

Table 1 compares ContourVLA with existing methods on GRES and conventional RES benchmarks under their respective model and training configurations. All results are obtained from our own evaluation runs, while component-level contributions are examined through the ablations in Section 4.3.

GRES results. On gRefCOCO, ContourVLA achieves 83.5, 80.3, and 74.4 gIoU on val, testA, and testB, respectively, exceeding the strongest evaluated baseline on each split by 8.7, 2.8, and 2.7 points. It also achieves the highest cIoU on val and testA, while its testB cIoU of 71.8 is just 0.1 point below HiMTok. These results combine consistent gains in expression-averaged quality with competitive cumulative pixel overlap, rather than showing an advantage under only one aggregation criterion. Since gIoU also accounts for no-target expressions, Section 4.3 further examines no-target accuracy and instance F1 to characterize rejection and instance-level segmentation more directly.

RES results. On conventional RES benchmarks, the same jointly trained checkpoint achieves the highest mIoU on all eight evaluated splits. Improvements over the strongest baselines range from 1.3 to 1.8 points on RefCOCO and from 2.8 to 3.2 points on RefCOCO+, with additional gains of 0.7 and 0.3 points on RefCOCOg val and test, respectively. These results extend the evidence beyond the generalized setting: the shared contour policy also performs well on conventional referring segmentation. Importantly, all benchmarks use the same checkpoint without further benchmark-specific fine-tuning, supporting the applicability of one policy to both task formulations under the joint-training protocol.

Table 1: Comparison with methods across GRES and RES benchmarks (%). Red, orange, gold, and yellow shading denote the first- through fourth-best scores in each column, respectively.

Method	gRefCOCO						RefCOCO						RefCOCO+						RefCOCOg			
	val		testA		testB		val		testA		testB		val		testA		testB		val		test	
	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU	mIoU	cIoU	mIoU	cIoU	mIoU	cIoU	mIoU	cIoU	mIoU	cIoU	mIoU	cIoU	mIoU	cIoU	mIoU	cIoU
Grounded-SAM (Ren et al., 2024)	46.0	44.1	50.8	48.7	43.5	41.3	46.2	44.4	48.6	47.0	42.9	40.9	38.2	36.5	42.4	40.5	33.0	30.8	44.1	42.0	44.5	43.0
LISA-7B (Lai et al., 2024)	61.4	61.7	66.0	68.5	58.8	60.6	68.1	74.6	70.6	79.3	64.5	72.3	54.8	65.1	60.3	70.8	48.4	58.1	63.1	67.9	66.2	70.6
LISA-13B (Lai et al., 2024)	63.4	62.9	68.1	69.6	61.8	62.2	75.3	76.0	77.8	78.9	71.7	72.9	64.9	65.0	70.1	70.1	58.9	58.1	65.0	69.6	66.0	70.4
PerceptionGPT-7B (Pi et al., 2024)	61.5	59.9	66.1	64.3	58.9	56.8	70.7	74.7	73.9	78.7	66.8	71.7	64.0	68.1	69.4	73.8	56.8	61.0	65.5	70.2	67.2	71.5
PerceptionGPT-13B (Pi et al., 2024)	62.2	60.4	66.9	64.6	59.0	57.4	70.8	75.7	74.5	79.4	67.9	72.0	64.4	69.2	69.8	74.0	57.6	62.0	66.4	71.1	67.4	72.0
LaSagnA-7B (Wei et al., 2024)	32.4	38.1	47.3	50.4	38.9	42.1	72.8	76.8	74.2	78.8	70.7	73.8	61.4	66.4	65.3	70.6	57.3	60.1	66.1	70.5	67.3	72.0
OMG-LLaVA-7B (Zhang et al., 2024a)	36.1	39.3	50.1	52.4	42.2	43.7	73.5	77.8	75.7	80.3	69.3	74.1	64.6	69.3	68.6	73.0	58.8	63.0	68.1	73.0	68.2	73.2
GroundHog-7B (Zhang et al., 2024b)	66.7	64.8	71.4	71.4	64.0	62.7	74.0	78.3	75.4	80.1	71.6	75.7	66.2	70.2	70.6	75.0	60.6	64.8	69.6	74.2	69.8	74.6
GLaMM-7B (Rasheed et al., 2024)	68.7	66.9	73.5	71.6	66.2	64.1	75.0	79.4	78.8	83.5	72.7	76.9	67.8	72.7	74.1	78.7	60.2	64.5	69.8	74.6	70.6	74.7
UniRES-7B (Wang et al., 2024)	74.4	69.9	76.0	74.5	69.8	66.6	75.8	80.1	77.2	82.0	71.7	75.8	66.9	71.8	71.8	76.0	60.0	64.4	69.0	73.7	69.5	74.1
GSVA-7B (Xia et al., 2024)	66.5	63.3	71.1	69.9	61.9	60.5	72.1	76.7	74.3	79.6	68.0	73.0	64.0	68.2	69.1	73.1	58.5	62.3	61.9	66.6	62.4	67.5
GSVA-13B (Xia et al., 2024)	67.6	64.0	71.8	70.5	63.8	61.3	73.5	77.9	75.7	80.6	69.9	74.6	65.7	69.9	70.9	74.9	59.7	63.6	63.2	68.2	63.5	69.0
SAM4MLLM-7B (Chen et al., 2024)	68.9	67.0	73.6	71.8	66.3	64.2	75.4	79.7	78.1	82.8	71.8	75.9	69.1	73.3	73.4	77.8	61.1	66.0	70.0	74.8	71.1	75.7
SegLLM-7B (Wang et al., 2025b)	69.8	67.9	74.6	72.5	67.4	65.2	75.7	80.0	77.0	81.6	71.0	75.1	66.1	70.1	68.5	72.9	57.9	62.6	68.0	72.8	69.3	73.5
UniPixel-7B (Liu et al., 2026)	70.6	68.8	75.3	73.4	68.0	65.9	76.3	80.7	78.5	83.1	73.0	77.3	70.7	75.5	75.6	78.4	65.5	69.7	72.1	76.2	72.4	77.1
UFO-8B (Tang et al., 2026)	71.0	69.1	75.6	73.8	68.4	66.3	76.5	81.1	77.9	82.6	73.7	78.6	72.8	77.1	75.9	80.3	67.9	**72.7**	72.6	76.7	73.1	77.3
LIRA-8B (Li et al., 2025b)	36.7	40.9	50.4	52.4	42.4	44.9	77.3	81.4	78.9	82.4	73.5	77.9	71.8	76.1	76.6	80.8	66.0	70.6	73.8	78.5	73.5	78.3
HiMTok-8B (Wang et al., 2025a)	72.1	70.4	73.4	74.9	71.7	**71.9**	80.0	**85.9**	80.7	81.7	78.6	**83.9**	74.1	**80.5**	76.8	**83.7**	70.0	72.4	75.9	**80.1**	75.9	**78.9**
RAS-13B (Cao et al., 2025)	74.6	70.5	77.5	77.0	69.4	67.8	76.5	81.1	79.1	**83.8**	74.5	78.8	70.4	75.4	75.5	79.7	65.7	70.5	71.1	76.0	73.2	77.6
Seg-R1-7B (You & Wu, 2025)	60.6	58.5	65.3	63.1	57.7	55.8	69.9	74.3	74.2	78.8	63.3	67.5	58.1	63.0	66.3	71.0	53.1	57.9	66.9	71.2	67.1	71.2
Seg-Zero-7B (Liu et al., 2025a)	16.5	26.3	44.7	45.7	40.2	41.3	78.7	76.4	80.7	80.3	75.6	72.5	73.1	69.4	77.2	76.2	67.1	62.7	75.1	72.9	75.5	72.6
VisionReasoner-7B (Liu et al., 2025b)	27.0	39.2	58.3	65.1	48.6	51.8	76.7	73.4	78.9	77.6	72.3	67.6	69.6	63.8	74.9	70.9	61.9	55.5	71.6	67.7	71.3	68.4
SAM-R1-7B (Huang et al., 2025)	68.5	66.4	73.1	71.2	65.6	63.8	72.4	76.3	74.8	79.5	68.7	72.9	65.0	69.8	70.2	74.8	59.3	63.4	75.1	72.2	75.4	73.3
Text4Seg-7B (Lan et al., 2025)	73.6	67.9	74.1	72.8	66.0	64.8	74.5	79.1	77.3	82.2	71.9	76.0	67.6	72.3	73.1	77.3	61.7	66.1	67.3	72.1	69.4	73.7
Text4Seg-13B (Lan et al., 2025)	74.8	69.8	75.1	74.3	68.0	67.1	75.7	80.1	78.0	82.7	72.9	77.1	69.5	73.5	73.8	78.6	63.3	67.6	69.5	74.1	70.7	74.8
RESAnything-12B (Wang & Zhang, 2026)	54.8	52.6	48.3	46.2	48.3	46.4	64.0	68.3	67.5	72.2	65.9	70.0	55.9	60.7	61.3	65.7	47.6	52.2	55.6	59.8	56.1	61.0
SAM3 (top-1 / union) (Carion et al., 2026)	68.1	24.0	39.9	23.1	49.2	36.4	61.0	54.8	62.5	59.1	59.8	51.9	48.5	42.9	53.8	50.0	43.6	36.5	62.5	55.5	61.7	54.4
RefAM-SD (Kukleva et al., 2026)	27.0	24.8	31.6	29.7	24.4	22.0	50.1	41.9	52.5	44.8	46.3	38.4	40.4	33.5	46.1	38.0	35.0	28.9	45.1	37.3	46.1	39.3
RefAM-Flux-12B (Kukleva et al., 2026)	27.6	25.5	32.4	30.2	24.8	22.9	52.2	42.7	55.6	46.2	49.6	40.8	43.1	34.4	48.6	38.7	37.6	30.5	48.1	39.6	48.6	41.1
LGD-7B (Li et al., 2025a)	24.0	21.9	28.7	26.8	21.4	19.2	45.4	34.5	48.8	37.8	39.3	30.8	47.2	36.7	54.6	44.4	38.1	30.1	47.3	36.6	48.1	37.7
LGD+DINO-7B (Li et al., 2025a)	28.5	26.6	33.2	31.0	25.8	23.7	49.5	42.7	54.7	48.2	41.0	36.3	49.6	47.3	58.4	50.9	38.7	34.4	50.3	45.5	51.1	47.7
RefChess (Tong et al., 2026)	29.3	27.4	33.9	32.1	26.7	24.6	55.2	48.5	57.9	52.5	50.0	43.6	48.6	41.2	54.6	48.3	39.6	32.7	50.6	42.9	51.3	44.2
Early Semantic Grounding-9B (He et al., 2026)	30.1	28.0	34.5	32.8	27.2	25.3	54.0	50.2	56.1	53.0	50.2	46.1	44.1	40.2	46.7	43.3	40.3	36.1	50.6	46.9	50.5	46.5
Early Semantic Grounding-12B (He et al., 2026)	31.5	29.3	36.2	34.3	29.0	26.9	57.6	53.9	60.3	58.1	53.8	49.6	48.5	44.4	52.4	49.3	44.4	40.1	54.5	51.2	54.3	50.9
TALENT (Jin et al., 2026)	59.1	57.0	63.7	61.8	56.6	54.4	77.8	75.9	79.4	78.3	74.8	72.8	70.1	66.9	74.9	72.3	63.4	58.8	69.7	65.9	69.2	66.8
DeRIS-B (Dai et al., 2025)	73.9	68.0	72.5	71.6	65.2	63.1	81.1	78.4	81.9	79.8	79.5	77.6	74.3	70.1	77.9	75.4	71.2	67.5	75.3	72.1	76.4	75.9
ContourVLA-3B	**83.5**	**72.3**	**80.3**	**78.1**	**74.4**	71.8	**82.4**	81.3	**83.7**	82.9	**81.1**	79.2	**77.5**	76.4	**80.7**	80.7	**74.2**	69.3	**76.6**	76.9	**76.7**	77.3

Qualitative results. Figure 4 complements the quantitative evaluation with multi-referent, part-level, and no-target examples. In the person-and-umbrella example, ContourVLA segments both requested referents; in the truck example, it delineates the specified front portion rather than extending the mask over the entire vehicle. It also correctly returns an empty prediction for the absent-target expression. Compared with the displayed baselines, these examples illustrate fewer omitted referents and less over-segmentation while preserving fine boundaries. Together, they highlight two complementary requirements of GRES: identifying the complete language-specified referent set and accurately delineating the corresponding regions.

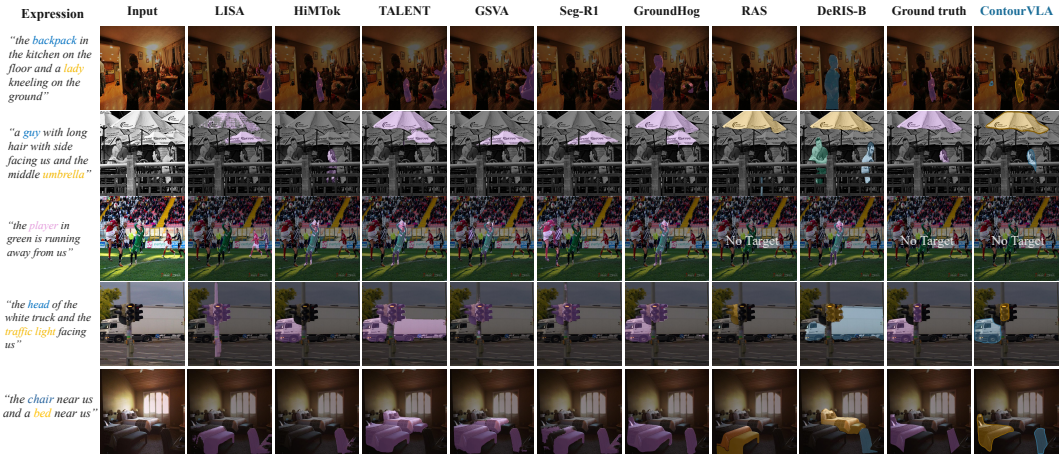

Figure 4: **Qualitative comparison on challenging GRES examples.** ContourVLA recovers multiple and part-level referents, preserves fine boundaries, and correctly rejects the no-target case.

4.3 Ablation Studies and Analysis

Closed-loop perception and EASS. Table 2 evaluates spatial sampling, semantic routing, and re-observation on gRefCOCO val. All variants are trained separately under the same SFT protocol,

with identical grounding predictions, initial contours, GAD configuration, and evolution budget. The baseline (A) uses vertex sampling with a predefined VLIM layer schedule. Adding SVS (B) or MSR (C) improves both metrics, while their combination (D) raises gIoU from 71.1 to 77.6 and cIoU from 60.8 to 67.4. Freezing the initial EASS readouts across later iterations (E), while contour states and actions continue to evolve, reduces performance to 72.0 gIoU and 61.3 cIoU, supporting the benefit of state-conditioned re-observation.

Effectiveness of DECT-GRPO. Table 3 evaluates RL post-training and transport-derived instance-level advantage modulation on gRefCOCO val. All RL variants start from the same SFT checkpoint and share training data, rollout settings, transport-based reward, and optimization budget. Shared-advantage GRPO sets $\lambda_c = 0$, while hard-assignment credit replaces soft transport only in credit computation. Shared-advantage GRPO improves over SFT, while full DECT-GRPO gains a further 3.0 gIoU and 5.1 instance F1@0.5 points, supporting instance-level modulation beyond shared rollout feedback. Its gains over hard-assignment credit (1.3 gIoU, 2.0 F1@0.5) further support soft correspondences. Removing dustbins lowers all four metrics, particularly no-target accuracy and instance F1@0.5, supporting dustbin-aware credit assignment for unmatched predictions and targets.

Table 2: Ablation of EASS and re-observation on gRefCOCO val (%). Refresh denotes EASS readout recomputation after contour updates.

Variant	SVS	MSR	Refresh	gIoU ↑	cIoU ↑
A: Baseline	×	×	✓	71.1	60.8
B: + Spatial sampling	✓	×	✓	74.2	63.5
C: + Semantic routing	×	✓	✓	74.7	63.1
D: Full EASS	✓	✓	✓	**77.6**	**67.4**
E: Frozen observations	✓	✓	×	72.0	61.3

Table 3: Ablation of DECT-GRPO on gRefCOCO val (%). N-acc is no-target accuracy; instance F1@0.5 uses one-to-one mask matching.

Variant	gIoU ↑	cIoU ↑	N-acc ↑	F1@0.5 ↑
SFT only	77.6	67.4	74.6	73.8
GRPO (shared advantage)	80.5	70.1	78.0	77.3
Hard-assignment credit	82.2	71.6	80.9	80.4
Dustbin-free credit	82.4	71.8	78.7	78.6
Full DECT-GRPO	**83.5**	**72.3**	**83.2**	**82.4**

Contour evolution and computational cost. Figure 5(a) compares Full EASS (D) and frozen observations (E) under identical grounding predictions and initial contours, reporting cumulative contour latency at action steps 2, 4, and 8. At the matched eight-step action budget, Full EASS improves Boundary IoU by 6.07 points over frozen observations, from 59.91% to 65.98%. Cumulative contour latency increases from 226.96 to 233.63 ms, corresponding to an additional 6.67 ms (2.94%). This comparison supports the benefit of state-conditioned re-observation at matched action budgets with a modest increase in contour-branch latency. Figure 5(b) visualizes the corresponding Full EASS evolution from initial contours to final masks. Detailed evaluation and timing protocols are provided in Appendix A.1.



(a) Quality and latency (b) Contour trajectories

Figure 5: **Contour evolution and latency on a subset of gRefCOCO val.** (a) Boundary IoU (lines) and cumulative contour latency at steps 2, 4, and 8 (bars) for Full EASS and frozen observations. (b) Full EASS evolution from initial contours to final masks, with consistent instance colors.

5 Conclusion

We presented ContourVLA, which recasts GRES as a closed-loop perception–action process in which editable contours guide multimodal perception and evolve through geometric actions. EASS couples contour-guided boundary sampling with multilevel semantic routing, while DECT-GRPO jointly optimizes grounding and contour actions through transport-based rewards and instance-level credit assignment. Among the compared methods, ContourVLA achieves the highest gIoU on all three gRefCOCO splits (by 2.7–8.7 points) and mIoU on all eight RES splits. Ablations further support the benefits of re-observation and instance-level credit, highlighting the value of coupling adaptive perception with explicit geometric refinement.

AI USE STATEMENT

In preparing this manuscript, we used generative AI tools to assist with translation, polish the language for clarity and readability, and identify and correct grammatical errors. All AI-assisted text was reviewed and, where necessary, revised by the authors to verify its accuracy and ensure that it preserved the intended technical meaning and remained consistent with the methods and results reported in this work. The authors take full responsibility for the final content of this paper, including all AI-assisted text.

REFERENCES

Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76:966–1005, 1988.

Shengcao Cao, Zijun Wei, Jason Kuen, Kangning Liu, Lingzhi Zhang, Jiuxiang Gu, HyunJoon Jung, Liang-Yan Gui, and Yu-Xiong Wang. Refer to any segmentation mask group with vision-language prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21853–21863, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, volume 2026, pp. 138846–138923, 2026.

Yi-Chia Chen, Wei-Hua Li, Cheng Sun, Yu-Chiang Frank Wang, and Chu-Song Chen. SAM4MLLM: Enhance multi-modal large language model for referring expression segmentation. In *European Conference on Computer Vision*, pp. 323–340, 2024.

Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C. Berg, and Alexander Kirillov. Boundary IoU: Improving object-centric image segmentation evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15329–15337, 2021.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Ming Dai, Wenxuan Cheng, Jiang-jiang Liu, Sen Yang, Wenxiao Cai, Yanpeng Sun, and Wankou Yang. DeRIS: Decoupling perception and cognition for enhanced referring image segmentation through loopback synergy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19936–19946, October 2025.

Xinheng Han, Jianfei Wang, Yu Chen, Xiang Wang, Shuai Li, Weixing Li, and Feng Pan. Credit the right box: Marginal contribution assignment for structured visual perception. *arXiv preprint arXiv:2608.01055*, 2026.

Jingxuan He, Xiyu Wang, Yunke Wang, Mengyu Zheng, and Chang Xu. Early semantic grounding in image editing models for zero-shot referring image segmentation. *arXiv preprint arXiv:2605.13122*, 2026.

Jiaqi Huang, Zunnan Xu, Jun Zhou, Ting Liu, Yicheng Xiao, Mingwen Ou, Bowen Ji, Xiu Li, and Kehong Yuan. SAM-R1: Leveraging SAM for reward feedback in multimodal segmentation via reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 138362–138383, 2025.

Shuo Jin, Siyue Yu, Bingfeng Zhang, Chao Yao, Meiqin Liu, and Jimin Xiao. TALENT: Target-aware efficient tuning for referring image segmentation. *arXiv preprint arXiv:2604.00609*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Namyup Kim, Jinsung Lee, and Suha Kwak. Improving target presence and plurality recognition for generalized referring image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 5700–5708, 2026.

Anna Kukleva, Enis Simsar, Alessio Tonioni, Muhammad Ferjad Naeem, Federico Tombari, Jan Eric Lenssen, and Bernt Schiele. RefAM: Attention magnets for zero-shot referral segmentation. *Transactions on Machine Learning Research*, 2026.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. LISA: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9579–9589, 2024.

Mengcheng Lan, Chaofeng Chen, Yue Zhou, Jiaxing Xu, Yiping Ke, Xinjiang Wang, Litong Feng, and Wei Zhang. Text4Seg: Reimagining image segmentation as text generation. In *International Conference on Learning Representations*, volume 2025, pp. 1634–1661, 2025.

Jiachen Li, Qing Xie, Renshu Gu, Jinyu Xu, Yongjian Liu, and Xiaohan Yu. LGD: Leveraging generative descriptions for zero-shot referring image segmentation. *Pattern Recognition*, 172: 112549, 2025a.

Zhang Li, Biao Yang, Qiang Liu, Shuo Zhang, Zhiyin Ma, Liang Yin, Linger Deng, Yabo Sun, Yuliang Liu, and Xiang Bai. LIRA: Inferring segmentation in large multi-modal models with local interleaved region assistance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24056–24067, 2025b.

Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23592–23601, 2023a.

Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R. Manmatha. PolyFormer: Referring image segmentation as sequential polygon generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18653–18663, 2023b.

Ye Liu, Zongyang Ma, Junfu Pu, Zhongang Qi, Yang Wu, Ying Shan, and Chang Chen. UniPixel: Unified object referring and segmentation for pixel-level visual reasoning. *Advances in Neural Information Processing Systems*, 38:126078–126108, 2026.

Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-Zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025a.

Yuqi Liu, Tianyuan Qu, Zhisheng Zhong, Bohao Peng, Shu Liu, Bei Yu, and Jiaya Jia. VisionReasoner: Unified visual perception and reasoning via reinforcement learning. *arXiv preprint arXiv:2505.12081*, 2025b.

Zhenyu Lu, Liupeng Li, Jinpeng Wang, Haoqian Kang, Yan Feng, Ke Chen, and Yaowei Wang. SegCompass: Exploring interpretable alignment with sparse autoencoders for enhanced reasoning segmentation. *arXiv preprint arXiv:2605.22658*, 2026.

Zhuoyan Luo, Yinghao Wu, Tianheng Cheng, Yong Liu, Yicheng Xiao, Hongfa Wang, Xiao-Ping Zhang, and Yujiu Yang. CoHD: A counting-aware hierarchical decoding framework for generalized referring expression segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22685–22694, 2025.

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*, pp. 11–20, 2016.

Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8530–8539, 2020.

Renjie Pi, Lewei Yao, Jiahui Gao, Jipeng Zhang, and Tong Zhang. PerceptionGPT: Effectively fusing visual perception into LLM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27114–27123, 2024.

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*, pp. 13009–13018, 2024.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded SAM: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4937–4946, 2020.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Hao Tang, Chenwei Xie, Haiyang Wang, Xiaoyi Bao, Tingyu Weng, Pandeng Li, Yun Zheng, and Liwei Wang. UFO: A unified approach to fine-grained visual perception via open-ended language interface. *Advances in Neural Information Processing Systems*, 38:83761–83791, 2026.

Shiyan Tong, Jinxia Zhang, Zhiyuan Wang, Hao Tian, Yingying Wang, Kanjian Zhang, and Haikun Wei. RefChess: Training-free contextual search for zero-shot referring image segmentation. In *Proceedings of the International Conference on Machine Learning*, 2026.

Ruiqi Wang and Hao Zhang. Resanything: Attribute prompting for arbitrary referring segmentation. *Advances in Neural Information Processing Systems*, 38:66893–66944, 2026.

Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, Jan Kautz, Lei Zhang, and Zhiding Yu. LocateAnything: Fast and high-quality vision-language grounding with parallel box decoding. In *European Conference on Computer Vision*, pp. 336–357, 2026.

Tao Wang, Changxu Cheng, Lingfeng Wang, Senda Chen, and Wuyue Zhao. HiMTok: Learning hierarchical mask tokens for image segmentation with large multimodal model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23267–23278, 2025a.

Wenxuan Wang, Tongtian Yue, Yisi Zhang, Longteng Guo, Xingjian He, Xinlong Wang, and Jing Liu. Unveiling parts beyond objects: Towards finer-granularity referring expression segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12998–13008, 2024.

Xudong Wang, Shaolun Zhang, Shufan Li, Konstantinos Kallidromitis, Kehan Li, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. SegLLM: Multi-round reasoning segmentation with large language models. In *International Conference on Learning Representations*, volume 2025, pp. 56526–56547, 2025b.

Cong Wei, Haoxian Tan, Yujie Zhong, Yujiu Yang, and Lin Ma. LaSagnA: Language-based segmentation assistant for complex queries. *arXiv preprint arXiv:2404.08506*, 2024.

Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. GSVA: Generalized segmentation via multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3858–3869, 2024.

Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip H. S. Torr. LAVT: Language-aware vision transformer for referring image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18134–18144, 2022.

Zuyao You and Zuxuan Wu. Seg-R1: Segmentation can be surprisingly simple with reinforcement learning. *arXiv preprint arXiv:2506.22624*, 2025.

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

Ruicheng Zhang, Haowei Guo, Zeyu Zhang, Puxin Yan, and Shen Zhao. GAMED-Snake: Gradient-aware adaptive momentum evolution deep snake model for multi-organ segmentation. In *2025 IEEE International Conference on Multimedia and Expo*, pp. 1–6, 2025a. doi: 10.1109/ICME59968.2025.11209898.

Ruicheng Zhang, Yu Sun, Zeyu Zhang, Jinai Li, Xiaofan Liu, Hoi Fan Au, Haowei Guo, and Puxin Yan. MARL-MambaContour: Unleashing multi-agent deep reinforcement learning for active contour optimization in medical image segmentation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, pp. 7815–7824, New York, NY, USA, 2025b. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3755147. URL https://doi.org/10.1145/3746027.3755147.

Tao Zhang, Xiangtai Li, Hao Fei, Haobo Yuan, Shengqiong Wu, Shunping Ji, Chen Change Loy, and Shuicheng Yan. OMG-LLaVA: Bridging image-level, object-level, pixel-level reasoning and understanding. In *Advances in Neural Information Processing Systems*, volume 37, pp. 71737–71767, 2024a.

Yichi Zhang, Ziqiao Ma, Xiaofeng Gao, Suhaila Shakiah, Qiaozi Gao, and Joyce Chai. GROUNDHOG: Grounding large language models to holistic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14227–14238, 2024b.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Proceedings of Robotics: Science and Systems*, 2023.

A Additional Implementation and Experimental Details

A.1 Training and Evaluation Protocol

The complete training pipeline comprises grounding pretraining followed by the two-stage contour-policy training described in the main text: supervised fine-tuning (SFT) and DECT-GRPO post-training. Across these three chronological stages, ContourVLA progressively acquires referent grounding, geometric contour control, and joint grounding-action optimization. The visual and language backbones are initialized from MoonViT and Qwen2.5-3B-Instruct, respectively.

Stage I: Comprehensive Grounding and Detection. We first train VLIM on LocateAnything-Data (Wang et al., 2026), a large-scale mixture of object detection and vision-language grounding tasks. Stage I contains 139M training instances; Figure 6 summarizes their composition. Detection provides most of the supervision, while UI, referring, OCR, layout, and pointing data contribute complementary signals for fine-grained, language-conditioned localization. This stage adapts the initialized multimodal backbone to structured referent localization. VLIM learns to produce language-conditioned grounding outputs, including bounding boxes and no-target responses, while retaining the general visual-language representations inherited from pretraining. Because LocateAnything-Data primarily provides localization annotations, Stage I trains grounding and detection rather than contour actions.

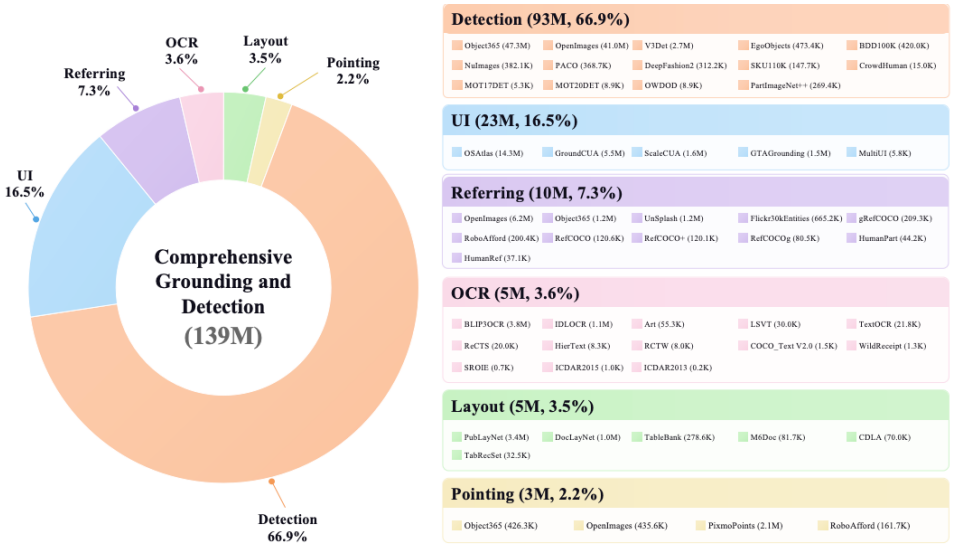

Figure 6: **Composition of the 139M LocateAnything-Data training instances used in Stage I.** The mixture contains detection (66.9%), UI (16.5%), referring (7.3%), OCR (3.6%), layout (3.5%), and pointing (2.2%) data.

Stage II: Supervised Contour Fine-Tuning. Starting from the Stage-I checkpoint, we perform SFT jointly on the training sets of gRefCOCO (Liu et al., 2023a), RefCOCO and RefCOCO+ (Yu et al., 2016), and RefCOCOg (Mao et al., 2016) to initialize the contour-based perception-action policy. Ground-truth boxes supervise VLIM grounding, and segmentation masks are converted into aligned contour targets for GAD. Perturbed copies of the ground-truth boxes initialize the contour-action branch, after which reference contour trajectories supervise the evolution from these initial states toward the target boundaries.

This stage jointly establishes referent grounding and geometric action prediction. Auxiliary geometric and routing objectives stabilize contour evolution and EASS optimization, as detailed in Appendix A.6. The resulting checkpoint initializes DECT-GRPO post-training.

Stage III: DECT-GRPO Post-Training. The final stage applies DECT-GRPO to the same four-dataset training mixture, initialized from the Stage-II checkpoint. For each image-expression pair,

Table 4: **Complete training pipeline of ContourVLA.** The four-dataset mixture comprises the training sets of gRefCOCO, RefCOCO, RefCOCO+, and RefCOCOg.

Stage	Training data	Supervision and objective
I. Grounding	LocateAnything-Data	Boxes and grounding responses; comprehensive grounding and detection pretraining.
II. Contour SFT	Four-dataset mixture	Boxes, masks, and aligned contour trajectories; joint grounding and geometric-action learning.
III. DECT-GRPO	Four-dataset mixture	Final segmentation feedback; joint optimization of the grounding and contour-action policies.

VLIM samples discrete grounding outputs that define the initial contours, and GAD then samples continuous geometric action trajectories. The final contours are rasterized into segmentation masks and evaluated by dustbin-augmented entropic transport.

The resulting transport plan provides both a transport-based rollout reward and instance-level credit maps. The reward is used to compute the shared group-relative advantage, whereas the credit maps assign instance-specific corrections to the corresponding grounding tokens and contour actions. DECT-GRPO therefore jointly optimizes the autoregressive grounding policy and the continuous contour-action policy from final segmentation feedback, without retaining the supervised contour losses used in Stage II.

Data splits and checkpoint evaluation. Stage I performs grounding and detection pretraining on the 139M-instance LocateAnything-Data corpus. To prevent benchmark information leakage, entries originating from gRefCOCO, RefCOCO, RefCOCO+, or RefCOCOg are restricted to their official training splits; no validation or test annotations or expressions from these benchmarks are used during training. For Stages II and III, we likewise pool only the designated training splits of the four benchmarks, retaining the same data mixture for SFT and DECT-GRPO. After Stage III, a single checkpoint is evaluated on all validation and test splits without further adaptation to individual benchmarks.

Mask-level metrics. Each image-expression pair is treated as one evaluation sample. Predicted instance masks are merged by union, and the target mask is the union of all instances referred to by the expression. For RES, mIoU averages foreground IoU over the evaluated samples. For GRES, gIoU extends this average to no-target samples (Liu et al., 2023a): an empty prediction receives a score of one when the target is empty, whereas a nonempty prediction receives zero. An empty prediction for a nonempty target also receives zero. The cIoU metric divides the summed intersection area by the summed union area over the evaluated split. Correct empty predictions contribute zero to both sums, while foreground predictions on no-target samples contribute to the union.

No-target accuracy. N-acc is the percentage of ground-truth no-target samples for which the predicted mask contains no foreground pixels. Its denominator includes only no-target samples, rather than the full evaluation set.

Instance-level evaluation. Instance F1@0.5 is computed from individual instance masks before union. For each expression, we construct a one-to-one matching that maximizes the number of prediction-target pairs with mask IoU of at least 0.5. Matched pairs are true positives (TP); unmatched predictions and targets are false positives (FP) and false negatives (FN), respectively. Counts are accumulated over all evaluated expressions, and

$$\mathrm{F1@0.5} = \frac{2\sum \mathrm{TP}}{2\sum \mathrm{TP} + \sum \mathrm{FP} + \sum \mathrm{FN}}. \tag{16}$$

For no-target expressions, predicted instances contribute false positives, while a correct empty prediction contributes no counts. For expressions with targets but no predictions, all targets contribute false negatives. The final score is reported as a percentage.

Boundary evaluation. For Figure 5, Boundary IoU (bIoU) (Cheng et al., 2021) is evaluated at the original image resolution on the same fixed subset of gRefCOCO validation expressions containing targets. At each action step, the predicted contours are rasterized and merged by union; the ground-truth instance masks are merged in the same way. Boundary regions are extracted by subtracting an eroded mask from the original mask, using a 3×3 kernel with

$$r = \max\Big(1, \operatorname{round}\Big(0.02\sqrt{H^2 + W^2}\Big)\Big) \tag{17}$$

erosion iterations, where H and W are the original image dimensions. Zero padding accounts for boundaries touching the image border. Boundary IoU is the intersection-over-union of the predicted and ground-truth boundary regions, averaged over expressions and reported as a percentage. Samples with no predicted mask receive zero and remain in the average at every step. This evaluation operates on expression-level union masks and does not require instance matching.

Latency protocol. For Figure 5(a), Full EASS and frozen observations use identical grounding predictions and initial contours, with two evolution iterations and four action steps per iteration. Cumulative contour latency is measured at action steps 2, 4, and 8 on the shared subset with predicted boxes. The step-2 measurement includes the cost of predicting the complete first four-step action chunk and executing its first two residual steps. Timing includes contour-branch preparation, EASS readouts, GAD execution, and contour updates, while excluding image encoding, grounding generation, and mask rasterization. Intermediate contours are recorded from each model's complete inference run.

A.2 Additional Compared Methods

Table 1 includes several compared methods that are not described individually in the main text. We summarize them below by their primary modeling interface. Backbone-specific variants of the same method share the corresponding formulation and are reported separately in Table 1 according to their model configurations.

Modular and task-specific segmentation. Grounded-SAM (Ren et al., 2024) assembles open-vocabulary grounding and promptable segmentation models for diverse visual tasks. UniRES (Wang et al., 2024) extends referring segmentation toward fine-grained object parts. DeRIS (Dai et al., 2025) decouples perception and cognition and reconnects them through a loopback-synergy mechanism; Table 1 reports its DeRIS-B variant. TALENT (Jin et al., 2026) adopts target-aware efficient tuning for referring image segmentation.

Pixel-grounded multimodal models. PerceptionGPT (Pi et al., 2024) connects large multimodal reasoning with fine-grained visual perception and pixel prediction. LaSagnA (Wei et al., 2024) addresses language-driven segmentation under complex queries, whereas OMG-LLaVA (Zhang et al., 2024a), GroundHog (Zhang et al., 2024b), UniPixel (Liu et al., 2026), and UFO (Tang et al., 2026) unify image-, object-, region-, or pixel-level understanding through multimodal language interfaces. SAM4MLLM (Chen et al., 2024) couples an MLLM with SAM-based segmentation, SegLLM (Wang et al., 2025b) performs multi-round reasoning for segmentation, and LIRA (Li et al., 2025b) introduces local interleaved region assistance. Text4Seg (Lan et al., 2025) formulates segmentation as text generation, while RESAnything (Wang & Zhang, 2026) uses attribute prompting for arbitrary referring segmentation.

Reasoning and zero-shot transfer. VisionReasoner (Liu et al., 2025b) jointly develops visual perception and reasoning through reinforcement learning. RefAM (Kukleva et al., 2026) uses attention-based localization for zero-shot referral segmentation, LGD (Li et al., 2025a) leverages generated descriptions, RefChess (Tong et al., 2026) performs training-free contextual search, and Early Semantic Grounding (He et al., 2026) extracts grounding signals from image-editing models.

A.3 Additional Qualitative Comparisons

Figures 7 and 8 provide additional side-by-side comparisons on challenging referring expressions. The examples cover plural and multi-instance descriptions, numerical and spatial constraints, het-

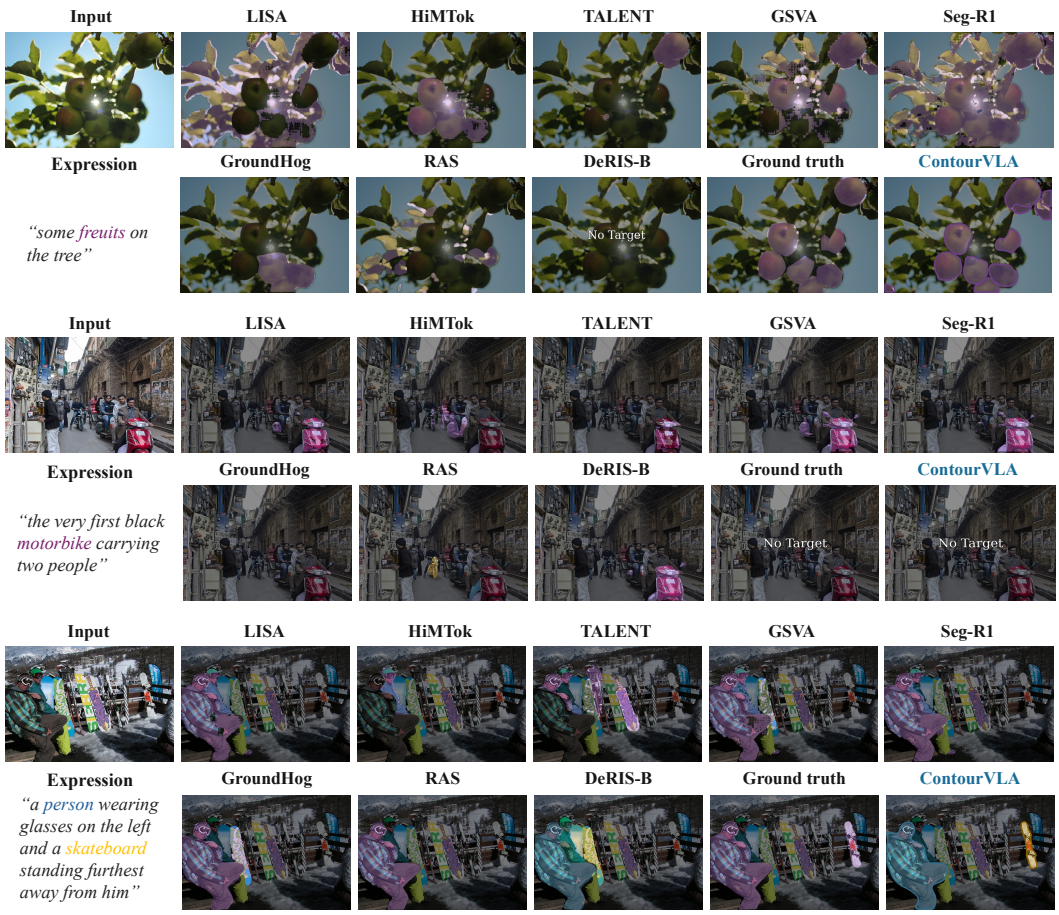

Figure 7: **Additional qualitative comparisons on challenging GRES expressions (I).** The examples include plural fruit referents, an absent target, and a multi-referent expression combining visual attributes and spatial relations. Different mask colors distinguish referred instances; "No Target" denotes an empty prediction.

erogeneous referents, and a no-target case, complementing the quantitative evaluation in the main text.

A.3.1 Out-of-Distribution Video Segmentation

To examine qualitative robustness beyond the image benchmarks, we apply the trained ContourVLA checkpoint to sampled frames from four out-of-distribution video examples. Figure 9 shows predictions in temporal order for the query displayed at the left of each panel. Across object manipulation, hand–object interaction, and human motion, the predicted masks generally remain aligned with the referred regions despite changes in pose, position, appearance, and partial occlusion. This experiment provides a qualitative OOD assessment and does not report video-level temporal-consistency metrics.

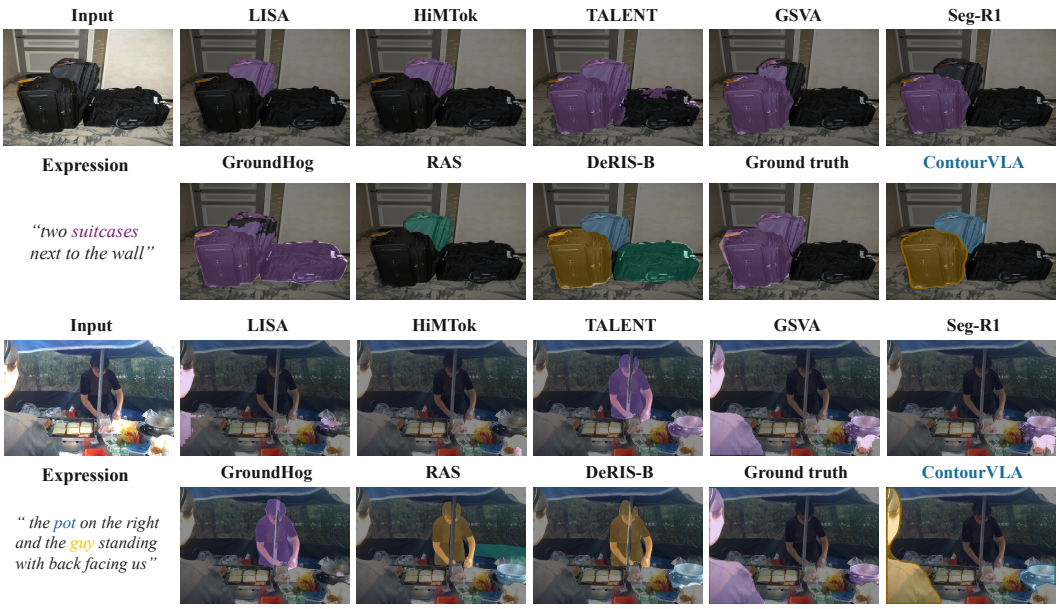

Figure 8: **Additional qualitative comparisons on challenging GRES expressions (II).** The examples evaluate numerical and spatial constraints for multiple suitcases and a heterogeneous object-person query involving a pot and a person. Different mask colors distinguish referred instances.

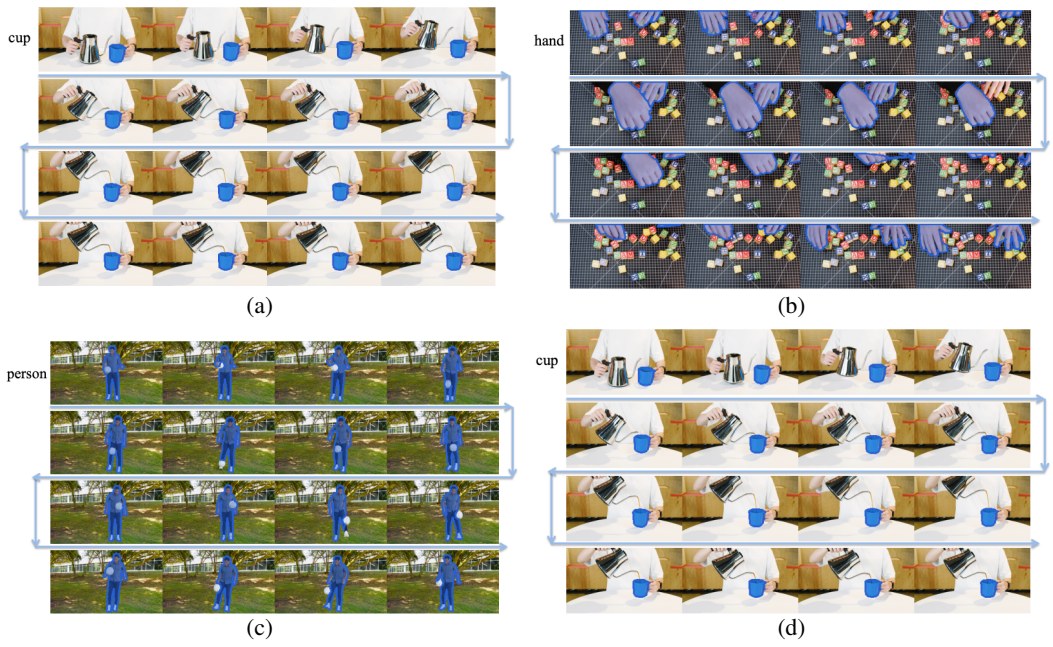

Figure 9: **Qualitative segmentation on out-of-distribution videos.** Panels (a)–(d) show sampled frames in temporal order, indicated by the arrows. The referring query is displayed at the left of each panel.

A.4 State Descriptors and Decoder Initialization

Seven-dimensional routing state. The implementation instantiates the factors (g_i, c_i^k, r^k) in equation 5 with a seven-dimensional state descriptor: log box aspect ratio, log box area, normalized point

allocation, contour irregularity, normalized evolution index, and the mean and maximum displacement from the preceding evolution iteration. Contour irregularity is the coefficient of variation of the vertex radii about the box center in box-relative coordinates. Displacements are also normalized by box width and height. All statistics exclude padded vertices, and the two displacement entries are set to zero at the initial iteration. For fixed boxes and point allocations, the first three descriptors remain constant across iterations and encode instance characteristics rather than evolution progress.

Initialization and routing regularization. Within f_q, the final projection of the state-dependent component is zero-initialized. Consequently, the initial query is determined by the evolution-depth slot embedding, while the state-dependent offset is introduced gradually during training. The output projection of the residual semantic-attention branch is also zero-initialized, so the initial GAD block preserves its spatial and instance-context pathway. During SFT, aggregate-usage, maximum-probability, and expected-depth regularizers constrain the router as specified in equation 35. During DECT-GRPO, EASS is optimized through the action likelihood and the straight-through routing path, without an additional router loss.

Routing-score scale. In equation 6, normalized queries and keys yield dot products in $[-1, 1]$. When the prior weight is zero, the largest soft routing probability is bounded by

$$\max_{\ell} p_{i,\ell}^{k,d} \le \frac{\exp(2)}{\exp(2) + |\mathcal{L}| - 1}. \tag{18}$$

The forward argmax still selects a single layer, but the bound limits the attainable value of a confidence objective defined on the soft probabilities. Equation 6 uses neither a routing temperature nor an additional logit scale.

A.5 Coordinate Conventions and Action-Chunk Execution

Box-normalized normal-direction sampling. All image coordinates are normalized to $[0, 1]^2$. For box $b_i = (x_i^1, y_i^1, x_i^2, y_i^2)$, define its origin $q_i = (x_i^1, y_i^1)$ and anisotropic scale matrix

$$D_i = \mathrm{diag}(w_i, h_i), \qquad \bar{c}_{i,p}^k = D_i^{-1}(c_{i,p}^k - q_i). \tag{19}$$

The cyclic central-difference tangent and outward unit normal are computed in this box-normalized coordinate system:

$$\bar{\tau}_{i,p}^k = \bar{c}_{i,p+1}^k - \bar{c}_{i,p-1}^k, \qquad n_{i,p}^k = \frac{R_{\mathrm{out}} \bar{\tau}_{i,p}^k}{\max(\|\bar{\tau}_{i,p}^k\|_2, 10^{-6})}, \tag{20}$$

where R_{out} rotates the tangent by $-\pi/2$ for a counterclockwise contour and by $+\pi/2$ for a clockwise contour. Indices are cyclic, and the contour orientation is determined from its signed area. The sampling coordinates used in equation 3 are therefore

$$s_{i,p,j}^{k,\pm} = c_{i,p}^k \pm o_j D_i n_{i,p}^k = c_{i,p}^k \pm o_j \big(n_{i,p}^k \odot (w_i, h_i)\big). \tag{21}$$

This is strict normal sampling in the normalized-box domain and box-scale-normalized directional sampling in the image domain. Under anisotropic scaling, $D_i n_{i,p}^k$ need not be parallel to the Euclidean image-space normal; the latter would instead be proportional to $D_i^{-\top} n_{i,p}^k$ and is not used here. Before bilinear interpolation, $s = (s_x, s_y)$ is mapped to feature-grid coordinates $((W_F - 1)s_x, (H_F - 1)s_y)$ and clipped to the valid grid boundary.

Chunk prediction and state transition. At closed-loop iteration k, EASS is evaluated once on $C_i^{k,0} = C_i^k$. Conditioned on this observation, GAD predicts the entire chunk $A_i^k = \{a_{i,p}^{k,t}\}_{p=1,t=1}^{P_i,T}$ in parallel. Each bounded action is a residual in box-normalized coordinates:

$$a_{i,p}^{k,t} = \delta_{\max} \tanh(u_{i,p}^{k,t}), \qquad \bar{c}_{i,p}^{k,t} = \bar{c}_{i,p}^{k,t-1} + a_{i,p}^{k,t}, \tag{22}$$

or, equivalently,

$$c_{i,p}^{k,t} = c_{i,p}^{k,t-1} + D_i a_{i,p}^{k,t}. \tag{23}$$

The T residual fields are executed sequentially without refreshing visual or semantic observations inside the chunk. Coordinates are clipped to the image domain, while vertex ordering and valid-point

Table 5: Default spatial sampling, action execution, and optimization configuration. Stage-II and Stage-III batch sizes are global.

Component	Setting	Interpretation
Input / feature grid	896×896 / 56×56	Fixed-resolution input and stride-16 VLIM grid
Contour vertices	$P_i \in [64, 128]$	Perimeter-adaptive valid-point count
Spatial offsets	$J = 4$, $\{o_j\} = \{1/64, 1/32, 1/16, 1/8\}$	Fractions of box-normalized distance on each side
GAD depth	$D = 6$ blocks	One routed semantic feature per block
Action horizon	$K = 2$ chunks, $T = 4$ steps/chunk	Eight action steps, with re-observation between the two chunks.
Action bound	$\delta_{\max} = 0.05$	Per-axis bound in normalized-box coordinates
VLIM adaptation	LoRA rank 64, $\alpha = 128$, dropout 0.05	Applied to attention Q/K/V/O and MLP gate/up/down projections
Stage-II SFT	30k steps; batch 128; LR 2×10^{-5} (LoRA) and 1×10^{-4} (EASS/GAD)	Four-dataset joint supervised training
Stage-III DECT-GRPO	8k updates; 32 input groups/update; $G = 8$; LR 5×10^{-6} (LoRA) and 2×10^{-5} (EASS/GAD)	One update per sampled rollout group
Grounding sampling	temperature $[0.65, 0.75]$, top-$p = 0.95$	Grouped rollout diversity with fixed nucleus threshold
Optimizer	AdamW, $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.01	Cosine decay, 3% warmup, gradient clipping at 1.0
Numerics	bfloat16 with float32 rewards and transport	Mixed-precision policy optimization

counts remain fixed. After the final within-chunk step, $C_i^{k+1} = C_i^{k,T}$; normals, spatial samples, routing queries, and GAD observations are then recomputed before predicting the next chunk. Thus, T controls the open-loop action horizon, whereas K controls the number of perception-action cycles. Deterministic SFT and inference use $u = \mu$; DECT-GRPO samples u according to equation 39.

Centralized implementation configuration. Table 5 reports the default geometry and optimization settings. With $K = 2$ and $T = 4$, the policy executes eight action steps and re-observes the contour after every four steps. Since $\delta_{\max} = 0.05$, one step can move a vertex by at most 5% of the box width and height along the two normalized coordinate axes.

A.6 Supervised Targets and Loss Details

Grounding targets and initial contours. SFT uses teacher-forced grounding answers constructed from quantized and subsequently dequantized ground-truth boxes. A separate copy of each box is perturbed to initialize the contour-action branch; the grounding targets themselves are not jittered. Each perturbed box initializes an elliptical contour with 64-128 valid vertices, and the corresponding ground-truth boundary is resampled by arc length to the same number of vertices. The perturbation magnitude increases linearly from 0.05 to 0.10 during the first 3,000 optimization steps.

Reference contour trajectories. Let Γ_i be the resampled ground-truth contour. We align it to C_i^0 by minimizing squared vertex distances over cyclic shifts and both traversal directions, denoted by $\mathcal{P}_{\text{cyc}}$:

$$\pi_i^\star = \arg\min_{\pi\in\mathcal{P}_{\text{cyc}}} \sum_{p=1}^{P_i} \|\Gamma_{i,\pi(p)} - c_{i,p}^0\|_2^2, \qquad c_{i,p}^\star = \Gamma_{i,\pi_i^\star(p)}. \tag{24}$$

This operation fixes the within-instance vertex correspondence and is distinct from the inter-instance transport used by DECT-GRPO. At evolution iteration k, the reference position for action step t interpolates from the iteration's actual starting contour to the aligned target:

$$\widetilde{C}_i^{k,t} = \text{sg}(C_i^k) + \frac{t}{T}\big(C_i^\star - \text{sg}(C_i^k)\big). \tag{25}$$

The references in equation 25 provide soft positional supervision through equation 29, rather than executable expert-action trajectories. The predicted contour remains subject to the bounded transitions in equation 22-equation 23, so these references are not guaranteed to be reachable within every chunk. Reaching the aligned target at each chunk endpoint is therefore a supervision objective, not a hard constraint. The stop-gradient operator is applied only when constructing the reference. The predicted contour trajectory itself remains connected across iterations, so later trajectory losses can backpropagate through earlier contour updates.

Smooth-ℓ_1 and trajectory weights. For a scalar residual, the smooth-ℓ_1 penalty is

$$h_\beta(r) = \begin{cases} r^2/(2\beta), & |r| < \beta, \\ |r| - \beta/2, & \text{otherwise}, \end{cases} \qquad \beta = 0.05. \tag{26}$$

With $k = 0, \dots, K-1$ and $t = 1, \dots, T$, the time weights in equation 29 are

$$\omega_{k,t} = \frac{((kT+t)/(KT))^\gamma}{\sum_{k'=0}^{K-1}\sum_{t'=1}^{T}((k'T+t')/(KT))^\gamma}, \qquad \gamma = 1. \tag{27}$$

These weights sum to one over all evolution iterations and action steps. To emphasize vertices that must travel farther, we define box-normalized distances from the initial contour:

$$\begin{aligned} d_{i,p} &= \|(c^\star_{i,p} - c^0_{i,p}) \oslash b^{\text{size}}_i\|_2, \qquad \bar{d}_i = P_i^{-1}\sum_p d_{i,p}, \\ v'_{i,p} &= 1 + \lambda_{\text{travel}}\frac{d_{i,p}}{\max(\bar{d}_i, \epsilon_{\text{num}})}, \qquad v_{i,p} = \frac{v'_{i,p}}{P_i^{-1}\sum_{q=1}^{P_i} v'_{i,q}}, \end{aligned} \tag{28}$$

where $\lambda_{\text{travel}} = 1$ and $\epsilon_{\text{num}} > 0$ prevents division by zero. All averages exclude padded vertices. Normalizing $v_{i,p}$ to unit mean preserves the average weight of each instance, and equation 29 weights instances equally rather than in proportion to their vertex counts.

Trajectory supervision. Let $\mathcal{I}$ contain instances with valid target contours. Their box-normalized residual and trajectory loss are

$$\begin{aligned} r_{i,k,t,p} &= (c^{k,t}_{i,p} - \tilde{c}^{k,t}_{i,p}) \oslash b^{\text{size}}_i, \\ \mathcal{L}_{\text{traj}} &= \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\frac{1}{P_i}\sum_{p=1}^{P_i} v_{i,p}\sum_{k,t}\omega_{k,t}\sum_{d\in\{x,y\}} h_{0.05}(r_{i,k,t,p,d}). \end{aligned} \tag{29}$$

Box dimensions are lower-bounded before normalization. The temporal weights emphasize later actions, while the unit-mean travel weights emphasize vertices farther from their aligned targets.

Region supervision. At the end of each evolution iteration, we differentiably rasterize the predicted contour within an expanded region of interest (ROI) and compare it with the ground-truth mask rasterized in the same ROI:

$$\mathcal{L}_{\text{region}} = \frac{\sum_{k=0}^{K-1}\eta_k \,\text{Mean}_{i\in\mathcal{I}}\left[1 - \text{SoftIoU}\Big(\mathcal{R}_{\text{soft}}(C_i^{k,T}), Y_i^{\text{ROI}}\Big)\right]}{\sum_{k=0}^{K-1}\eta_k}, \qquad \eta_k = \frac{k+1}{K}. \tag{30}$$

The soft-rasterization temperature is 0.0025. Unlike the mask evaluation used to construct the RL reward, this operation is differentiable with respect to the predicted contour.

Final-contour regularization. Let $X_{i,p} = c^K_{i,p} \oslash b^{\text{size}}_i$ and use cyclic adjacency within each instance's valid point count. Smoothness penalizes deviation from the midpoint of neighboring vertices:

$$\mathcal{L}_{\text{smooth}} = \text{Mean}_{i,p}\left\| X_{i,p} - \frac{X_{i,p-1} + X_{i,p+1}}{2}\right\|_1. \tag{31}$$

With $l_{i,p} = \|X_{i,p+1} - X_{i,p}\|_2$ and $\bar{l}_i = P_i^{-1}\sum_p l_{i,p}$, the spacing penalty is

$$\mathcal{L}_{\text{spacing}} = \text{Mean}_i \frac{1}{P_i}\sum_{p=1}^{P_i}(l_{i,p} - \bar{l}_i)^2. \tag{32}$$

This term promotes uniform edge lengths within each contour without imposing a common absolute spacing across objects. Both regularizers are applied only to the final contour.

Auxiliary signed-distance supervision. For locations q on a 16×16 grid in each object ROI, let $d_i^\star(q) = \mathrm{SDF}(q, C_i^\star)$ denote the target signed distance. We normalize it by $r_i = \max((w_i + h_i)/2, 10^{-3})$ and truncate only the target:

$$\begin{aligned} \widetilde{d}_i(q) &= \mathrm{clip}(d_i^\star(q)/r_i, -\tau_{\mathrm{sdf}}, \tau_{\mathrm{sdf}}), \qquad \tau_{\mathrm{sdf}} = 0.25, \\ \mathcal{L}_{\mathrm{SDF}} &= \mathrm{Mean}_{i,q}\, h_\beta\Big(\widehat{d}_i(q)/r_i - \widetilde{d}_i(q)\Big). \end{aligned} \tag{33}$$

The predicted distance is not clipped, which preserves gradients outside the target truncation interval.

Grounding supervision. Let $\mathcal{S}$ collect supervised answer-token positions. The teacher-forced grounding objective is

$$\mathcal{L}_{\mathrm{ground}} = -\frac{1}{|\mathcal{S}|} \sum_{n \in \mathcal{S}} \log \pi_\theta^y(y_n^\star \mid I, x, y_{<n}^\star). \tag{34}$$

Supervised positions include coordinate tokens, structural tags, no-target responses, and the end-of-sequence token. Prompt positions are excluded, and copied expression tokens inside `<ref>` tags are masked by default. The target coordinates are obtained from the annotated boxes before geometric perturbation. If an example contains no valid target contour, the geometric losses provide no contour-learning signal, whereas grounding supervision, including no-target responses, remains active.

Routing regularization during SFT. Index decoder slots first by evolution iteration $k = 0, \ldots, K-1$ and then by block depth $d = 1, \ldots, D$, with $s = kD + d \in \{1, \ldots, KD\}$, and write $p_{i,s,\ell}$ for their layer probabilities. Define $\overline{p}_\ell = \mathrm{Mean}_{i,s}\, p_{i,s,\ell}$ and expected layer depth $m_{i,s} = \sum_\ell \ell p_{i,s,\ell}$. The regularizer is

$$\begin{aligned} \mathcal{L}_{\mathrm{router}} = {} & -\lambda_{\mathrm{div}} H(\overline{p}) - \lambda_{\mathrm{conf}}\, \mathrm{Mean}_{i,s} \max_\ell p_{i,s,\ell} \\ & + \lambda_{\mathrm{mono}}\, \mathrm{Mean}_{i,\, 1 \le s < KD}\, \mathrm{ReLU}(m_{i,s} - m_{i,s+1}). \end{aligned} \tag{35}$$

The first term encourages diverse aggregate layer usage; the second rewards confident selection through the maximum probability rather than by minimizing per-slot entropy. The final term penalizes decreases in expected layer depth between adjacent evolution–depth slots. The confidence objective respects the finite probability range in equation 18 and therefore does not require probabilities to approach one. These auxiliary terms are used only during SFT and are not added separately during DECT-GRPO.

Effective SFT coefficients. The SFT objective used in our experiments is

$$\begin{aligned} \mathcal{L}_{\mathrm{SFT}} = {} & \mathcal{L}_{\mathrm{ground}} + 2\mathcal{L}_{\mathrm{traj}} + 0.04\mathcal{L}_{\mathrm{smooth}} + 0.10\mathcal{L}_{\mathrm{spacing}} \\ & + 0.70\mathcal{L}_{\mathrm{region}} + 0.178\mathcal{L}_{\mathrm{SDF}} + \mathcal{L}_{\mathrm{router}}. \end{aligned} \tag{36}$$

The coefficients internal to $\mathcal{L}_{\mathrm{router}}$ are specified separately. These supervised objectives initialize the policy and are not retained during DECT-GRPO post-training.

A.7 DECT-GRPO Replay and Training Protocol

Configuration and group-relative advantages. The DECT-GRPO configuration uses $G = 8$, $\tau = 0.001$, $\sigma_a = 0.05$, $\lambda_c = 0.5$, $\lambda_a = 1.0$, clipping range $c = 0.2$, $\beta_{\mathrm{KL}} = 0.02$, transport entropy $\varepsilon = 0.05$, and null score $b_\varnothing = 0.25$. For the G rewards associated with one image–expression pair, define

$$\overline{R} = \frac{1}{G}\sum_{g=1}^{G} R_g, \qquad \sigma_R = \sqrt{\frac{1}{G}\sum_{g=1}^{G}(R_g - \overline{R})^2}, \qquad A_g = \mathbf{1}\{\sigma_R \ge \tau\}\frac{R_g - \overline{R}}{\max(\sigma_R, \tau)}. \tag{37}$$

The threshold τ suppresses numerically unstable updates when the rewards within a group are effectively identical.

Rollout likelihoods and replay. For a sampled grounding sequence, the grounding branch follows an autoregressive categorical policy with token-level log-probability

$$\ell^y_{g,n} = \log \pi^y_\theta(y_{g,n} \mid I, x, y_{g,<n}). \tag{38}$$

For each valid action index $\xi = (i, k, t, p)$, GAD predicts a two-dimensional mean $\mu_{g,\xi}$ and samples $u_{g,\xi} \sim \mathcal{N}(\mu_{g,\xi}, \sigma_a^2 I_2)$. The bounded contour action is $a_{g,\xi} = \delta_{\max} \tanh(u_{g,\xi})$, with log-probability

$$\ell^a_{g,\xi} = \log \mathcal{N}(u_{g,\xi}; \mu_{g,\xi}, \sigma_a^2 I_2) - \sum_{h=1}^{2} \log\left[\delta_{\max}\left(1 - \tanh^2 u_{g,\xi,h}\right)\right]. \tag{39}$$

During rollout collection, we store the sampled grounding outputs, pre-$\tanh$ action latents, behavior-policy log-probabilities, valid-point masks, grounding boxes, and historical contour states. During replay, the sampled quantities remain fixed, while current grounding log-probabilities, trainable observations, and action means are recomputed with gradients enabled. Dropout is disabled during both collection and replay.

Direct geometric policy gradients. For a fixed latent u and fixed variance, the transformation Jacobian is constant with respect to the current mean. Thus

$$\frac{\partial \ell^a}{\partial \mu} = \frac{u - \mu}{\sigma_a^2}, \qquad \nabla_\phi \mathcal{L}^a_g \approx -\frac{1}{|\mathcal{D}_g|} \sum_{\xi \in \mathcal{D}_g} A^a_{g,\xi} \nabla_\phi \ell^a_{g,\xi}. \tag{40}$$

This approximation holds when the likelihood ratios are close to one and clipping is inactive. The sampled continuous actions and their historical states are detached, while the current mean network remains differentiable. The resulting score-function gradient trains GAD and its observation pathways without differentiating through the mask reward. DECT-GRPO retains none of the SFT trajectory, region, smoothness, spacing, SDF, or router losses, and it introduces no action-reference KL or action-entropy term. Inference and SFT use deterministic actions $\delta_{\max} \tanh(\mu)$.

Output-count weighting. Grounding-token counts and valid-action counts can induce different centered corrections for the same instance. With correctly aligned, disjoint grounding spans and valid-point masks, each branch preserves its rollout-level advantage sum under instance-level modulation:

$$\sum_{n=1}^{L_g} A^y_{g,n} = L_g A_g, \qquad \sum_{\xi \in \mathcal{D}_g} A^a_{g,\xi} = |\mathcal{D}_g| A_g. \tag{41}$$

This conservation property does not make the branch gradients equal, because their likelihood models and normalizations differ. Within an instance, all valid action steps and vertices receive the same instance-corrected advantage; DECT-GRPO does not estimate an additional temporal credit signal. For a single predicted instance, branch-wise centering yields zero instance correction, but the rollout still receives the shared advantage A_g.

Grounding sampling and reference convention. Grounding answers use temperature and nucleus sampling. For each rollout, the temperature is sampled from $[0.65, 0.75]$, while the nucleus threshold remains fixed. Replay scores the re-tokenized answer under the model's untruncated softmax. Because collection and replay therefore use different token distributions, the resulting objective should be interpreted as the implemented branch-wise policy-gradient surrogate rather than as an exactly unbiased on-policy estimator of the transformed grounding sampler. The grounding reference branch is formed by disabling the LoRA adapters while sharing the current visual-to-language projector. Reference log-probabilities are detached, but the reference is not a separately frozen copy of the complete SFT model. The exponential reference penalty is applied only to grounding and is not treated as an exact KL estimate under the transformed sampler. Behavior-policy log-probabilities are retained for both branches, and each rollout group is used for a single update rather than for multiple PPO epochs.

A.8 Derivation of Dustbin-Augmented Entropic Transport

For rollout g, DECT solves the entropy-regularized transport problem

$$P_g^\star = \arg\max_{P \in \mathcal{U}_g} \left[\langle P, \overline{Q}_g \rangle - \varepsilon \sum_{i,j} P_{ij} \log P_{ij} \right], \tag{42}$$

where $\mathcal{U}_g = \{P \geq 0 \mid P\mathbf{1} = \mathbf{a}_g,\ P^\top \mathbf{1} = \mathbf{b}_g\}$ specifies the prediction-side and target-side marginals, with $\mathbf{1}^\top \mathbf{a}_g = \mathbf{1}^\top \mathbf{b}_g$. The augmented score matrix $\overline{Q}_g$ contains both real prediction–target IoUs and dustbin scores, so matched and unmatched instances are handled within a single transport problem.

Operational marginals, dustbin capacity, and solver settings. For $M_g > 0$ and $N > 0$, let $T_g = M_g + N$. We use the normalized SuperGlue-style marginals

$$\mathbf{a}_g = \frac{1}{T_g} \begin{bmatrix} \mathbf{1}_{M_g} \\ N \end{bmatrix}, \qquad \mathbf{b}_g = \frac{1}{T_g} \begin{bmatrix} \mathbf{1}_N \\ M_g \end{bmatrix}. \tag{43}$$

Thus, every real prediction supplies $1/T_g$ mass and every real target demands $1/T_g$ mass. The null-prediction row has capacity N/T_g, enough to explain all targets as missed, while the null-target column has capacity M_g/T_g, enough to absorb all predictions as unmatched. The null–null entry receives whichever part of these capacities remains after real matches are formed. Consequently, dustbin capacity is determined by equation 43, rather than merely by appending a row and column.

Mask IoUs are kept on their native $[0, 1]$ scale. In all experiments we fix $\varepsilon = 0.05$ and $b_\varnothing = 0.25$; hence a weak real match competes directly with a null match of score 0.25, while entropy smooths competition within a scale of 0.05. We perform Sinkhorn scaling in the log domain, initialized with zero log-scalings, until the maximum row/column marginal residual is below 10^{-6} or 1000 iterations are reached. Cases with $M_g = 0$ or $N = 0$ bypass transport and use the explicit set rewards in Section A.9.

Dustbin-free credit ablation. For the variant in Table 3, the rollout reward remains unchanged. For $M_g > 0$ and $N > 0$, we compute a separate entropic transport plan $\Pi_g^\star$ on the real prediction–target IoU matrix Q_g, with uniform row and column marginals $1/M_g$ and $1/N$, respectively. Instance credits are computed as $e_{g,i} = \sum_{j=1}^{N} \Pi_{g,ij}^\star Q_{g,ij} / \sum_{j=1}^{N} \Pi_{g,ij}^\star$. Entropy regularization, solver settings, branch-wise credit normalization, advantage modulation, and empty-set handling remain unchanged.

Equivalent reward identity. For $M_g, N > 0$, define the total real-match overlap mass as

$$S_g = \sum_{i=1}^{M_g} \sum_{j=1}^{N} P_{g,ij}^\star Q_{g,ij}. \tag{44}$$

When the marginal constraints in equation 43 are satisfied exactly, every real prediction row and real target column has total mass $1/(M_g + N)$. Substitution into equation 12 gives

$$\mathrm{Prec}_g = \frac{M_g + N}{M_g} S_g, \qquad \mathrm{Rec}_g = \frac{M_g + N}{N} S_g, \qquad \boxed{R_g = 2S_g}. \tag{45}$$

The zero-reward convention following equation 12 extends this identity to $S_g = 0$, including the case in which all real prediction–target IoUs vanish. This equality provides an implementation check rather than a new reward definition. Because finite Sinkhorn scaling satisfies the marginals only up to the stopping tolerance, numerical checks of the identity allow the corresponding residual error.

Introducing Lagrange multipliers $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for the row and column constraints gives

$$\mathcal{L}(P, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i,j} P_{ij} \overline{Q}_{g,ij} - \varepsilon \sum_{i,j} P_{ij} \log P_{ij} + \boldsymbol{\alpha}^\top (\mathbf{a}_g - P\mathbf{1}) + \boldsymbol{\beta}^\top (\mathbf{b}_g - P^\top \mathbf{1}). \tag{46}$$

Stationarity with respect to P_{ij} yields

$$\overline{Q}_{g,ij} - \varepsilon(\log P_{ij} + 1) - \alpha_i - \beta_j = 0, \tag{47}$$

and therefore

$$P_{ij} = u_i \exp\left(\frac{\overline{Q}_{g,ij}}{\varepsilon}\right) v_j, \tag{48}$$

where the constant term is absorbed into the positive scaling factors u_i and v_j. Defining the elementwise exponential kernel

$$\mathbf{K}_g = \exp\left(\frac{\overline{Q}_g}{\varepsilon}\right), \tag{49}$$

the optimal plan takes the matrix form

$$P_g^\star = \mathrm{diag}(\mathbf{u})\, \mathbf{K}_g\, \mathrm{diag}(\mathbf{v}). \tag{50}$$

The marginal constraints determine the scaling vectors:

$$\mathbf{u} \odot (\mathbf{K}_g \mathbf{v}) = \mathbf{a}_g, \qquad \mathbf{v} \odot (\mathbf{K}_g^\top \mathbf{u}) = \mathbf{b}_g, \tag{51}$$

which leads to the alternating Sinkhorn updates

$$\mathbf{u} \leftarrow \mathbf{a}_g \oslash (\mathbf{K}_g \mathbf{v}), \qquad \mathbf{v} \leftarrow \mathbf{b}_g \oslash (\mathbf{K}_g^\top \mathbf{u}), \tag{52}$$

where $\odot$ and $\oslash$ denote elementwise multiplication and division. For numerical stability, the implementation maintains dual log-scalings $\mathbf{f} = \varepsilon \log \mathbf{u}$ and $\mathbf{h} = \varepsilon \log \mathbf{v}$ and alternates

$$\begin{aligned} f_i &\leftarrow \varepsilon \left[\log a_{g,i} - \mathrm{LSE}_j\left(\frac{\overline{Q}_{g,ij} + h_j}{\varepsilon}\right)\right], \\ h_j &\leftarrow \varepsilon \left[\log b_{g,j} - \mathrm{LSE}_i\left(\frac{\overline{Q}_{g,ij} + f_i}{\varepsilon}\right)\right], \\ P_{g,ij} &= \exp\left(\frac{\overline{Q}_{g,ij} + f_i + h_j}{\varepsilon}\right), \end{aligned} \tag{53}$$

where LSE denotes log-sum-exp. This is algebraically equivalent to equation 52 but avoids explicitly forming large entries of $\mathbf{K}_g$.

This factorization clarifies the role of entropy regularization in DECT. Without it, small IoU differences can induce nearly hard assignments, making instance credit sensitive to ambiguous or overlapping predictions. Entropy regularization instead produces the positive affinity kernel $\mathbf{K}_g$, allowing mass to be distributed among plausible matches before the marginal constraints reconcile competing assignments globally. Dustbin entries participate in the same normalization, so false-positive predictions and missed targets compete directly with real correspondences rather than being handled by separate heuristics.

Equation 13 converts the transport plan into instance-level credit. Real matches contribute in proportion to mask IoU, whereas dustbin assignments contribute no overlap credit. Thus, $P_g^\star$ resolves ambiguous correspondences globally while allowing unmatched predictions and targets to reduce the downstream reward and credit signals.

Representative transport and credit cases. The following examples use the fixed settings above. Matrices are rounded to four decimals; their last row and last column denote the null prediction and null target, respectively. We report the conditional instance credit e_i from equation 13. For the normalized correction $\widehat{e}_i$, we assume equal grounding/action output counts per predicted instance; the final branch advantage is then $A_{g,i}^b = A_g + 0.5|A_g|\widehat{e}_i$.

Ambiguous matching. For two nearly interchangeable predictions and targets,

$$Q = \begin{bmatrix} 0.70 & 0.68 \\ 0.68 & 0.70 \end{bmatrix}, \qquad P^\star \approx \begin{bmatrix} 0.1479 & 0.0991 & 0.0030 \\ 0.0991 & 0.1479 & 0.0030 \\ 0.0030 & 0.0030 & 0.4940 \end{bmatrix}. \tag{54}$$

The off-diagonal mass preserves the matching uncertainty. Both predictions receive $e = (0.6837, 0.6837)$ and therefore $\widehat{e} = (0, 0)$, avoiding an arbitrary instance-level preference; the rollout reward is $R_g = 0.6837$.

Duplicate predictions. For two predictions competing for one target,

$$Q = \begin{bmatrix} 0.88 \\ 0.62 \end{bmatrix}, \qquad P^\star \approx \begin{bmatrix} 0.3103 & 0.0231 \\ 0.0230 & 0.3103 \\ 0.0000 & 0.3333 \end{bmatrix}. \tag{55}$$

Most target mass is assigned to the stronger prediction, whereas the weaker duplicate is sent to the null target. The credits are $e = (0.8191, 0.0429)$ and $\widehat{e} = (1, -1)$, so the former receives $A_g + 0.5|A_g|$ and the latter $A_g - 0.5|A_g|$; the set reward is $R_g = 0.5747$.

Missed target. For one prediction and two targets,

$$Q = \begin{bmatrix} 0.84 & 0.12 \end{bmatrix}, \qquad P^\star \approx \begin{bmatrix} 0.3324 & 0.0001 & 0.0009 \\ 0.0009 & 0.3333 & 0.3325 \end{bmatrix}. \tag{56}$$

The second target is explained almost entirely by the null-prediction row. The existing prediction has $e_1 = 0.8376$ and, because it is the only prediction, $\widehat{e}_1 = 0$. The missed target instead lowers recall and the shared rollout reward to $R_g = 0.5584$, thereby affecting all sampled outputs through A_g even though no instance-specific span exists for the missing object.

A.9 DECT-GRPO Optimization Details

Credit-map normalization. For rollout g, the instance credit $e_{g,i}$ in equation 13 is broadcast to the grounding-token span $S_{g,i}$ and valid action indices $\mathcal{D}_{g,i}$. Because different instances can contribute different numbers of grounding-token and action-likelihood terms, the two branches normalize the credits separately. Let

$$w^y_{g,i} = |S_{g,i}|, \qquad w^a_{g,i} = |\mathcal{D}_{g,i}|, \qquad \overline{e}^b_g = \frac{\sum_i w^b_{g,i} e_{g,i}}{\sum_i w^b_{g,i}}, \tag{57}$$

for $b \in \{y, a\}$. The normalized instance credit is

$$\widehat{e}^b_{g,i} = \frac{e_{g,i} - \overline{e}^b_g}{\max_j |e_{g,j} - \overline{e}^b_g|}, \tag{58}$$

when the denominator exceeds 10^{-6}; otherwise, the correction is set to zero. The credit maps are constructed by assigning $\mathbf{c}^y_g[S_{g,i}] = \widehat{e}^y_{g,i}$ and $\mathbf{c}^a_g[\mathcal{D}_{g,i}] = \widehat{e}^a_{g,i}$. This output-count-weighted centering preserves the branch-wise mean advantage after instance-level modulation, as shown in equation 41. It requires valid, nonoverlapping grounding spans and valid-point masks for the action branch.

Sign-preserving modulation. For $A_g \neq 0$, the instance-modulated advantage satisfies

$$\frac{A^b_{g,\nu}}{A_g} = 1 + \lambda_c \, \mathrm{sgn}(A_g) \mathbf{c}^b_g[\nu] \in [1 - \lambda_c, \, 1 + \lambda_c].$$

Thus, for $0 \leq \lambda_c < 1$, the correction preserves the sign of the shared advantage. With the default $\lambda_c = 0.5$, the ratio lies in $[0.5, 1.5]$. When $A_g = 0$, the correction also vanishes. The mechanism therefore differentiates the strength of shared rollout feedback across instances without assigning independent advantage signs.

Empty outputs and set rewards. For positive-target inputs with no predicted objects, the rollout reward is set to

$$R_g = 0, \qquad M_g = 0, \; N > 0. \tag{59}$$

For no-target inputs, the reward depends only on the predicted object count,

$$R_g = \frac{1}{1 + M_g}, \qquad N = 0. \tag{60}$$

For these no-target examples, grounding advantages are multiplied by the configured weight α_{neg}. The action branch is disabled because the reward evaluates object count rather than contour quality, even when the rollout contains predicted objects. Missed targets have no corresponding grounding span or action trajectory and therefore influence optimization through target coverage, the set reward R_g, and the shared group-relative advantage A_g.

Likelihood ratios and clipped objectives. During rollout collection, we store grounding and action log-probabilities under the behavior policy. During replay, the current log-probabilities are recomputed on the same sampled outputs. Their likelihood ratios are

$$\rho^{y}_{g,n} = \exp\big(\ell^{y}_{g,n} - \ell^{y,\mathrm{old}}_{g,n}\big), \qquad \rho^{a}_{g,\xi} = \exp\Big(\ell^{a}_{g,\xi} - \ell^{a,\mathrm{old}}_{g,\xi}\Big). \tag{61}$$

Both branches use the clipped surrogate from equation 15, defined explicitly as

$$\mathcal{S}_{\mathrm{clip}}(\rho, A) = \min(\rho A,\ \mathrm{clip}(\rho, 1-c, 1+c)A). \tag{62}$$

For grounding, the instance-modulated token advantages from equation 14 yield

$$\mathcal{L}^{y}_{g} = -\frac{1}{L_g}\sum_{n=1}^{L_g} \mathcal{S}_{\mathrm{clip}}(\rho^{y}_{g,n}, A^{y}_{g,n}) + \beta_{\mathrm{KL}}\mathcal{K}_g. \tag{63}$$

For geometric actions,

$$\mathcal{L}^{a}_{g} = -\frac{1}{|\mathcal{D}_g|}\sum_{\xi\in\mathcal{D}_g} \mathcal{S}_{\mathrm{clip}}(\rho^{a}_{g,\xi}, A^{a}_{g,\xi}), \tag{64}$$

where $\mathcal{D}_g$ contains all valid two-dimensional actions in the rollout. Padding points do not contribute to the action likelihood or loss.

Grounding reference regularization. The grounding branch is regularized toward a reference policy. Let

$$d_{g,n} = \mathrm{sg}\big(\ell^{y,\mathrm{ref}}_{g,n}\big) - \ell^{y}_{g,n}. \tag{65}$$

We use

$$\mathcal{K}_g = \frac{1}{L_g}\sum_{n=1}^{L_g} \left(\exp(d_{g,n}) - d_{g,n} - 1\right). \tag{66}$$

This nonnegative exponential surrogate is applied only to grounding; the continuous action branch uses no reference-policy regularizer.

Fixed-sample replay. Rollout collection stores the sampled grounding output, pre-$\tanh$ action latents, behavior-policy log-probabilities, valid-point masks, boxes, and historical contour states. During replay, these sampled quantities remain fixed while the current VLIM, EASS, and GAD features and action means are recomputed with gradients enabled. Rewards, rasterized masks, transport plans, and credit maps are detached from gradient computation.

Consequently, the action branch is optimized through score-function gradients of the sampled continuous actions rather than differentiation through contour rasterization, mask rewards, or transport matching. Gradients from the action likelihood propagate through GAD and its trainable observation pathways, including EASS and the trainable VLIM features.

Update aggregation. Rollouts with zero group-relative advantage or without scoreable grounding outputs are omitted from the policy loss, including the grounding reference term. The action loss is additionally omitted when $M_g = 0$, $N = 0$, or $|\mathcal{D}_g| = 0$. Each remaining rollout retains its $1/G$ weight in equation 15; grounding and action losses are normalized independently before applying λ_a. Across gradient-accumulation steps, gradients are averaged over the accumulation count and across devices, then clipped before the optimizer update. Each sampled group is used for a single policy update.

A.10 Additional Ablation Studies

A.10.1 State Conditioning in Semantic Routing

Experimental setup. We compare four semantic-routing variants to distinguish learned layer selection from conditioning on static instance attributes and evolving contour geometry. All variants retain Spatial Visual Sampling (SVS), contour-aligned spatial re-observation, and the default action horizon $K = 2$ and $T = 4$. They follow the same Stage-II SFT protocol from the same Stage-I

initialization and are evaluated with identical grounding outputs and initial contours, as in Table 2. No RL post-training is used in this comparison.

The *predefined schedule* retains SVS and spatial re-observation but uses the predefined VLIM layer schedule instead of learned semantic routing. *Slot-conditioned routing* retains evolution–depth slot embeddings and the normalized evolution index, but masks instance-specific and dynamic geometric descriptors at the query input. *Static-instance-conditioned routing* additionally retains log box aspect ratio, log box area, and normalized point allocation. *Full state-conditioned routing* further includes contour irregularity and the mean and maximum displacement from the preceding evolution iteration, recovering Full EASS. These descriptors follow Appendix A.4.

For the three learned-routing variants, masked query dimensions are set to zero during both training and inference without changing the query-network width. Semantic-key construction, candidate layers, straight-through selection, the routing-prior schedule, and routing regularizers remain unchanged. Only routing-query inputs are ablated; geometry in SVS, contour tokens, and instance context is retained. Because semantic keys depend on VLIM features, slot-conditioned routing can still vary across image–expression inputs rather than imposing a dataset-wide fixed schedule.

Evaluation. We report gIoU and cIoU on gRefCOCO val under the common-grounding protocol of Table 2. We additionally report mean Boundary IoU on the fixed positive-target expression subset used in Figure 5, denoted bIoU_+. It is evaluated after eight action steps using expression-level union masks, original image resolution, and the boundary-width parameter of 2% of the image diagonal specified in Appendix A.1. Empty predictions receive zero and remain in the average.

Table 6: **State conditioning in semantic routing on gRefCOCO val (%).** All variants retain SVS and spatial re-observation. gIoU and cIoU use the validation split; bIoU_+ uses the fixed positive-target subset from Figure 5 at action step 8. The Full EASS boundary score is the Figure 5 endpoint (65.98%), rounded to one decimal place. Bold denotes the best score in each column.

Routing variant	gIoU ↑	cIoU ↑	bIoU_+ ↑
Predefined schedule	74.2	63.5	61.4
Slot-conditioned	74.7	64.2	62.2
Static-instance-conditioned	76.5	65.1	64.3
Full state-conditioned	**77.6**	**67.4**	**66.0**

Analysis. Table 6 shows modest gains from slot-conditioned routing over the predefined schedule, followed by further improvements when static instance attributes are included. Full state conditioning adds 1.1 gIoU, 2.3 cIoU, and 1.7 bIoU_+ points over the static variant. With SVS and spatial re-observation retained, this comparison supports the value of evolving contour descriptors for semantic routing beyond evolution–depth slots and static instance attributes. The boundary improvement on the fixed positive-target subset is consistent with more effective semantic conditioning during contour refinement.

A.10.2 Grounding and Contour-Action Optimization

Experimental setup. We ablate the two policy-gradient (PG) terms while retaining the complete architecture. Let $\mathcal{L}^{y}_{\mathrm{PG}}$ and $\mathcal{L}^{a}_{\mathrm{PG}}$ denote the aggregated grounding-token and contour-action clipped-surrogate losses, respectively, excluding reference regularization and the action coefficient. We use

$$\mathcal{L}_{\eta_y,\eta_a} = \eta_y \mathcal{L}^{y}_{\mathrm{PG}} + \eta_a \lambda_a \mathcal{L}^{a}_{\mathrm{PG}} + \beta_{\mathrm{KL}} \mathcal{K}. \tag{67}$$

The switches $(\eta_y, \eta_a) = (1, 0)$, $(0, 1)$, and $(1, 1)$ give *grounding-PG only*, *action-PG only*, and *joint DECT-GRPO*, respectively. $\mathcal{K}$ retains the original grounding reference penalty and aggregation rules in Appendix A.9 for all RL variants. SFT-only denotes the checkpoint before post-training.

All RL variants use the same SFT initialization, training data, rollout settings, transport reward, dustbin treatment, instance credits, and optimization budget. Both grounding and actions are sampled regardless of which PG term is retained. Loss coefficients and validity masks are unchanged, and no SFT losses are added. No-target rollouts receive no action PG loss. This is an objective ablation, not module freezing: shared VLIM features can change even when one PG term is absent.

End-to-end and fixed-grounding evaluation. Table 7(a) evaluates the full gRefCOCO validation split using each model's own grounding. Panel (b) fixes the same complete SFT grounding sequences, boxes, and initial contours for every model on the fixed positive-target subset used in Figure 5. Models recompute their own VLIM features, EASS observations, and eight-step contour actions without regenerating grounding outputs. Thus, upstream predictions are controlled, but learned observation pathways are not frozen.

On this subset, mIoU_+ averages expression-level union-mask IoU, while bIoU_+ follows the boundary protocol above. Empty predictions remain in both averages with zero scores; no model-specific filtering is applied. The SFT-only boundary reference is shared with Full EASS in Table 6. Panel (b) is a separate controlled evaluation, not a positive-target breakdown of panel (a).

Table 7: **Ablation of grounding and contour-action PG objectives (%).** (a) End-to-end evaluation on the full gRefCOCO validation split using each model's own grounding. (b) Eight-step contour evaluation on the fixed positive-target subset from Figure 5, with the same SFT grounding and initial contours for every model. Bold denotes the best score in each column.

	(a) End-to-end				(b) Fixed grounding	
Variant	gIoU ↑	cIoU ↑	N-acc ↑	F1@0.5 ↑	mIoU_+ ↑	bIoU_+ ↑
SFT only	77.6	67.4	74.6	73.8	78.4	66.0
Grounding-PG only	81.4	70.2	80.7	79.8	79.0	66.7
Action-PG only	79.8	69.5	76.3	76.1	80.5	68.5
Joint DECT-GRPO	**83.5**	**72.3**	**83.2**	**82.4**	**81.2**	**69.3**

Analysis. Both single-PG objectives improve over SFT, but their relative performance differs between the two evaluation settings. Grounding-PG-only training achieves higher end-to-end scores, whereas action-PG-only training yields better region overlap and boundary quality under fixed SFT grounding. Joint DECT-GRPO achieves the highest scores in both panels. Relative to grounding-PG only, it improves end-to-end gIoU and F1@0.5 by 2.1 and 2.6 points, respectively. Under fixed grounding, it retains gains of 2.2 mIoU_+ and 2.6 bIoU_+ points, supporting a refinement benefit from including the action objective beyond changes in predicted boxes and referent counts at evaluation. Joint training also improves over action-PG only under both protocols, supporting the retention of both PG terms within the evaluated training budget. These comparisons support the two training objectives without attributing their gains to isolated modules.

A.10.3 Controlled Transfer to HiMTok-8B

Objective. To examine whether ContourVLA remains effective with a different pretrained vision–language backbone, we adapt it to HiMTok-8B (Wang et al., 2025a). This study controls the backbone initialization and downstream training corpus while omitting the original ContourVLA Stage-I grounding pretraining described in Appendix A.1.

Compared variants. **A: Original HiMTok-8B.** We evaluate the released checkpoint without additional training, retaining its native mask-token prediction and mask de-tokenizer. This serves as the unadapted reference.

B: HiMTok-8B with continued SFT. Starting from A, we retain the original segmentation formulation and continue teacher-forced training on the shared downstream corpus. Ground-truth masks are encoded with the pretrained hierarchical mask tokenizer, and the native answer sequences are optimized using token-level cross-entropy. The mask tokenizer and de-tokenizer remain fixed. No contour-action modules or DECT-GRPO are introduced.

C: ContourVLA with the HiMTok backbone. Starting from the same vision–language checkpoint, we replace the native mask-token segmentation pathway with EASS and GAD. Stage-II SFT jointly trains structured instance-box or no-target prediction and contour evolution, followed by Stage-III DECT-GRPO. Spatial features and multilevel hidden states are obtained from the HiMTok backbone; no external grounding or segmentation model is introduced.

Controlled training protocol. B and C initialize their shared visual encoder, visual-to-language projector, language model, and inherited token parameters from the same HiMTok checkpoint. Newly introduced modules in C are initialized anew; no trained weights, teacher outputs, or cached features from the original ContourVLA pipeline are transferred. Both methods use the training splits of gRefCOCO, RefCOCO, RefCOCO+, and RefCOCOg, with the same sampling distribution and image preprocessing. Grounding, mask-token, and contour targets are derived from the same source annotations, without additional external supervision. Target representations and learning objectives differ by design, while the update policy for shared backbone parameters is identical during SFT.

B and C are trained using the same hardware configuration, including the same GPU type and number of GPUs. Hyperparameters and checkpoints are selected using gRefCOCO validation gIoU. The selected checkpoint from each run is evaluated on both test splits without test-based tuning or further adaptation. The hardware control does not imply matched total computation or equal numbers of input presentations.

Evaluation. We follow the expression-level union-mask evaluation in Appendix A.1 and report gIoU and cIoU on gRefCOCO val, testA, and testB. Each method uses its own predictions at inference, without ground-truth boxes or referent counts. B and C are trained with three independent random seeds and reported as mean $\pm$ sample standard deviation. A is evaluated deterministically using the original checkpoint.

Table 8: **Controlled transfer to HiMTok-8B on gRefCOCO (%).** A is the original HiMTok-8B checkpoint, B uses continued SFT, and C applies ContourVLA with the HiMTok backbone. B and C share the same pretrained backbone initialization, downstream corpus, and hardware configuration. B and C report mean $\pm$ sample standard deviation over three independent training runs; A is a single-checkpoint reference. Bold denotes the highest mean in each column.

	val		**testA**		**testB**	
Variant	gIoU ↑	cIoU ↑	gIoU ↑	cIoU ↑	gIoU ↑	cIoU ↑
A	72.1	70.4	73.4	74.9	71.7	71.9
B	73.6 ± 0.2	70.9 ± 0.3	74.7 ± 0.2	75.4 ± 0.4	72.5 ± 0.3	$\mathbf{72.3} \pm 0.3$
C	$\mathbf{80.4} \pm 0.1$	$\mathbf{72.0} \pm 0.4$	$\mathbf{78.8} \pm 0.4$	$\mathbf{77.0} \pm 0.2$	$\mathbf{73.9} \pm 0.3$	72.2 ± 0.2

Results and analysis. Table 8 shows that continued SFT improves the original HiMTok checkpoint on all six reported metrics. Relative to this adapted baseline B, C further improves mean gIoU by 6.8, 4.1, and 1.4 points on val, testA, and testB, respectively. These gains are large relative to the reported standard deviations across three training seeds. C also improves cIoU by 1.1 and 1.6 points on val and testA, while its testB cIoU is slightly lower than B's (72.2 ± 0.2 vs. 72.3 ± 0.3). The advantage is therefore consistent in gIoU rather than uniform across all metrics.

Because B and C share the same pretrained backbone initialization and downstream training corpus, these results support the cross-backbone applicability of the complete ContourVLA method without requiring its original Stage-I grounding pretraining. The comparison evaluates the complete architecture and training recipe rather than isolating EASS, GAD, or DECT-GRPO individually. It does not establish independence from pretraining in general, out-of-distribution generalization, or an advantage under matched total computation, and it does not quantify the contribution of Stage-I pretraining to the original Table 1 results.

A.11 Limitations and Scope

Scope of the closed loop. In the current formulation, "closed-loop VLA" specifically denotes feedback between the evolving geometry of initialized contours and state-conditioned visual–semantic readouts. VLIM determines the instance set and initial boxes before contour evolution; the subsequent action space refines these instances but does not regenerate boxes or introduce instance-birth and instance-deletion actions within the same inference rollout. Consequently, an initially missed referent—including the $M = 0$ case—cannot be recovered through later contour updates. DECT-

GRPO improves the grounding policy across training rollouts, but is not intended to imply online bidirectional revision of grounding and segmentation. This scoped design isolates geometric correction and keeps inference stable; extending the action space with re-grounding and cardinality-changing operations is a natural direction for a broader closed-loop policy.

Contour representation. The fixed ordered-vertex state provides efficient, interpretable boundary updates, but retains the initialized contour topology and point budget. Highly disconnected regions, holes, or exceptionally thin structures may therefore benefit from topology-changing actions or adaptive vertex insertion. These extensions are complementary to EASS and DECT-GRPO rather than changes to their underlying perception and credit-assignment principles.