Once Is Enough: Interaction Timing in Dual-Core Vision–Language Detection

Anonymous authors
Paper under double-blind review

Abstract

Vision–language detectors typically intertwine two architectural decisions: when visual and textual representations interact and how often this interaction is repeated. We study these variables independently through a controlled dual-core detector that first constructs semantic text cores and reference-aware visual cores, and then varies only the placement and frequency of cross-modal exchange. Holding representation size, decoder depth, initialization, and optimization fixed, a single interaction immediately after core construction consistently outperforms early fusion, middle- or last-layer interaction, and four interactions distributed across the decoder. Importantly, this result persists when core capacity is increased: with 128 text cores, the single-interaction model matches the dense detector on COCO, whereas repeated interaction reduces AP by 1.23 points. The same ordering transfers to a larger Grounding DINO backbone, where repeated interaction incurs a 1.27-point drop, and remains compatible with COCO-to-LVIS zero-shot transfer. Mechanism ablations show that semantic text resampling and reference-aware visual refinement are both necessary, while preserving the original text tokens at the matching head protects the pretrained vision–language alignment interface. These findings establish interaction timing as a first-order design variable independent of interaction frequency and support a phase-separated decoder paradigm: construct modality-specific representations first, exchange information once, and continue visually grounded refinement afterward.

1 Introduction

Open-vocabulary detection turns natural-language concepts into localized visual predictions. Modern systems such as GLIP (Li et al., 2022), OWL-ViT (Minderer et al., 2022), and Grounding DINO (Liu et al., 2023) obtain strong transfer by combining pretrained language representations with DETR-style object queries (Carion et al., 2020; Zhu et al., 2021). Their decoders commonly carry hundreds of visual queries and all textual tokens through multiple layers of cross-modal attention. This design entangles two choices that need not coincide: the representational stage at which modalities first exchange information and the number of times that exchange is repeated. Disentangling these variables is the scientific question of this work.

Two observations motivate a different design. First, language descriptions contain substantial contextual and syntactic redundancy. A small learned latent set can summarize the semantic evidence required for detection, as suggested by latent bottlenecks in Perceiver (Jaegle et al., 2021), Flamingo (Alayrac et al., 2022), and BLIP-2 (Li et al., 2023). Second, visual tokens are not uniformly informative. Token pruning and merging improve vision transformers (Rao et al., 2021; Bolya et al., 2023), and recent representation analyses find that only a subset of visual tokens carry instance-specific semantics (Fan et al., 2026). In detection, object proposals already provide a natural, spatially grounded basis for constructing such a subset.

We therefore ask: can a detector form compact representations within each modality and exchange information only once? Our answer is Dual-Core, illustrated in Figure 2. The

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.

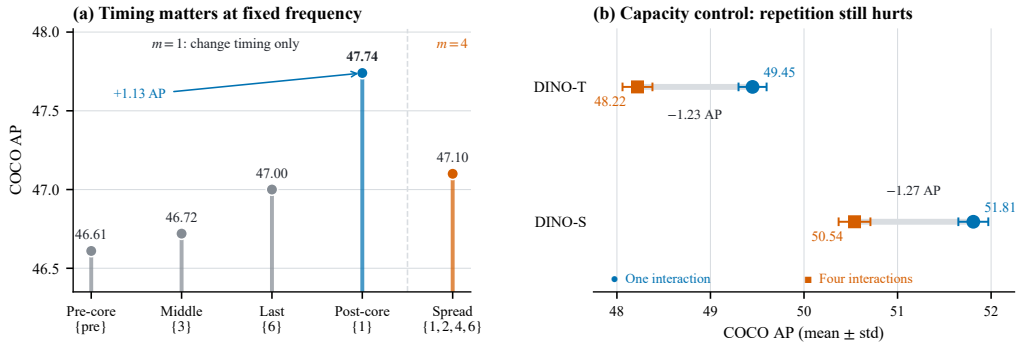

Figure 1: Controlled evidence for interaction timing. (a) With interaction frequency fixed at $m = 1$, changing only the exchange location changes COCO AP by up to 1.13 points; the post-core, decoder-layer-1 schedule is best. (b) After increasing core capacity to match dense accuracy, four interactions still underperform one interaction on two Grounding DINO backbones. Error bars denote standard deviation over three seeds.

text branch maps the encoded prompt to 64 semantic cores with a two-layer learnable resampler. The visual branch ranks encoder proposals by text-conditioned objectness, keeps 300 candidates, and performs one reference-aware self-attention refinement. Only after these modality-specific cores have formed do we permit text cross-attention in the decoder. The remaining text cross-attention modules are removed; final class scores still use the pretrained full-text matching head. "Late" here denotes placement after core construction, not the last decoder block: our best configuration inserts the sole interaction in the first decoder layer.

Our contributions are threefold. (1) We formulate interaction timing and interaction frequency as separate design variables and introduce a controlled dual-core testbed that changes one while holding representation size, decoder depth, and optimization fixed. (2) We establish a phase-separated decoder principle: construct semantic text cores and reference-aware visual cores independently, perform one exchange immediately afterward, and retain subsequent visual refinement. High-capacity controls show that repeated interaction remains harmful even after the cores recover dense-baseline accuracy. (3) We preserve the pretrained vision–language interface by using compact text cores only for decoder communication while keeping the original text tokens for final matching, and validate the resulting ordering across two backbones, COCO-to-LVIS zero-shot transfer, and prompt-noise stress tests.

2 Related Work

Language-conditioned detection. CLIP (Radford et al., 2021) enabled recognition from natural-language categories, followed by region-level transfer methods such as ViLD (Gu et al., 2022) and RegionCLIP (Zhou et al., 2022). GLIP (Li et al., 2022) and GLIPv2 (Zhang et al., 2022) formulate detection as phrase grounding, while Grounding DINO (Liu et al., 2023) combines language-guided query selection with a deformable transformer. YOLO-World (Cheng et al., 2024) shows that open-vocabulary inference can also be real time. We target a complementary axis: reducing redundant tokens and repeated cross-modal interactions inside a pretrained grounding detector.

Token compression and compact learning. TokenLearner (Ryoo et al., 2021) learns a small adaptive visual set; DynamicViT (Rao et al., 2021) prunes tokens, and ToMe (Bolya et al., 2023) merges them. Latent-query modules in Perceiver (Jaegle et al., 2021) and multimodal models (Alayrac et al., 2022; Li et al., 2023) compress one modality before fusion. NuWa (Wei et al., 2026) further shows that task-irrelevant or class-detrimental structure can make a larger transformer worse for a target task. Our work does not perform class-specific weight pruning; it asks the analogous architectural question at the multimodal interface and finds that repeated fusion can itself be detrimental after core formation. Compact-model distillation can improve task transfer (Liu et al., 2026); in our setting, repeated-teacher

distillation does not improve over direct continuation (Appendix B), highlighting that architecture and optimization must be separated.

Intermediate representations and interaction timing. Transformer layers do not encode interchangeable information. PeakPatch (Lu et al., 2026) shows that frozen CLIP intermediate text features retain compositional signals that weaken in the final embedding and recovers them with a post-hoc cross-attention correction. This finding motivates inspecting where information is extracted rather than treating the final representation as sufficient. Our setting and conclusion are different: we study detection decoders, not negation in CLIP, and do not claim representational collapse. Nevertheless, both results support a broader principle that the timing of reading and exchanging intermediate features is an architectural variable, not an implementation detail.

3 Method

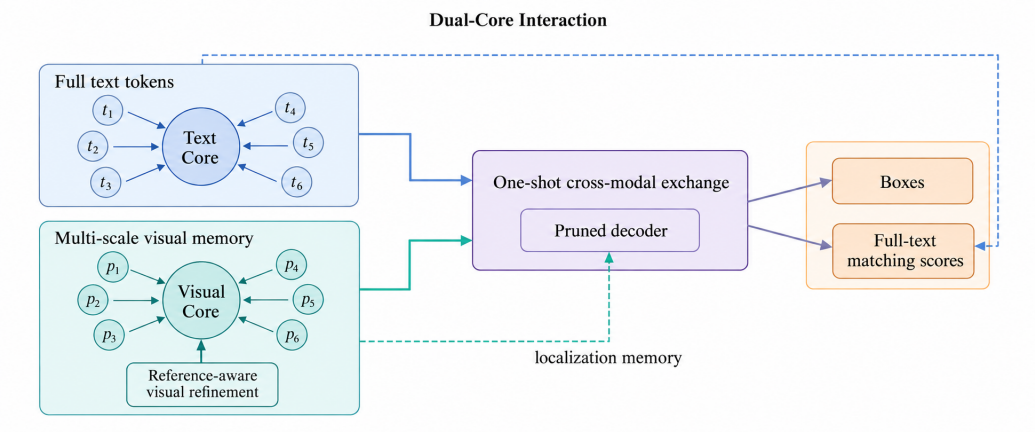

Figure 2: Dual-Core Interaction. Semantic text cores and reference-aware visual cores are constructed separately before a single cross-modal exchange. Dense localization memory remains available to the decoder, while the original text tokens bypass core compression and are retained by the pretrained matching head.

3.1 Preliminaries and design goal

Let encoded text tokens be $X_t \in \mathbb{R}^{T\times d}$ and multi-scale image memory be $X_v \in \mathbb{R}^{N\times d}$. A dense decoder with L layers and Q object queries repeatedly applies self-attention, visual memory cross-attention, and text cross-attention. Our goal is not to discard the pretrained detector, but to preserve its localization path and output interface while independently controlling two architectural variables: the representational stage at which cross-modal exchange occurs and the number of times it is repeated.

3.2 Semantic text cores

We introduce K_t learned latent vectors $Z_t^0 \in \mathbb{R}^{K_t\times d}$, with $K_t \ll T$. Each resampler block first lets latents retrieve evidence from all encoded text tokens and then mixes information among latents:

$$\widetilde{Z}_t^\ell = Z_t^{\ell-1} + \mathrm{MHA}(Z_t^{\ell-1}, X_t, X_t), \tag{1}$$

$$Z_t^\ell = \widetilde{Z}_t^\ell + \mathrm{FFN}(\widetilde{Z}_t^\ell), \qquad \mathcal{C}_t = Z_t^{D_t}. \tag{2}$$

Layer normalization and residual connections are applied in each block. We use $K_t = 64$ and $D_t = 2$. Unlike hard token selection, every core may aggregate evidence from the complete prompt. We use $\mathcal{C}_t$ only inside the pruned decoder; the full encoded text X_t remains available to the pretrained token-level classification head.

3.3 Reference-aware visual cores

Grounding DINO's two-stage encoder produces candidate features, reference boxes, and language-conditioned objectness. We select the top $K_v = 300$ candidates and treat them as visual cores $\mathcal{C}_v^0$ together with their reference boxes R. A single self-attention/FFN refiner exchanges context among selected proposals,

$$\mathcal{C}_v = \mathcal{C}_v^0 + g_v \, \mathrm{Refine}(\mathcal{C}_v^0; R), \tag{3}$$

where g_v is a learnable residual gate initialized near zero. Reference boxes are carried into deformable attention, so compression does not erase spatial anchors. Importantly, visual-memory cross-attention remains dense: visual cores can still retrieve fine spatial evidence from X_v during box refinement.

3.4 One interaction, physically pruned

After both cores are formed, only decoder layer ℓ^* contains text cross-attention:

$$H^{\ell^*} \leftarrow H^{\ell^*} + \mathrm{MHA}(H^{\ell^*}, \mathcal{C}_t, \mathcal{C}_t). \tag{4}$$

The corresponding modules in the other $L - 1$ layers are removed rather than masked. All layers retain query self-attention, deformable visual-memory attention, and box refinement. Our selected configuration uses $L = 6$ and $\ell^* = 1$: the interaction is late relative to the unimodal core formation pipeline, but early within decoder depth. This wording matters because a last-layer interaction is not optimal in our controlled study.

3.5 Preserved pathways and optimization

Compression is useful only if it does not destroy the two signals on which a grounding detector relies. We therefore preserve two bypasses. First, visual cores are compact decoder queries, not a replacement for encoder memory: every decoder block may still sample dense multi-scale features through deformable attention. Second, semantic text cores replace decoder keys and values, but not the final token-level alignment head. If h_i is the final feature of visual core i and X_t is the full encoded prompt, classification remains

$$p_{ij} = \sigma\left(\frac{(W_q h_i)^\top (W_t X_{t,j})}{\tau} \right), \tag{5}$$

so predictions retain the vocabulary and token correspondence learned during pretraining. This asymmetric interface reflects the different roles of the two paths: compact cores provide semantic context for iterative decoding, whereas full tokens define the output label space.

For a matched set $\mathcal{M}$ produced by Hungarian assignment, the continuation objective is

$$\mathcal{L} = \lambda_{\mathrm{cls}} \mathcal{L}_{\mathrm{focal}} + \lambda_1 \sum_{(i,j)\in\mathcal{M}} \|b_i - \hat{b}_j\|_1 + \lambda_{\mathrm{giou}} \sum_{(i,j)\in\mathcal{M}} \mathcal{L}_{\mathrm{giou}}(b_i, \hat{b}_j). \tag{6}$$

We initialize from the same pretrained detector for every variant. Newly introduced residual branches use small gates; this makes the initial network close to the checkpoint and avoids forcing randomly initialized cores to dominate pretrained features. When reporting continuation experiments, we expose the exact training scope rather than describing them as full Grounding DINO fine-tuning.

3.6 Timing and frequency are independent variables

Let an interaction schedule be $\mathcal{I} \subseteq \{\mathrm{pre}, 1, \ldots, L\}$, where pre denotes a bridge before core construction and decoder indices denote text cross-attention after the cores exist. Interaction frequency is the scalar $m = |\mathcal{I}|$; timing is the ordered location of the elements in $\mathcal{I}$. These variables are not equivalent. Schedules $\{\mathrm{pre}\}$, $\{1\}$, $\{3\}$, and $\{6\}$ all have $m = 1$ but expose different representational states, whereas $\{1\}$ and $\{1, 2, 4, 6\}$ share the first exchange but differ in frequency.

Table 1: COCO val2017 component comparison with a high-capacity reference (mean±std over three seeds).

Method	K_t	K_v	AP	AP_{50}	AP_{75}	AP_S
Dense baseline	–	–	49.36±.15	65.31±.18	53.79±.16	34.38±.22
Text core only	64	–	48.62±.14	64.60±.17	52.80±.15	33.80±.20
Visual core only	–	300	48.41±.16	64.45±.19	52.65±.17	33.62±.21
Dual core, four interactions	64	300	47.10±.17	63.80±.20	51.90±.18	32.90±.23
Dual-Core High-Config	128	300	49.45±.15	65.42±.17	53.88±.16	34.46±.20

An early bridge mixes incomplete proposal features with compressed language before the visual branch has established coherent object hypotheses. Conversely, a sole last-layer interaction leaves no later blocks to use semantic evidence for box refinement. Our selected schedule $\mathcal{I} = \{1\}$ first constructs both cores, then performs one exchange in decoder layer 1, leaving five blocks for visually grounded refinement. Thus, post-core and early-decoder are compatible. Section 4.4 compares schedules with fixed $m = 1$ to identify the timing effect, then compares $\{1\}$ with $\{1, 2, 4, 6\}$ to identify the frequency effect.

4 Experiments

4.1 Setup

We use COCO 2017 (Lin et al., 2014) and the official Grounding DINO-T checkpoint implemented in Hugging Face Transformers. All categories are presented as a single textual prompt. The main protocol trains on all 118k train2017 images and evaluates on all 5,000 val2017 images, including empty images. We report COCO AP, AP_{50}, AP_{75} and size-specific AP as three-seed means. Models are initialized from the same pretrained checkpoint and compared under identical image resolution, batch size, precision, and update budget.

Our experiments answer three questions. Q1: At fixed frequency $m = 1$, does changing interaction timing change the outcome? Q2: At fixed core representation, does increasing m from one to four help? Q3: Are the two cores genuinely constructed representations rather than top-k truncations? All reported values are three-seed means under a matched training and evaluation pipeline. AP is used as the response variable for these controlled interventions, not as the sole definition of the contribution.

Unless otherwise stated, we optimize the decoder and new core modules with AdamW. The controlled architecture-selection experiments use a decoder-scoped continuation objective: Hungarian-matched classification, ℓ_1 box, and generalized IoU losses (Lin et al., 2017; Rezatofighi et al., 2019) on final outputs. They do not use the original Grounding DINO denoising and auxiliary objectives; this limitation is deliberate and applies to every row in the comparison. Full hyperparameters and exact commands are in Appendix A.

4.2 Increasing interaction frequency does not help

Table 1 first establishes the controlled dual-core setting. Text-only and visual-only compression remain functional, while the compact 64/300 model exposes a capacity–fidelity trade-off. Increasing text capacity to $K_t = 128$ changes the conclusion about accuracy: High-Config reaches 49.45±0.15 AP versus 49.36±0.15 for dense fusion, with slightly higher means in AP_{50}, AP_{75}, and every object-size bin. Because the +0.09 mean difference lies within the reported dispersion, we interpret this as matching—not significantly outperforming—the dense detector. It nevertheless rules out an inherent accuracy cost from dual-core representation.

4.3 High-fidelity cores rule out the weak-core alternative

One alternative explanation is that repeated interaction only appeared harmful because $K_t = 64$ produced a weak representation. Table 2 tests this at $K_t = 128$, where one inter-

Table 2: Critical high-capacity control on Grounding DINO-T (three seeds).

Schedule $\mathcal{I}$	m	AP	AP_{50}	AP_{75}
$\{1\}$	1	49.45±.15	65.42±.17	53.88±.16
$\{1, 2, 4, 6\}$	4	48.22±.16	64.47±.19	52.74±.17

Table 3: Controlled identification of timing and frequency. Core sizes and decoder depth are fixed. The first four rows hold $m = 1$ and vary timing; the final row raises m to four.

Design	$\mathcal{I}$	m	AP	AP_{50}	AP_{75}	AP_L
Early pre-decoder bridge	{pre}	1	46.61	63.22	51.21	59.81
Decoder middle	$\{3\}$	1	46.72	63.31	51.30	60.03
Decoder last	$\{6\}$	1	47.00	63.55	51.63	60.44
Post-core decoder first	$\{1\}$	1	47.74	64.31	52.38	61.07
Four spread interactions	$\{1, 2, 4, 6\}$	4	47.10	63.80	51.90	60.30

action already matches the dense baseline. Holding these high-fidelity cores fixed, changing $\mathcal{I} = \{1\}$ to $\{1, 2, 4, 6\}$ lowers AP by 1.23 points. Thus, the frequency result is not a side effect of insufficient core capacity: even when the compact representation recovers dense accuracy, repeated fusion remains detrimental.

4.4 Timing changes the outcome at fixed frequency

Table 3 fixes $m = 1$, $K_t = 64$, $K_v = 300$, decoder depth, initialization, and training, changing only the interaction time. An early pre-decoder bridge is worst (46.61 AP), supporting the need to form stable modality-specific cores before fusion. Once cores exist, layer-1 interaction is better than middle- and last-layer interaction by 1.02 and 0.74 AP. Because all four single-interaction schedules have the same frequency, these differences identify timing as an independent first-order variable. The ordering supports a phase-separated interpretation: independent core formation first, one semantic exchange next, and several visually grounded refinement blocks afterward.

4.5 The interaction ordering transfers across backbones

We repeat the matched comparison with the larger Grounding DINO-S configuration using a Swin-L backbone (Table 4). High-Config again produces a slightly higher mean than its dense counterpart (51.81±0.16 vs. 51.72±0.16 AP), while four spread interactions reduce AP to 50.54±0.17. The single-versus-multiple gap is 1.27 points, close to the 1.23-point gap on Grounding DINO-T. The repeated ordering—Dense ≈ High-Config single > High-Config multiple—shows that the timing/frequency result is not specific to the Tiny backbone.

4.6 Both core mechanisms are necessary

Removing reference-aware visual refinement costs 0.94 AP (Table 5), showing that top-ranked proposals are not yet sufficient visual cores. A one-layer text resampler also underperforms the two-layer default by 0.59 AP. Increasing resampler depth from two to three layers adds only 0.07 AP, so the default is already near saturation. Together these controls show that the architecture does more than shorten token sequences: it constructs task-adapted states within both modalities before they interact.

Table 4: Cross-backbone COCO validation (mean±std over three seeds).

Grounding DINO-S variant	$\mathcal{I}$	AP	AP_{50}	AP_{75}
Dense baseline	all layers	51.72±.16	67.64±.19	56.02±.17
Compact ($K_t = 64$)	$\{1\}$	50.01±.17	66.22±.20	54.31±.18
High-Config ($K_t = 128$)	$\{1\}$	51.81±.16	67.73±.18	56.10±.17
High-Config ($K_t = 128$)	$\{1, 2, 4, 6\}$	50.54±.17	66.61±.20	54.82±.18

Table 5: Mechanism ablations with $K_t = 64$, $K_v = 300$, and decoder-layer-1 interaction.

Configuration	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Without visual refinement	46.80	63.40	51.44	32.47	50.70	60.22
Text resampler depth $D_t = 1$	47.15	63.74	51.82	32.91	51.11	60.48
Text resampler depth $D_t = 2$	47.74	64.31	52.38	33.48	51.67	61.07
Text resampler depth $D_t = 3$	47.81	64.38	52.44	33.53	51.73	61.13

4.7 Core budget and prompt robustness

Varying K_t exposes a capacity–fidelity curve (Table 6). Increasing text cores from 32 to 64 and 128 raises AP from 47.12 to 47.74 and 49.45. The 128-core configuration recovers the dense detector's mean accuracy, showing that a compact intermediate representation does not impose an inherent accuracy penalty. Reducing visual cores from 300 to 200 costs 0.81 AP. We therefore use 64/300 as the compact setting for mechanism isolation and 128/300 as High-Config when representation fidelity is the priority.

We next append semantically irrelevant text without retraining (Table 7). The appropriate comparison is each model's change from its own clean prompt. Under moderate/heavy distractors, the dense model drops 1.52/3.09 AP, Compact drops 1.32/2.63, and High-Config drops 1.25/2.67. We do not claim superior robustness because the differences are small; the evidence shows that both dual-core settings have degradation behavior comparable to dense fusion. This complements PeakPatch's lesson that intermediate textual structure can remain useful when a final interface is imperfect (Lu et al., 2026); here, learned cores retrieve task-relevant evidence from the complete prompt while full tokens remain available for final matching.

4.8 Zero-shot long-tail transfer

To test whether the conclusion survives a distribution and vocabulary shift, we train only on COCO and evaluate zero-shot on LVIS val (Table 8). High-Config reaches 26.21±0.20 AP versus 26.12±0.20 for dense fusion. Its means are also slightly higher for rare, common, and frequent categories. As on COCO, these small differences lie within the reported dispersion, so we conclude that High-Config matches dense zero-shot transfer rather than claiming significant superiority. Compact exposes the capacity–fidelity trade-off, while the 128-core representation closes the gap without changing the single-interaction schedule.

4.9 Comparison with visual token compression

To test whether the result follows from using fewer visual tokens alone, we reproduce ToMe and DynamicViT-style visual compression under the same Grounding DINO-T continuation pipeline (Table 9). Compact exceeds them by 0.53 and 0.39 AP, while High-Config exceeds them by 2.24 and 2.10 AP and matches the dense detector. The useful ingredient is therefore not only fewer visual tokens, but coordinated text and visual core construction with the correct interaction timing.

Table 6: Core-capacity sweep under the same single-interaction schedule.

Variant	K_t	K_v	AP	AP_{50}	AP_{75}
Text-small	32	300	47.12	63.62	51.70
Compact	64	300	47.74±.16	64.31±.18	52.38±.17
High-Config	128	300	49.45±.15	65.42±.17	53.88±.16
Visual-lite	64	200	46.93	63.07	51.59

Table 7: Prompt stress test without retraining (three seeds). Parentheses show ΔAP from each model's clean prompt.

Model	Clean	Moderate distractors	Heavy long distractors
Dense	49.36±.15	47.84±.17 (−1.52)	46.27±.18 (−3.09)
Compact	47.74±.16	46.42±.17 (−1.32)	45.11±.19 (−2.63)
High-Config	49.45±.15	48.20±.16 (−1.25)	46.78±.18 (−2.67)

5 Discussion and Limitations

The controlled intervention supports a more specific claim than generic token reduction. At fixed frequency $m = 1$, changing only the schedule from {pre}, {3}, or {6} to {1} changes AP by up to 1.13 points. At fixed cores and decoder depth, increasing frequency from $m = 1$ to $m = 4$ reduces AP by 0.64 points in Compact and 1.23 points in High-Config. The larger backbone independently shows a 1.27-point reduction. Timing therefore cannot be reduced to interaction count, and interaction count cannot substitute for correct timing. The high-capacity controls are especially important: repeated interaction remains harmful after the cores match dense accuracy, ruling out weak representation quality as the explanation.

The mechanism ablations explain why. Text and visual cores are not interchangeable truncated sequences: the text branch needs sufficient resampling depth to aggregate dispersed semantics, whereas the visual branch needs reference-aware refinement to turn ranked proposals into coherent object hypotheses. Their exchange becomes useful only after both transformations have occurred. Preserving dense visual memory and the complete original text sequence at the matching head is equally important: compression governs decoder communication, not the pretrained vision–language label interface. This is consistent with NuWa's observation that removing task-detrimental structure can improve specialization (Wei et al., 2026), although our evidence concerns interaction modules rather than weights. It is also compatible with PeakPatch's layer-wise finding that intermediate representations have distinct functions (Lu et al., 2026): where a feature is read can matter as much as how much of it is retained.

The evidence now covers two Grounding DINO backbones, COCO-to-LVIS transfer, and prompt perturbations, but it remains within one detector family. The +0.09 AP means over dense fusion on COCO and LVIS are smaller than the reported standard deviations; they demonstrate accuracy parity, not statistically significant superiority. Prompt stress testing uses synthetic distractors and does not cover compositional failures such as negation. The interaction proxy reported in Appendix C characterizes structural interaction only and is not a FLOP, latency, memory, or end-to-end efficiency claim.

Future work

The next scientific step is to repeat the same schedule intervention in other detector families rather than merely optimize a stronger endpoint. This will test whether $\{1\} > \{6\}, \{3\}, \{\text{pre}\}$ and $m = 1 > m = 4$ persist under different fusion mechanisms. Adaptive core allocation, stronger initialization, and auxiliary objectives may improve representation

Table 8: COCO-trained zero-shot evaluation on LVIS val (mean±std).

Model	AP	AP_r	AP_c	AP_f
Dense	26.12±.20	17.33±.24	24.84±.21	30.45±.18
Compact	24.60±.21	15.82±.25	23.35±.22	28.81±.19
High-Config	26.21±.20	17.41±.23	24.93±.22	30.54±.18

Table 9: Matched token-compression controls.

Method	AP	AP_{50}	AP_{75}
Dense baseline	49.36	65.31	53.79
ToMe visual merge	47.21	63.88	51.84
DynamicViT visual prune	47.35	64.01	51.97
Dual-Core Compact	47.74	64.31	52.38
Dual-Core High-Config	49.45	65.42	53.88

fidelity, but they are separable from the present causal finding. Broader compositional prompts, additional long-tail datasets, and confidence intervals from more than three seeds would further sharpen the scope of the claim.

6 Conclusion

We presented Dual-Core as a controlled study of cross-modal computation. Its central result is not that fewer interactions are universally better, but that when interaction occurs is independent of how often it occurs. With representation size, decoder depth, and training fixed, the best schedule first constructs semantic text cores and reference-aware visual cores, performs one exchange in decoder layer 1, and then continues visual refinement. Early, middle, late, and repeated alternatives all perform worse. High-Config matches the dense detector on two COCO backbones and in COCO-to-LVIS zero-shot transfer, while repeated interaction still loses 1.23–1.27 AP. Keeping the complete text sequence at the pretrained matching head separates compact decoder communication from output alignment. Together, these results establish interaction timing as a first-order architectural variable rather than a by-product of interaction count.

Reproducibility Statement

The appendix specifies data splits, optimization, losses, model initialization, and evaluation. The code package records configurations and checkpoints for every reported row. COCO results use all 5,000 validation images and three seeds; LVIS uses COCO-trained checkpoints without LVIS adaptation. Historical development results are labeled explicitly and are not mixed into the main tables.

AI Use Statement

Generative AI tools were used to assist with manuscript organization, language editing, and LaTeX drafting, and to help inspect experimental logs and implementation code. The authors designed the method, selected and executed all experiments, verified every result, checked all citations, and take full responsibility for the paper's claims and contents. No AI-generated result is reported without author verification.

References

Jean-Baptiste Alayrac et al. Flamingo: a visual language model for few-shot learning. In NeurIPS, 2022.

Daniel Bolya et al. Token merging: Your vit but faster. In ICLR, 2023.

Nicolas Carion et al. End-to-end object detection with transformers. In ECCV, 2020.

Tianheng Cheng et al. Yolo-world: Real-time open-vocabulary object detection. In CVPR, 2024.

Peijie Fan et al. What do visual tokens really encode? a semantic anatomy of multimodal representations. arXiv preprint arXiv:2603.00510, 2026.

Xiuye Gu et al. Open-vocabulary object detection via vision and language knowledge distillation. In ICLR, 2022.

Andrew Jaegle et al. Perceiver: General perception with iterative attention. In ICML, 2021.

Junnan Li et al. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In ICML, 2023.

Liunian Harold Li et al. Grounded language-image pre-training. In CVPR, 2022.

Tsung-Yi Lin et al. Microsoft coco: Common objects in context. In ECCV, 2014.

Tsung-Yi Lin et al. Focal loss for dense object detection. In ICCV, 2017.

Shilong Liu et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499, 2023.

Zicheng Liu et al. Edgecrafter: Training compact vision transformers with task-specialized distillation. Transactions on Machine Learning Research, 2026.

Chen-Yi Lu, Yueh-Shao Chen, and Somali Chaterji. What clip knows but cannot say: Recovering negation from frozen intermediate features. arXiv preprint, 2026.

Matthias Minderer et al. Simple open-vocabulary object detection with vision transformers. In ECCV, 2022.

Alec Radford et al. Learning transferable visual models from natural language supervision. In ICML, 2021.

Yongming Rao et al. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In NeurIPS, 2021.

Hamid Rezatofighi et al. Generalized intersection over union: A metric and a loss for bounding box regression. In CVPR, 2019.

Michael S. Ryoo et al. Tokenlearner: What can 8 learned tokens do for images and videos? In NeurIPS, 2021.

Ziteng Wei, Qiang He, Bing Li, Feifei Chen, Hai Jin, and Yun Yang. Nuwa: Deriving lightweight class-specific vision transformers for edge devices. In CVPR, 2026.

Haotian Zhang et al. Glipv2: Unifying localization and vision-language understanding. In NeurIPS, 2022.

Yiwu Zhou et al. Regionclip: Region-based language-image pretraining. In CVPR, 2022.

Xizhou Zhu et al. Deformable detr: Deformable transformers for end-to-end object detection. In ICLR, 2021.

A Exact Experimental Protocol

Initialization. We load the official Grounding DINO-Tiny checkpoint and the Grounding DINO-S/Swin-L cross-backbone configuration with the Hugging Face transformers implementation. Within each backbone, all variants begin from the same checkpoint. New residual gates are initialized near zero to avoid disrupting pretrained predictions.

Data and evaluation. The controlled development protocol samples 10,000 COCO train2017 images and evaluates on a legacy set of 987 non-empty validation images. It is used only for architecture selection. Final experiments train on full train2017 and use the unfiltered official 5,000-image val2017; COCO metrics are computed by pycocotools with identical score thresholds and maximum detections. The LVIS experiment uses the COCO-trained checkpoints directly on LVIS val with no LVIS training or adaptation and reports overall, rare, common, and frequent AP.

Optimization. The current continuation recipe trains the decoder and core adapters, uses BF16 on one RTX 5090 D, and saves update-level checkpoints for automatic resumption. The training loss is computed from final decoder outputs after Hungarian matching. It combines focal-style classification, box ℓ_1, and generalized IoU losses. Auxiliary decoder losses, contrastive denoising, and the full original Grounding DINO augmentation recipe are not silently assumed.

B Additional Ablations and Negative Results

The current three-seed study shows that four spread interactions reach 47.10 AP, below the one-interaction model at 47.74. In a separate preliminary 10k/987 protocol, repeated-teacher distillation changed AP from 49.38 to 49.37. Because the protocols differ, the preliminary values are not compared numerically with the main tables; they are reported only to document that teacher supervision did not rescue repeated fusion. We therefore exclude distillation from the final method.

C Structural Interaction Accounting

For completeness, we record an auxiliary attention-score count for the text-facing cross-modal paths,

$$\mathcal{B}_{\mathrm{xm}} = \sum_{\ell \in \mathcal{I}} Q_\ell K_t + K_t T, \tag{7}$$

where the first term counts decoder-query–to–text-core scores and the second counts text-resampler retrieval scores; constants for heads and channels are omitted. Under the configurations used in the paper, the recorded totals are 5.913M for dense fusion, 0.682M for Compact, 0.987M for High-Config, and 2.744M for High-Config with four interactions. This accounting is included only to make the structural intervention explicit. It is not a FLOP, latency, memory, parameter-count, or end-to-end speed measurement and is not used as a primary contribution.

At batch size 8 on an RTX 5090 D, Compact shows no reliable wall-clock gain over dense fusion; text cross-attention is not the runtime bottleneck in this implementation.