RETHINKING OBJECT DETECTION: WHAT IF VISUAL TARGETS DIVERGE FROM UNIVERSAL TAXONOMIES?

Anonymous authors
Paper under double-blind review

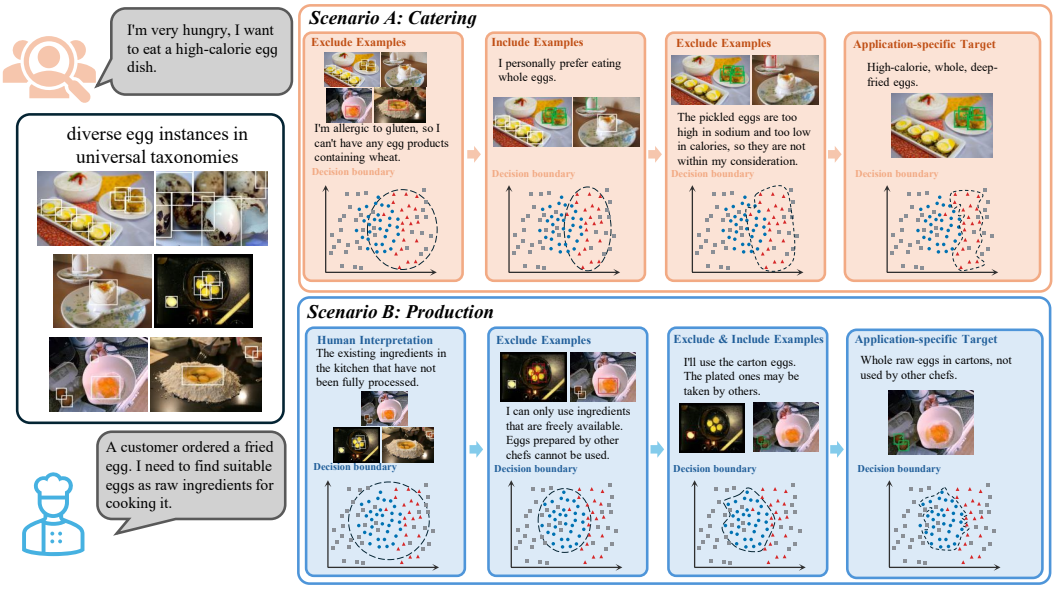


Figure 1: **An illustrative example of application-specific targets.** Red indicates the target in scenario A, while blue indicates the target in scenario B. Starting from the same semantic category, different application requirements and human interpretation progressively align the application intent with the visual target, resulting in different operational visual decision boundaries.

ABSTRACT

Existing object detection methods typically assume that detection targets are specified in advance by semantic categories or language descriptions. Real-world applications, however, may require targets whose visual boundaries depend on application requirements and cannot be fully specified initially. We introduce **Progressive Visual Target Alignment**, a new detection task in which targets preserve their semantic identity while being defined according to application scenarios and progressively calibrated through interaction. We construct **AST-50K**, the first large-scale benchmark for this task, comprising 52,577 images from 43 application scenarios and 117 application-specific targets. We further establish a **training-free agentic baseline** that iteratively calibrates target and orchestrates complementary frozen visual tools through multimodal interaction. Experiments demonstrate that existing detection paradigms, including OVD, vision-language, and image-generation-based methods, suffer from inherent limitations on AST-50K, whereas our framework delivers substantial improvements.

1 INTRODUCTION

Recent advances in object detection and open-vocabulary detection (OVD) have dramatically expanded the range of visual concepts that machines can recognize by learning from massive image-text corpora Zhu et al. (2020); Li et al. (2022); Liu et al. (2024c). Despite their impressive generalization capability, these paradigms typically formulate detection around an explicitly specified

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.

semantic vocabulary: conventional detectors operate on predefined categories, while OVD and vision-language models extend this vocabulary through novel semantic concepts or language descriptions Liu et al. (2024c); Cheng et al. (2024). Such formulations have driven remarkable success on existing benchmarks, where recognition objectives are defined in advance and evaluation focuses on semantic category coverage Lin et al. (2014); Gupta et al. (2019); Cheng et al. (2024).

Real-world applications, however, reveal a different challenge in a class of *niche visual scenarios*, where the desired targets are defined by application requirements rather than by conventional semantic taxonomies. In these scenarios, the difficulty is no longer only recognizing semantic categories, but determining *which visual instances should be recognized for a particular purpose*. Users are often interested in a small set of targets whose intended visual boundary is governed by application requirements rather than semantic taxonomies. Such requirements may be incomplete at the outset: the user's intent may identify what is needed at a high level, while leaving ambiguous which visual instances should be included or excluded. Consequently, the desired target may not be faithfully expressed by an existing category name or a simple combination of a category and visual attributes.

This phenomenon is consistent with established accounts of human concepts and decision making. Research on family resemblance, prototype-, and exemplar-based categorization suggests that concepts need not be fully captured by fixed verbal rules, while concrete instances can inform category judgments Ahmed (2011); Rosch & Lloyd (1978); Medin & Schaffer (1978). Complementarily, Simon's theory of bounded rationality recognizes that real-world objectives may be incompletely specified and refined as information becomes available Simon (1955; 1978). Together, these perspectives suggest that an initial application requirement may provide only a partial specification of an intended visual target, motivating progressive alignment between the application intent and its operational visual interpretation as additional information becomes available.

Figure 1 illustrates this phenomenon through two niche visual scenarios involving the same semantic category, *egg*. Starting from the same semantic category, *egg*, different application scenarios lead to different interpretations of which visual instances should be selected. In catering, the target is shaped by dietary preferences and preparation characteristics, while in production, it is shaped by ingredient availability and preparation status. Such distinctions depend on both application context and human interpretation. As concrete examples, contextual constraints, and human feedback become available, the intended visual target is progressively aligned with the application requirement, leading to an increasingly precise operational visual boundary.

These observations motivate a different formulation of visual detection. We introduce **Progressive Visual Target Alignment (PVTA)**, a task in which an initially under-specified application intent is progressively aligned with its intended visual target through additional examples, contextual information, and human feedback. The goal is to establish an operational visual boundary that determines which instances should be recognized for the intended application. Within a PVTA scenario, the application intent specifies the purpose or requirement underlying the desired target, while the visual target denotes the set of instances that should ultimately be recognized. The two need not be fully aligned from the outset: the application intent may constrain the desired target without uniquely determining its visual boundary.

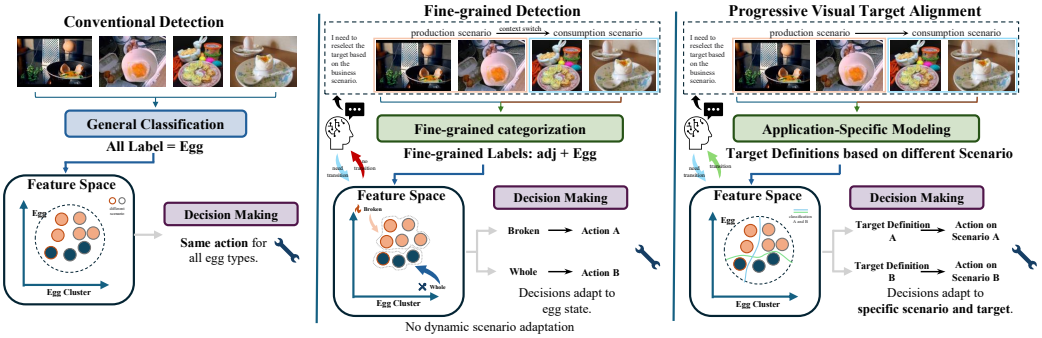


Figure 2: **Comparison of three detection paradigms.** Progressive Visual Target Alignment defines operational targets across application scenarios while preserving semantic identity.

Under this formulation, PVTA differs from existing detection paradigms in how the detection target is established. Fine-grained detection distinguishes visually similar instances as different semantic

subclasses Zhao et al. (2017); Grcić et al. (2024); Liu et al. (2024a), whereas PVTA preserves the underlying semantic identity while allowing the operational visual boundary to vary according to the intended application. OVD similarly allows targets to be specified beyond a fixed vocabulary through semantic concepts or language descriptions Li et al. (2022); Liu et al. (2024c); Cheng et al. (2024), but generally assumes that the desired target is specified at inference time. In PVTA, the initial application intent may be insufficient to uniquely determine the visual target, requiring additional visual evidence or human feedback to progressively align the two. Figure 2 summarizes these distinctions; detailed discussion is provided in Appendix A.

To facilitate systematic research on this problem, we construct **AST-50K** (Application-Specific Target), the first large-scale benchmark dedicated to **Progressive Visual Target Alignment**. AST-50K contains 52,577 images from 43 real-world scenarios and covers 117 application-specific targets. Unlike conventional detection benchmarks Kuznetsova et al. (2020); Gupta et al. (2019); Lin et al. (2014), which annotate objects primarily by semantic categories, AST-50K additionally provides application-specific target annotations, allowing the same semantic category to correspond to different operational targets across scenarios.

Our evaluation reveals that existing visual recognition paradigms struggle with this setting. Conventional detectors, OVD models, vision-language models, and recent vision generalists based on image generation paradigms Liu et al. (2026); Han et al. (2026) all exhibit substantial limitations on AST-50K. These models are primarily designed for targets whose semantics or language descriptions are specified in advance, rather than for determining application-specific target. Parameter-efficient fine-tuning and prompt engineering can adapt models to new objectives Hu et al. (2021); Zhou et al. (2022b;a), but generally treat adaptation as a configuration step, making repeated adaptation to evolving application requirements costly and difficult to maintain.

As an initial solution to this new task, we establish a **training-free agentic baseline** that treats Progressive Visual Target Alignment as an iterative target calibration and workflow orchestration problem. Instead of updating model parameters, our framework composes frozen visual tools according to the current application intent and progressively calibrates the visual target through multimodal interaction. A Target Calibration Critic (TCC) analyzes intermediate visual results, target descriptions, and model feedback to identify failures and update the operational visual boundary, while the planner orchestrates complementary visual tools to execute the calibrated target. The resulting workflow is consolidated into a reusable inference procedure without additional training.

The main contributions of this work are summarized as follows:

- We identify and formulate **Progressive Visual Target Alignment (PVTA)**, a new visual recognition task in which an initially under-specified application intent is progressively aligned with an operational visual target through additional information and interaction.
- We construct **AST-50K**, the first large-scale benchmark for Progressive Visual Target Alignment, comprising 52,577 images from 43 scenarios with 117 targets.
- We establish a **training-free agentic baseline** that progressively calibrates visual targets and orchestrates complementary visual tools through iterative interaction. Extensive experiments demonstrate its effectiveness across diverse application scenarios.

2 Related Work

Our work intersects with object detection, fine-grained recognition, and visual agents, whereas PVTA addresses the prior problem of progressively establishing application-specific visual targets. A detailed discussion is provided in Appendix A.

3 AST-50K Benchmark

3.1 Task Definition

Let $\mathcal{C}$ denote the set of semantic categories and $\mathcal{S}$ the space of *niche visual scenarios*. For a semantic category $c \in \mathcal{C}$, let $\mathcal{T}_c$ denote the conceptual space of possible application-specific visual targets associated with c, which is inherently open-ended due to varying application requirements. For a

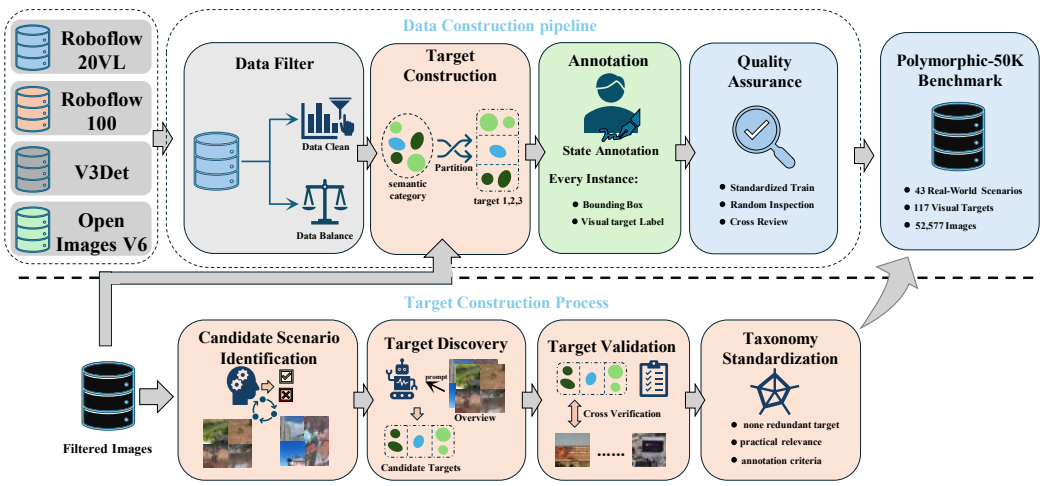


Figure 3: **Overall construction pipeline of AST-50K.** The upper part illustrates the benchmark construction process from data collection to quality control, while the lower part details the taxonomy construction procedure used to define Target states.

scenario $s_i \in \mathcal{S}$, the intended visual target may not be fully specified by the initial *application intent*. PVTA progressively aligns the application intent with an operational visual target through visual evidence and human feedback, refining ambiguous inclusion and exclusion criteria as necessary.

For benchmark construction and evaluation, we instantiate a finite set of representative targets for each selected scenario:

$$\mathcal{T}_{c,s_i} = \left\{ t_1^{(c,s_i)}, t_2^{(c,s_i)}, \ldots, t_{K_{c,s_i}}^{(c,s_i)} \right\} \subseteq \mathcal{T}_c, \tag{1}$$

where $\mathcal{T}_{c,s_i}$ denotes the set of instantiated application-specific visual targets for category c under scenario s_i, and K_{c,s_i} is the number of targets included in the benchmark. Each $t_k^{(c,s_i)}$ represents a scenario-specific visual target obtained through this alignment process. These targets are not semantically distinct subclasses; instead, they preserve the underlying semantic identity of their associated category while allowing their operational visual boundaries to vary across scenarios. Formally, the semantic identity of each target is preserved as

$$\mathrm{Sem}\left(t_k^{(c,s_i)}\right) = c, \qquad \forall\, t_k^{(c,s_i)} \in \mathcal{T}_{c,s_i}, \tag{2}$$

where $\mathrm{Sem}(\cdot)$ maps an application-specific visual target to its underlying semantic category. Once the operational visual target has been established, the corresponding detection objective is to localize its instances in an image I captured under scenario s_i. A detector produces

$$\mathcal{D}(I, s_i) = \{(b_j, c_j, t_j)\}_{j=1}^{N}, \tag{3}$$

where N is the number of detected instances, b_j denotes the predicted bounding box, $c_j \in \mathcal{C}$ is the underlying semantic category, and $t_j \in \mathcal{T}_{c_j,s_i}$ is the corresponding application-specific visual target.

3.2 Benchmark Overview

Table 1 summarizes the benchmark composition. AST-50K integrates four public datasets, covering 43 application scenarios, 117 application-specific targets, and 52,577 images. Each object instance is annotated with both its semantic category and its application-specific target under the corresponding niche visual scenario, allowing the same semantic category to map to different targets across scenarios. Figure 3 summarizes the construction pipeline, while Figure 6 presents representative application-specific targets.

3.3 Benchmark Construction Pipeline

Figure 3 illustrates the overall benchmark construction pipeline. Starting from four public datasets, we progressively construct the benchmark through image filtering, target taxonomy construction, human annotation, and quality assurance. Application-specific targets are established through a human-in-the-loop process, followed by instance-level annotation and multi-round verification. Detailed procedures are provided in Appendix B.

Table 1: **Source composition of AST-50K.** Each source dataset contributes complementary application scenarios and targets, jointly providing broad scenario coverage and visual diversity.

Source	Scenarios	Targets	Images
V3Det	2 (4.7%)	7 (6.0%)	214 (0.4%)
Open Images V6	13 (30.2%)	38 (32.5%)	24,571 (46.7%)
Roboflow-20	10 (23.3%)	28 (23.9%)	12,688 (24.1%)
Roboflow-100	18 (41.9%)	44 (37.6%)	15,104 (28.7%)
Total	**43**	**117**	**52,577**

3.4 Benchmark Statistics

Target Complexity. Table 2 summarizes the number of application-specific targets instantiated within each scenario. Most scenarios contain two or three targets, while only four scenarios require four targets. This limited cardinality does not imply that the underlying target space is intrinsically small, rather, we select a compact set of representative targets for benchmark construction and evaluation. The number of instantiated targets is determined by the diversity of application-oriented requirements within each scenario.

Table 2: **Targets per scenario.**

Targets	Scenarios	Ratio
2	16	37.2%
3	23	53.5%
4	4	9.3%

Target Image Distribution. Figure 4 presents the number of images associated with each application-specific target, ranked by image frequency. Target definitions are determined by application requirements and contextual distinctions within each scenario, whereas image frequency depends on the availability of corresponding visual instances in the source data. The resulting distribution is naturally long-tailed, with a small number of frequently observed targets and many relatively rare but practically important ones. We retain this imbalance to reflect realistic application conditions, where application-specific targets vary substantially in visual prevalence. Together with the target complexity across scenarios, this distribution makes AST-50K a realistic and challenging benchmark for evaluating PVTA.

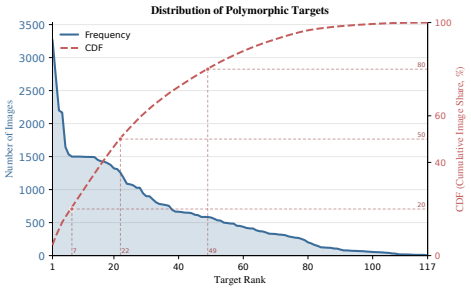


Figure 4: **Target frequency distribution.**

4 Methodology

As illustrated in Figure 5, we formulate niche detection as *workflow orchestration learning*: given a target description and demonstration images, a Planner and a Target Calibration Critic (TCC) iterate to discover a detection pipeline from candidate modules. Implementation details are provided in Appendix D.

4.1 Overall Orchestration Loop

Given annotated demonstration images $\mathcal{S} = \{(I_i, a_i)\}_{i=1}^{L}$, where a_i denotes the bounding-box annotations of I_i and L is the number of demonstration images, an initial target description $d = f_{\text{rtd}}^{(0)}$, and a set of target images $\mathcal{I} = \{I_1, \ldots, I_N\}$ with N images, the goal is to discover a workflow that maps images to accurate prediction results. The Planner is an MLLM orchestrating a set of candidate modules $\mathbb{T} = \{\text{M}_{\text{Seed2Pro}}, \text{M}_{\text{SAM3}}, \text{M}_{\text{RexOmni}}, \text{Fuser}\}$, where Seed2.0-Pro Seed Team (2026),

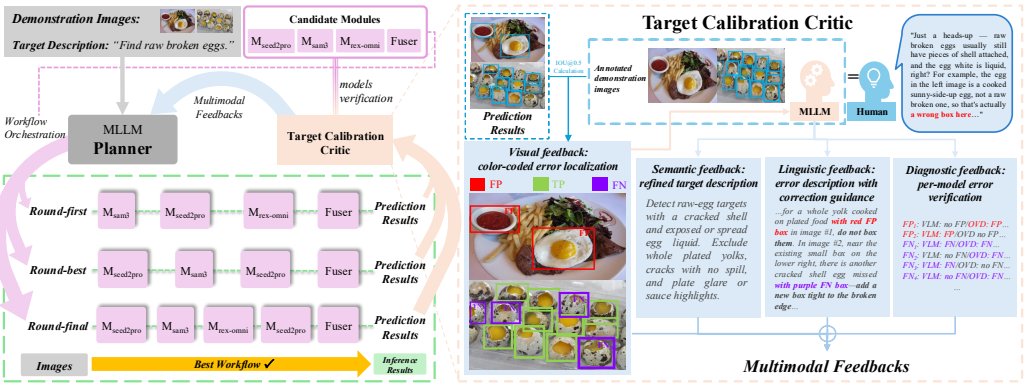


Figure 5: **Framework overview.** The MLLM-based Planner orchestrates candidate modules across rounds, while the Target Calibration Critic (TCC) generates corrective feedback from predictions and annotated reference examples. The best workflow is selected after the final round.

SAM3 Carion et al. (2026), and Rex-Omni Jiang et al. (2026) are off-the-shelf visual perception models and the Fuser is a deterministic post-processing module (its implementation is detailed in Appendix D.5). At round k, it outputs a workflow $\mathcal{W}^{(k)} = (\Pi^{(k)}, H^{(k)})$, where $\Pi^{(k)}$ is an ordered module sequence and $H^{(k)}$ the per-module configuration (hyperparameters and prompts). Executing on the demonstration images $\{I_i\}_{i=1}^{L}$ yields prediction results:

$$B^{(k)} = \Psi_{\text{exec}}\big(\{I_i\}_{i=1}^{L}, \mathcal{W}^{(k)}\big) \tag{4}$$

where Ψ_{exec} deterministically executes the module chain. The results $B^{(k)}$ and $\mathcal{S}$ are passed to the TCC, returning feedback $f^{(k)}$. Formally:

$$\mathcal{W}^{(k)} = \begin{cases} \Phi\big(d, \{I_i\}_{i=1}^{L}; \mathcal{P}_{\text{plan}}\big), & k = 1 \\ \Phi\big(f_{\text{rtd}}^{(k-1)}, \{I_i\}_{i=1}^{L}, f^{(k-1)}; \mathcal{P}_{\text{plan}}\big), & k > 1 \end{cases} \tag{5}$$

where Φ is the underlying MLLM and $\mathcal{P}_{\text{plan}}$ its prompt; round 1 uses the initial description and demonstrations, while later rounds add the previous refined description and feedback.

After K rounds, the best workflow is selected by maximizing a detection metric $\mathcal{E}$ (e.g., mAP) between predictions and the annotated demonstrations:

$$\mathcal{W}^{*} = \arg\max_{\{\mathcal{W}^{(k)}\}_{k=1}^{K}} \mathcal{E}\big(\Psi_{\text{exec}}(\{I_i\}_{i=1}^{L}, \mathcal{W}^{(k)}), \mathcal{S}\big) \tag{6}$$

The selected workflow $\mathcal{W}^{*}$ is then applied to the full target set $\mathcal{I}$ to produce the final predictions.

4.2 Target Calibration Critic

The TCC uses an MLLM to produce corrective feedback calibrating the Planner. Given $B^{(k)}$, $\mathcal{S}$, and the current description $f_{\text{rtd}}^{(k-1)}$, it generates four feedback signals $f^{(k)} = \{f_{\text{vis}}^{(k)}, f_{\text{rtd}}^{(k)}, f_{\text{fcd}}^{(k)}, f_{\text{ver}}^{(k)}\}$. All operations share the same model Φ with task-specific prompts $\mathcal{P}_{\text{rtd}}$ and $\mathcal{P}_{\text{fcd}}$, in four stages:

Visual feedback: localizes errors with color-coded boxes Predicted boxes are matched by IoU $\geq \tau$: true positives (green), unmatched predictions (red, FP), and unmatched references (purple, FN). The visualization $f_{\text{vis}}^{(k)}$ overlays these boxes; let $\mathcal{F}_{\text{fail}}^{(k)}$ denote the error set.

Semantic feedback: refines the target description Grounded in application context and human interpretation, it conditions on the visualization:

$$f_{\text{rtd}}^{(k)} = \Phi\big(f_{\text{rtd}}^{(k-1)}, f_{\text{vis}}^{(k)}; \mathcal{P}_{\text{rtd}}\big) \tag{7}$$

The refined description incorporates exclusion criteria (e.g., "exclude whole plated yolks and plate glare") and positive constraints (e.g., "cracked shell with exposed or spread egg liquid"), sharpening target semantics.

Linguistic feedback: describes errors with correction guidance Natural-language descriptions with correction guidance:

$$f_{\text{fcd}}^{(k)} = \Phi\big(\mathcal{F}_{\text{fail}}^{(k)}, f_{\text{vis}}^{(k)}, f_{\text{rtd}}^{(k)}; \mathcal{P}_{\text{fcd}}\big) \tag{8}$$

Each description identifies the error (e.g., a cooked sunny-side-up egg misdetected as a raw broken egg) and specifies actions (e.g., "do not box them").

Diagnostic feedback: verifies errors per model For each error $e \in \mathcal{F}_{\text{fail}}^{(k)}$, extract its box b_e, image I_e, and refined description $f_{\text{rtd}}^{(k)}$, then run each detection model $m \in \mathcal{M} = \{\text{M}_{\text{Seed2Pro}}, \text{M}_{\text{SAM3}}, \text{M}_{\text{RexOmni}}\} \subset \mathbb{T}$ and compare against $\mathcal{S}$:

$$f_{\text{ver}}^{(k)} = \Big\{\big(e, \{\delta_m(e)\}_{m\in\mathcal{M}}\big)\Big\}_{e\in\mathcal{F}_{\text{fail}}^{(k)}}, \quad \delta_m(e) = \Psi_{\text{ver}}\big(m, b_e, I_e, f_{\text{rtd}}^{(k)}\big) \in \{\text{correct}, \text{FP}, \text{FN}\} \tag{9}$$

where Ψ_{ver} runs model m on the image, conditioned on the refined description, returning correct/FP/FN.

5 Experiments

We evaluate representative visual recognition paradigms on AST-50K, examine how additional visual evidence and feedback progressively align application intent with visual targets, analyze visual tool complementarity, and ablate TCC feedback. Appendix C provides experimental details.

5.1 Main Results

Table 3: **Main comparison on AST-50K.** We compare representative models from different visual recognition paradigms. Results are reported under two IoU thresholds. †*Evaluated on a 20-images-per-scene subset due to the high cost of Nano-Banana-Pro Google DeepMind (2025) and GPT-6.0-Astra OpenAI (2026b). Detailed subset construction and inference settings are provided in Appendix D.6.* **Bold** and underline indicate the best and second-best results, respectively. The same convention applies to all subsequent tables.

	IoU = 0.15				IoU = 0.50			
Method	mAP	mF1	Prec.	Rec.	mAP	mF1	Prec.	Rec.
Open-Vocabulary Detection (OVD)								
YOLOE-26 Chakrabarty (2026)	0.1282	0.1902	0.3325	0.2287	0.1064	0.1639	0.2817	0.1960
SAM3	<u>0.3009</u>	0.2829	0.2842	0.5350	<u>0.2764</u>	0.2623	0.2648	0.4903
Rex-Omni	0.2358	0.3668	<u>0.3942</u>	0.4614	0.1864	0.3150	<u>0.3375</u>	0.3929
Vision-Language Models (VLM)								
Seed2.0-Pro	0.2678	<u>0.3967</u>	0.3674	0.6276	0.2107	<u>0.3370</u>	0.3121	<u>0.5383</u>
Qwen3.6-Flash Bai et al. (2025)	0.1634	0.2916	0.2745	0.4542	0.0878	0.1848	0.1718	0.3028
GPT-5.2 OpenAI (2025)	0.1455	0.2669	0.2679	0.3762	0.0396	0.1019	0.0956	0.1569
GPT-5.6-Sol OpenAI (2026a)	0.2370	0.3538	0.2971	<u>0.6712</u>	0.1530	0.2642	0.2207	0.5136
Ours	**0.4983**	**0.6212**	**0.5879**	**0.7587**	**0.4084**	**0.5493**	**0.5208**	**0.6691**
Controlled comparison on the 20-images-per-scene subset								
SAM3†	0.3443	0.2991	0.2715	0.5467	0.3174	0.2793	0.2551	0.5094
Seed2.0-Pro†	0.3494	0.4194	0.3868	0.6855	0.2786	0.3576	0.3282	<u>0.5999</u>
GPT-6.0-Astra†	<u>0.4636</u>	<u>0.5117</u>	<u>0.4759</u>	<u>0.7037</u>	<u>0.3793</u>	<u>0.4455</u>	<u>0.4176</u>	0.5991
Generative Models (GM)								
Nano-Banana-Pro†	0.3118	0.3991	0.4511	0.4478	0.1899	0.2721	0.3114	0.2931
Ours†	**0.5841**	**0.6324**	**0.6126**	**0.7794**	**0.5043**	**0.5692**	**0.5504**	**0.7051**

We first evaluate whether existing visual recognition paradigms can directly address PVTA. As shown in Table 3, existing paradigms exhibit substantial limitations on AST-50K. None of the individual paradigms consistently achieves strong performance across the diverse application scenarios, and their overall performance remains considerably below that of Ours. This suggests that PVTA is not adequately addressed by directly applying existing visual recognition paradigms, despite their

broad semantic and visual capabilities. These models are primarily designed to recognize targets specified before inference, whereas AST-50K involves application-specific visual targets whose operational boundaries may not be fully determined initially.

The performance gap is also consistent with the need for additional interaction in this setting. Rather than determining the visual target from an initially under-specified application intent, OURS iteratively examines visual results and feedback to progressively align the application intent with the visual target and its operational boundary.

5.2 HOW DOES VISUAL TARGET ALIGNMENT PROGRESS?

Table 4: **Progressive visual target alignment.** The comparison moves from coarse target names, to visual examples with detailed captions, to iterative calibration with TCC and increasingly capable visual tools, with an oracle condition providing complete target information. Results are reported on the representative evaluation subset of AST-50K (Appendix C.2).

	IoU = 0.15				IoU = 0.50			
Method	mAP	mF1	Prec.	Rec.	mAP	mF1	Prec.	Rec.
Seed2.0-Pro	0.3057	0.4500	0.4787	0.6381	0.1926	0.3346	0.3763	0.4812
Examples + Captions + Seed2.0-Pro	0.4444	0.5955	<u>0.6787</u>	0.6224	0.3029	0.4575	0.5440	0.4629
Planner(TCC, Seed2.0-Pro)	0.5372	0.6687	0.6696	<u>0.7609</u>	0.4086	0.5504	0.5693	0.6031
Planner(TCC, Modules)	<u>0.5488</u>	<u>0.6784</u>	0.6633	**0.7702**	<u>0.4255</u>	<u>0.5694</u>	<u>0.5714</u>	**0.6276**
Oracle + Seed2.0-Pro	**0.5648**	**0.6969**	**0.7222**	0.7516	**0.4316**	**0.5855**	**0.6242**	<u>0.6145</u>

To examine how additional visual evidence progressively aligns the application intent with the visual target, we first evaluate Seed2.0-Pro with increasingly informative target evidence (see Appendix D.2 for detailed settings). Providing concrete visual examples, together with the application-specific description, consistently improves detection performance over relying on the initial semantic specification alone. This suggests that visual targets cannot always be sufficiently characterized by their semantic category or textual description, and that visual instances provide important information for determining which instances should be included in the target.

Planner(TCC, Seed2.0-Pro) further improves over example- and caption-based inference, indicating that representative examples alone may still be insufficient to fully determine the desired visual target. By examining intermediate predictions and their failures, TCC progressively aligns the application intent with the visual target based on visual feedback. This supports our formulation of target calibration as an iterative process rather than a one-time target determination step.

We further observe a performance gain when extending Planner(TCC, Seed2.0-Pro) to Planner(TCC, Modules). Since the underlying calibration mechanism is retained, this improvement highlights the complementary capabilities of heterogeneous visual tools: different tools can provide useful evidence for different application scenarios, and workflow-level orchestration allows the system to exploit these differences.

Finally, the oracle condition provides complete target definitions and visual specifications and therefore serves as an approximate upper bound. PLANNER(TCC, MODULES) approaches this upper bound, suggesting that iterative calibration can recover a substantial portion of the target-defining information available, while the remaining gap suggests that the intended target boundary remains challenging to fully recover.

5.3 DO VISUAL TOOLS COMPLEMENT EACH OTHER?

To specify which visual capabilities are required to localization, we examine whether individual visual tools provide consistent performance across application scenarios. Among the evaluated models, Seed2.0-Pro, SAM3, and Rex-Omni exhibit relatively close overall performance, while showing substantially different strengths across scenarios. As shown in Table 5, SAM3 performs best on Egg, Rex-Omni on Coins, while Seed2.0-Pro is strongest on Actions and Smoke. This scenario-dependent ranking suggests that these tools provide complementary visual capabilities rather than simply different levels of overall accuracy.

Table 5: **mAP of individual visual tools on AST-50K (**IoU**=0.50).** All tools are evaluated independently. Full metrics across both IoU thresholds are provided in Table 14.

Tool	Overall	Actions	Egg	Coins	Smoke
SAM3	**0.2764**	0.0000	**0.3248**	0.3428	0.0013
Rex-Omni	0.1864	0.2305	0.1908	**0.5629**	0.1999
Seed2.0-Pro	0.2107	**0.2316**	0.2038	0.4408	**0.3162**
Qwen3.6-Flash	0.0878	0.0478	0.0517	0.3485	0.1775
GPT-5.2	0.0396	0.0005	0.0162	0.0826	0.1458
GPT-5.6-Sol	0.1530	0.1586	0.1340	0.3883	0.2835

We therefore retain Seed2.0-Pro, SAM3, and Rex-Omni as the core modules for workflow orchestration. Seed2.0-Pro is selected as the primary VLM due to its strongest overall performance among the evaluated VLMs and its flexible natural-language understanding for application-specific target descriptions. Detailed per-scenario results and metrics across both IoU thresholds are provided in Appendix E.4.

5.4 Ablation of Target Calibration Feedback

Table 6: **Ablation of TCC feedback.** Each row removes one feedback type while keeping the remaining components unchanged. Results are reported on the representative evaluation subset.

Configuration	$\text{mAP}_{0.50}$	$\text{mF1}_{0.50}$	$\text{Prec.}_{0.50}$	$\text{Rec.}_{0.50}$
Full TCC	**0.4255**	**0.5694**	**0.5714**	0.6276
w/o Failure Visualization	0.4091	0.5588	0.5614	0.6241
w/o Refined Target Description	0.4105	0.5598	0.5492	**0.6313**
w/o Failure Description	0.4056	0.5547	0.5686	0.6091
w/o Model-level Verification	0.3254	0.4712	0.5608	0.4954

As shown in Table 6, Full TCC achieves the highest mAP and mF1, and removing any individual feedback source leads to a degradation in both metrics. This indicates that each feedback source contributes useful information, while the relatively small changes in individual precision or recall suggest that they affect different aspects of the calibration process.

Notably, removing model-level verification causes a substantially larger performance drop than removing the other feedback sources. This is consistent with the tool analysis in Section 5.3, which reveals substantial capability differences across tools and scenarios. Without verification, mismatches between tool allocation and actual capabilities may remain undetected, leading to unreliable feedback and degraded calibration. These results highlight the complementary roles of the TCC feedback sources and the importance of model-level verification for effective calibration.

6 Conclusion

In this paper, we introduced *Progressive Visual Target Alignment (PVTA)*, a new visual detection task in which an initially under-specified application intent is progressively aligned with an operational visual target. Our study shows that the challenge extends beyond recognizing visual objects: it lies in determining which visual instances should constitute the target under a given application context, where the initial requirement may provide only a partial specification of the desired visual boundary. To study this problem, we constructed AST-50K and established a training-free agentic baseline that progressively calibrates visual targets and orchestrates complementary visual tools through multimodal interaction. Our experiments demonstrate the limitations of existing recognition paradigms in this setting and show that iterative target calibration can substantially improve detection performance. More broadly, PVTA highlights a previously underexplored layer of visual recognition: before detecting an object, a system may first need to align *what should be recognized* with *why it should be recognized*.

AI USE STATEMENT

In this work, we used generative AI tools for improving the writing and figures, assisting in identifying candidate scenarios for the benchmark, and reviewing the code. We have not used generative AI tools for other tasks with required disclosure, or the rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by our group authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Implementation details are provided in Appendix D. The full AST-50K benchmark and dataset, as well as the complete source code of our method, will be publicly released upon acceptance. All experimental results reported in this paper are supported by actual experimental data. The raw experiment result files amount to approximately 2GB, which exceeds the 100MB supplementary material limit and therefore cannot be included in full; these files can be made available through alternative channels upon request.

REFERENCES

Arif Ahmed. Wittgenstein: Philosophical investigations. *Philosophy of Language: The Key Thinkers*, pp. 121, 2011.

Anthropic. Introducing claude opus 5. `https://www.anthropic.com/news/claude-opus-5`, July 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L Alexander, David W Jacobs, and Peter N Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2011–2018, 2014.

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.

Sudip Chakrabarty. Yolo26: An analysis of nms-free end to end framework for real-time object detection. *arXiv preprint arXiv:2601.12882*, 2026.

Yuming Chen, Jiangyan Feng, Haodong Zhang, Lijun Gong, Feng Zhu, Rui Zhao, Qibin Hou, Ming-Ming Cheng, and Yibing Song. Re-aligning language to visual objects with an agentic workflow. In *International Conference on Learning Representations*, volume 2025, pp. 4628–4660, 2025.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16901–16911. IEEE, 2024.

Floriana Ciaglia, Francesco Saverio Zuppichini, Paul Guerrie, Mark McQuade, and Jacob Solawetz. Roboflow 100: A rich, multi-domain object detection benchmark. *arXiv preprint arXiv:2211.13523*, 2022.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Hui Feng, Shanshan Wang, and Shuzhi Sam Ge. Fine-grained visual recognition with salient feature detection. *arXiv preprint arXiv:1808.03935*, 2018.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Google DeepMind. Nano banana pro (gemini 3 pro image), 2025. URL `https://deepmind.google/models/gemini-image/pro/`. AI image generation and editing model built on Gemini 3 Pro.

Matej Grcić, Artyom Gadetsky, and Maria Brbić. Fine-grained classes and how to find them. *arXiv preprint arXiv:2406.11070*, 2024.

Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5351–5359. IEEE, 2019.

Xiaoyang Han, Jianhua Li, Kewang Deng, Zukai Chen, Xuanke Shi, Sihan Wang, Boxuan Li, Linyan Wang, Siyi Xie, Xin You, et al. Vision as unified multimodal generation. *arXiv preprint arXiv:2607.06560*, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia computer science*, 199:1066–1073, 2022.

Qing Jiang, Junan Huo, Xingyu Chen, Yuda Xiong, Zhaoyang Zeng, Yihao Chen, Tianhe Ren, Junzhi Yu, and Lei Zhang. Detect anything via next point prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10955–10965. IEEE, 2022.

Ming Li, Jike Zhong, Chenxin Li, Liuzhuozheng Li, Nie Lin, and Masashi Sugiyama. Vision-language model fine-tuning via simple parameter-efficient modification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14394–14410, 2024.

Di Lin, Xiaoyong Shen, Cewu Lu, and Jiaya Jia. Deep lac: Deep localization, alignment and classification for fine-grained recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1666–1674, 2015.

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

Mingxuan Liu, Subhankar Roy, Wenjing Li, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Democratizing fine-grained visual recognition with large language models. In *International conference on learning representations*, volume 2024, pp. 16942–16973, 2024a.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. Llava-plus: Learning to use tools for creating multimodal agents. In *European conference on computer vision*, pp. 126–142. Springer, 2024b.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024c.

Wei Liu, Jiaxin Lin, and Rui Chen. Open-source image editing models are zero-shot vision learners. *arXiv preprint arXiv:2605.04566*, 2026.

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Song XiXuan, Yifan Xu, Shudan Zhang, Hanyu Lai, Jiadai Sun, Xinyue Yang, et al. Visualagentbench: Towards large multimodal models as visual foundation agents. In *International Conference on Learning Representations*, volume 2025, pp. 95650–95707, 2025.

Douglas L Medin and Marguerite M Schaffer. Context theory of classification learning. *Psychological review*, 85(3):207, 1978.

OpenAI. Introducing gpt-5.2. `https://openai.com/index/introducing-gpt-5-2/`, 2025.

OpenAI. Gpt-5.6: Frontier intelligence that scales with your ambition. `https://openai.com/index/gpt-5-6/`, July 2026a.

OpenAI. Gpt-6 astra: A new generation of intelligence. `https://openai.com/index/gpt-6-astra/`, September 2026b.

Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, et al. Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*, 2024.

Eleanor Rosch and Barbara B Lloyd. Principles of categorization. 1978.

Samuel Schulter, Yumin Suh, Konstantinos M Dafnis, Zhixing Zhang, Shiyu Zhao, Dimitris Metaxas, et al. Omnilabel: A challenging benchmark for language-based object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11953–11962, 2023.

ByteDance Seed Team. Seed2.0 model card: Towards intelligence frontier for real-world complexity. *arXiv preprint arXiv:2607.00248*, 2026. URL `https://arxiv.org/abs/2607.00248`.

Herbert A Simon. A behavioral model of rational choice. *The quarterly journal of economics*, pp. 99–118, 1955.

Herbert A Simon. Rationality as process and as product of thought. *The American economic review*, 68(2):1–16, 1978.

Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine learning and knowledge extraction*, 5(4):1680–1716, 2023.

Chenyu Wang, Weixin Luo, Sixun Dong, Xiaohua Xuan, Zhengxin Li, Lin Ma, and Shenghua Gao. Mllm-tool: A multimodal large language model for tool agent learning. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6678–6687. IEEE, 2025.

Jiaqi Wang, Pan Zhang, Tao Chu, Yuhang Cao, Yujie Zhou, Tong Wu, Bin Wang, Conghui He, and Dahua Lin. V3det: Vast vocabulary visual detection dataset. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19787–19797. IEEE, 2023.

Chi Xie, Zhao Zhang, Yixuan Wu, Feng Zhu, Rui Zhao, and Shuang Liang. Described object detection: Liberating object detection with flexible expressions. *Advances in Neural Information Processing Systems*, 36:79095–79107, 2023.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Yi Xin, Jianjiang Yang, Siqi Luo, Yuntao Du, Qi Qin, Haoxing Chen, Kangrui Cen, Yangfan He, Bin Fu, Yuewen Cao, et al. Parameter-efficient fine-tuning for pre-trained vision models: A survey and benchmark. *International Journal of Computer Vision*, 134(6):304, 2026.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 16795–16804. IEEE, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International journal of computer vision*, 130(9):2337–2348, 2022b.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

A Related Work

A.1 Object Detection

Object detection has progressed through a broad range of architectures, from two-stage detectors Girshick (2015) and one-stage detectors Jiang et al. (2022); Terven et al. (2023) to transformer-based Zhu et al. (2020) and end-to-end detection frameworks Zhang et al. (2022). These approaches have been developed and evaluated primarily on large-scale benchmarks such as COCO Lin et al. (2014) and LVIS Gupta et al. (2019), where object categories and annotation taxonomies are defined in advance.

This formulation places the semantic taxonomy outside the detection process: given a predefined category vocabulary, the detector learns to localize instances belonging to each category. Extending the detector to a new category or a different target definition therefore typically requires new annotations, model adaptation, or a new detection specification. In contrast, **Progressive Visual Target Alignment (PVTA)** considers scenarios in which the visual target itself may need to be established according to application requirements before its instances can be reliably detected.

A.2 Fine-grained Detection

Fine-grained visual recognition focuses on distinguishing visually similar instances according to subordinate semantic categories Zhao et al. (2017). Benchmarks such as Birdsnap Berg et al. (2014), for example, distinguish visually similar birds at the species level. Existing approaches improve such discrimination by modeling subtle visual differences through localization, alignment, and fine-grained feature representations Lin et al. (2015); Feng et al. (2018). More recent work further explores the discovery and representation of fine-grained semantic categories, including approaches that leverage large language models Grcić et al. (2024); Liu et al. (2024a).

Although fine-grained recognition also operates on visually subtle distinctions, its underlying taxonomy remains defined by distinct semantic identities: different visual groups correspond to different semantic categories or subclasses. PVTA instead allows multiple operational targets to remain associated with the same underlying semantic category, with their distinctions determined by application requirements and context. Thus, fine-grained detection refines *what an object is*, whereas PVTA concerns *which instances of an existing semantic category should be recognized for a particular application*.

A.3 Vision-Language Object Detection

Vision-language pretraining has substantially expanded object detection beyond fixed category vocabularies. Open-vocabulary detectors and vision-language detection frameworks, including GLIP, Grounding DINO, and YOLO-World, leverage image–text alignment to detect concepts specified by category names, referring expressions, or natural-language descriptions Li et al. (2022); Liu et al. (2024c); Cheng et al. (2024). Subsequent work further improves open-vocabulary detection through larger-scale multimodal pretraining and more efficient vision-language architectures Ren et al. (2024). Prompt-based and parameter-efficient adaptation methods also enable pretrained vision-language models to specialize to downstream objectives Li et al. (2024); Xin et al. (2026).

These approaches substantially broaden how detection targets can be expressed, but the detection objective remains externally specified at inference time. The model is asked to recognize a semantic concept or language-level description whose intended visual referent is assumed to be given. PVTA considers a different setting in which the application requirement may not uniquely determine the desired visual target. The challenge therefore precedes conventional language-based grounding: the system must first establish which visual instances should constitute the target before performing target-specific detection.

A.4 Language-based Object Detection

Language-based object detection (LOD) further broadens target specification by allowing a given detection target to be described through flexible natural-language expressions. OmniLabel Schulter et al. (2023) introduces a challenging benchmark for language-based object detection with diverse free-form descriptions, including expressions referring to a single object, multiple objects, or no object. Described Object Detection (DOD) Xie et al. (2023) similarly moves beyond fixed category names by supporting flexible object descriptions and evaluating detection under both concise and more complex expressions Xie et al. (2023). These studies demonstrate that natural language can provide a substantially richer interface for specifying detection targets than fixed semantic labels.

More recently, Real-LOD Chen et al. (2025) introduces an agentic workflow that iteratively refines language descriptions to improve their alignment with visual objects. Through planning, tool use, and feedback, the system identifies and corrects mismatches between language descriptions and their visual referents. Despite its iterative alignment process, Real-LOD assumes that the visual object to be described is already given and focuses on improving the language representation of that object. PVTA addresses a different problem: the application requirement may itself leave the desired set of visual instances under-specified, requiring the target to be progressively established through visual evidence and human feedback. Thus, LOD primarily investigates *how a given target can be expressed or grounded in language*, whereas PVTA investigates *how the target to be recognized can be progressively established*.

A.5 LLM-based Visual Agent

Recent work increasingly integrates large language or multimodal models with external tools to construct visual agents capable of planning, reasoning, tool use, and multi-step task execution Liu et al. (2024b); Koh et al. (2024); Xie et al. (2024). Such systems enable multimodal models to select and coordinate perception tools, interact with visual environments, and combine intermediate observations to solve complex visual tasks Liu et al. (2024b); Wang et al. (2025); Liu et al. (2025).

Existing visual agents, however, generally assume that the underlying task objective is already specified and focus on determining *how* to accomplish it through planning, tool selection, and execution Koh et al. (2024); Xie et al. (2024). PVTA instead places the establishment of the visual target itself within the task. The agent must use visual evidence and feedback to progressively calibrate the target before executing the corresponding detection objective. Our work therefore combines target calibration with workflow orchestration, allowing an agent to acquire target-specific visual knowledge from representative examples and consolidate the resulting procedure into a reusable inference workflow without updating model parameters.

B BENCHMARK CONSTRUCTION DETAILS

B.1 DATA SOURCES

AST-50K is constructed from four large-scale public visual benchmarks: Roboflow 20VL, Roboflow 100 Ciaglia et al. (2022), V3Det Wang et al. (2023), and Open Images V6 Kuznetsova et al. (2020). These datasets provide complementary coverage in terms of application scenarios, semantic diversity, and visual appearance.

Roboflow 20VL and Roboflow 100 contribute numerous application-oriented application scenarios spanning infrastructure inspection, agriculture, environmental monitoring, manufacturing, and daily-life applications. Their scenario-centric organization facilitates the discovery of application-specific targets.

Open Images V6 and V3Det provide large-scale visual resources with broad semantic coverage and naturally long-tailed distributions. These datasets substantially enrich the diversity of semantic categories and visual appearances, enabling the identification of application-specific targets across diverse application scenarios.

B.2 APPLICATION-SPECIFIC TARGET CONSTRUCTION

To construct a benchmark that faithfully reflects PVTA requirements, we establish the target taxonomy through a multi-stage discovery, validation, and standardization procedure, as illustrated in the lower part of Figure 3. The procedure is designed to ensure that application-specific targets are not obtained by simply refining predefined semantic categories. Instead, candidate targets are identified from application scenarios and application requirements, and are subsequently validated and refined through human interpretation. The construction process consists of four stages: candidate scenario identification, application-specific target discovery, target validation and refinement, and target standardization.

Candidate Scenario Identification. We first identify candidate application scenarios from large-scale public visual datasets. Researchers inspect the semantic categories, category hierarchies, and existing task or scene descriptions provided by the source datasets, and select scenarios that potentially exhibit application-specific visual distinctions. The selection considers three criteria: (1) sufficient visual diversity within the semantic category, (2) potential practical relevance of visually distinguishable states, and (3) the possibility that the intended visual target cannot be fully determined by the existing semantic labels alone.

This initial screening produces 137 candidate scenarios. We then estimate the number of available images for each candidate scenario and apply dataset-scale constraints determined by the overall benchmark budget. Scenarios with fewer than 100 images are removed because they provide insufficient visual evidence for reliable target discovery and evaluation, while scenarios with more than 10,000 images are excluded to prevent a small number of highly populated scenarios from dominating the benchmark. After this filtering step, 63 candidate scenarios and 174 candidate targets remain for application-specific target discovery.

Application-specific Target Discovery. For each of the 63 candidate scenarios, we construct a representative visual context from the available images. For scenarios containing more than 500 images, 100 images are randomly sampled; for scenarios with fewer than 500 images, 50 images are sampled. Researchers then select 27 representative images from each scenario and organize them into three 3×3 image overviews. These overviews provide the scenario-level visual context for subsequent target discovery.

The resulting visual contexts are provided to a vision-language model (VLM) together with instructions describing the application context, application requirements, visual evidence, and commonsense reasoning. The VLM is asked to identify candidate application-specific visual targets rather than to introduce new semantic categories. For each scenario, the model generates multiple candidate targets, typically more than 10, together with a textual interpretation and preliminary decision criteria. These outputs serve only as candidate hypotheses for subsequent human review; they are not directly adopted as benchmark annotations.

Table 7: **Statistics of application-specific target construction.** The candidate space is progressively filtered, validated, and refined before obtaining the final benchmark taxonomy. The construction process combines visual evidence and human interpretation to establish operational target definitions.

Construction Stage	Scenarios	Targets	Scenario Retention	Target Retention
Initial candidate identification	137	–	–	–
After image-count filtering	63	–	$63/137 = 46.0\%$	–
Candidate target discovery	63	174	–	–
Final standardized taxonomy	43	117	$43/63 = 68.3\%$	$117/174 = 67.2\%$

Target Validation and Refinement. Candidate targets are subsequently reviewed by human researchers to determine whether they correspond to meaningful and visually supported application objectives. For each candidate interpretation, additional images are randomly sampled from the corresponding scenario and examined against the proposed decision criteria.

Candidates are removed when their supporting visual evidence is insufficient, their occurrence is too limited to support reliable evaluation, their decision boundaries are ambiguous, or their application rationale is considered impractical or overly specific. When a candidate target is visually meaningful but its interpretation does not accurately reflect human judgment, researchers revise the target definition and its decision criteria through iterative review. Thus, the final visual target emerges from progressively aligning the application requirements with human interpretation and visual evidence, while the resulting decision criteria are verified against real images.

Target Standardization. Finally, the candidate scenarios and their associated application-specific targets are jointly reviewed by multiple researchers to establish a consistent benchmark taxonomy. This cross-scenario review focuses on three criteria: (1) semantic consistency, ensuring that each target preserves the underlying semantic identity of its associated category; (2) practical relevance, ensuring that the target corresponds to a meaningful application-specific requirement; and (3) definition consistency, ensuring that targets within the same scenario have sufficiently clear and non-overlapping decision criteria. Redundant, ambiguous, or weakly supported targets are removed, while target definitions and their annotation criteria are unified when necessary. The same review is applied to scenario descriptions and application rationales to ensure that the final taxonomy forms a coherent set of application-oriented detection objectives.

Through this standardization process, 43 application scenarios and 117 application-specific targets are retained from the 63 candidate scenarios and 174 candidate targets obtained after target discovery. These scenarios and targets constitute the final taxonomy of AST-50K. Figure 6 provides representative examples of the resulting taxonomy, illustrating how different application requirements within specific application scenarios give rise to distinct application-specific targets while preserving their underlying semantic identities.

B.3 Annotation Protocol

After establishing the application-specific target taxonomy, we construct the final benchmark through a unified data preparation, human annotation, and quality-control pipeline. The procedure is designed to ensure that the resulting annotations are both consistent with the scenario-specific target definitions and reliable at the instance level.

Data Cleaning and Pre-balancing. Before application-specific target construction, we perform a preliminary cleaning and balancing process on the source datasets. Corrupted files, unreadable images, and samples with abnormal image resolutions are removed. We further mitigate extreme differences in category-level sample counts across source datasets. In the original data, the number of images associated with a source-category label may range from only a few dozen to several hundred thousand. Extremely sparse categories are removed, while excessively large categories are downsampled to a manageable five-digit scale.

This preprocessing is performed before scenario and target construction and is not intended to artificially balance the final benchmark. Instead, it provides a reasonably clean and tractable image

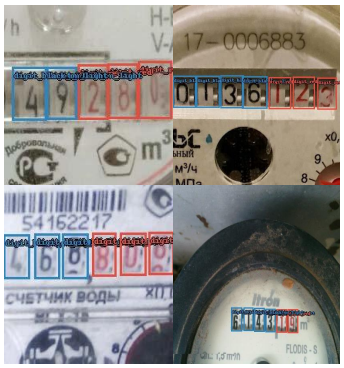
(a) On-the-spot Coconut Sales



opened consumed coconuts whole harvested coconuts growing_coconuts

Application & Rationale: Vendors need to distinguish opened coconuts from intact harvested coconuts because they require different handling, storage, and sales procedures.

(b) Water Meter Digit Maintenance



digit black on light digit white on black digit red

Application & Rationale: Supports water-meter maintenance and replacement planning by distinguishing display types associated with different aging conditions and replacement priorities. This enables automated meter statistics and targeted replacement scheduling, which cannot be achieved reliably with a generic water-meter label.

(c) Outdoor Billboard Compliance Review



digital led billboard static printed billboard

Application & Rationale: Billboard inspection distinguishes digital LED and static printed billboards because their display characteristics and compliance considerations differ.

(d) Volleyball Attack-Defense Analysis



attack play defend play

Application & Rationale: Tactical analysis requires distinguishing attacking and defensive plays, which represent different functional roles within volleyball actions.

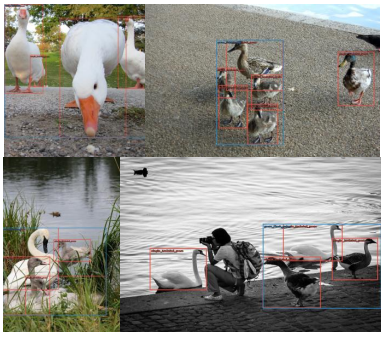
(e) Shelf Snack Packaging-type Differentiated Replenishment



small candy pouch large chip pillow bag boxed carton snack

Application & Rationale: Shelf replenishment requires packaging-aware recognition because different package formats correspond to different product groups and replenishment procedures.

(f) Goose Flock Monitoring for Breeding and Rearing



Single isolated goose goose flock group

Application & Rationale: Breeding management distinguishes isolated geese from flock groups to support individual-level and flock-level monitoring.

(g) Mask Wearing for Epidemic Control



Mask proper Mask improper No visible mask

Application & Rationale: Epidemic-control monitoring distinguishes proper, improper, and absent mask wearing because these conditions require different compliance assessments or interventions.

Figure 6: **Representative visual targets across application scenarios.** Each example illustrates a niche visual scenario, its application requirements, and the corresponding application-specific visual targets established through target discovery and human refinement.

Table 8: **Quality inspection results.** Distribution of inspected annotation batches according to their image-level pass rates. The majority of inspected batches achieve a pass rate of at least 99%, while only a small fraction fall below the 95% acceptance threshold.

Image-level Pass Rate	Batches	Ratio
[99%, 100%]	40	81.6%
[95%, 99%)	7	14.3%
[0, 95%)	2	4.1%
Total	**49**	**100%**

pool for subsequent taxonomy construction, while avoiding extreme sample-count disparities caused purely by differences between source datasets.

For task-oriented sources such as Roboflow 20VL and Roboflow 100, category-aware resampling is adopted to suppress dominant categories and improve visual diversity. For large-scale sources such as Open Images V6 and V3Det, sampling is performed at the scenario level to control the overall benchmark size while retaining diverse and naturally long-tailed visual distributions.

Human Annotation. After taxonomy construction, the curated image pool is annotated according to the application-specific target definitions described in Section B.2. Each object instance is annotated with both its semantic category and its corresponding application-specific target.

Before formal annotation, annotators undergo a scenario-specific calibration stage. For each scenario, annotators are provided with approximately 20–50 representative images and independently perform preliminary annotations. The results are reviewed by the research team to identify disagreements in the interpretation of target definitions and visual decision boundaries. When ambiguous cases arise, researchers jointly review the visual evidence and application requirements. Representative instances are explicitly examined as *include* or *exclude* cases to identify where the current target definition fails to provide a consistent decision boundary. The target definition and corresponding annotation criteria are then refined when necessary, and the revised interpretation is communicated to annotators together with representative examples.

Formal batch annotation begins only after annotators demonstrate sufficient understanding of the scenario-specific target definitions. During formal annotation, ambiguous cases are handled through the same researcher review process. When the operational decision boundary is clarified or refined, annotators revisit previously labeled images affected by the revised interpretation to maintain consistent target assignment throughout the scenario.

Quality Control. We conduct scenario-level random inspection throughout the formal annotation process. For each scenario, at least 10% of the annotated images are sampled for quality inspection, with a minimum inspection set of 50 images. The inspection focuses on bounding-box correctness, target-label consistency, and annotation completeness.

A three-level acceptance protocol is adopted according to the image-level pass rate of each inspected batch. If the pass rate is $\geq 99\%$, only the identified erroneous images are corrected. If the pass rate is between 95% and 99%, the erroneous images and their corresponding annotation groups are re-annotated to ensure consistency within the batch. If the pass rate is below 95%, the entire batch is returned for re-annotation. This hierarchical protocol allows the revision effort to be proportional to the observed annotation errors.

The resulting inspection statistics are summarized in Table 8. Among the 49 inspected batches, 40 (81.6%) achieved an image-level pass rate of at least 99%, while only two batches (4.1%) fell below the 95% threshold. These results indicate that the majority of annotations satisfy the acceptance criterion after the initial annotation and calibration process, with more extensive revision concentrated on a small number of batches.

Quality control also operates at the target-definition level. When recurring disagreements reveal an ambiguous operational target boundary, the corresponding target definition and decision criteria are revisited by the research team. The revised interpretation is accompanied by representative include/exclude examples and propagated to affected annotations. Annotators then revisit previously labeled images affected by the revised guideline to maintain consistency within the scenario.

Table 9: **Annotation revision statistics.** Distribution of scenarios and annotated images according to the number of revision rounds required during quality control. Most scenarios require no revision or only one revision round.

Revision Rounds	Scenarios	Scenario Ratio	Images
0 rounds	24	55.8%	24,034
1 round	14	32.6%	19,316
2 rounds	4	9.3%	6,211
$\geq$ 3 rounds	1	2.3%	3,016
Total	**43**	**100%**	**52,577**

The overall revision statistics are reported in Table 9. Across the 43 scenarios, 24 scenarios (55.8%) require no revision after the initial annotation, while only five scenarios (11.6%) require two or more revision rounds. This distribution suggests that the calibration and guideline refinement process is sufficient for most scenarios, while iterative revision remains available for scenarios with more ambiguous operational visual boundaries.

Thus, the quality-control process jointly addresses two complementary aspects: instance-level annotation accuracy and target-level boundary consistency. The former ensures the correctness and completeness of individual annotations, while the latter ensures that operational visual boundaries remain consistent with the application requirements and human interpretation underlying each target.

C Benchmark and Evaluation Protocol

C.1 Benchmark and Evaluation Metrics

We conduct all experiments on AST-50K, which contains 52,577 images covering 43 application scenarios and 117 application-specific targets. Each target is defined according to the corresponding application scenario and target requirement.

We evaluate all methods using bounding-box detection metrics, including mAP, F1 score, precision, and recall. The mAP metric follows the PASCAL VOC all-point interpolation protocol Everingham et al. (2010), computed as the area under the precision-recall curve using all recall points. Results are reported under two IoU thresholds, IoU=0.15 and IoU=0.50. We adopt VOC-style evaluation rather than COCO-style multi-threshold averaging because PVTA primarily concerns semantic discrimination between target definitions rather than precise sub-pixel localization; the two selected IoU thresholds already cover the spectrum from operationally ambiguous visual boundaries (IoU=0.15) to a stricter localization criterion (IoU=0.50).

Unless otherwise specified, all methods are evaluated under the same benchmark protocol and metric definitions as those used in the main paper.

C.2 Representative Evaluation Subset

Several experiments require iterative model interaction and therefore involve substantially higher inference cost. For these experiments, we use a restricted subset of AST-50K, referred to as the **Representative Evaluation Subset**. Representative evaluation subset contains five representative scenarios: *Actions*, *Cable-Damage*, *Smoke*, *Egg*, and *Recode-Waste*.

Representative evaluation subset is used for the oracle study and TCC ablation in the main paper. It is a computationally restricted evaluation subset rather than a separate benchmark split, and is not intended to replace the full AST-50K evaluation.

C.3 Compared Methods

We compare representative models from several visual recognition paradigms. The evaluated models include Qwen3.6-Flash Bai et al. (2025), Seed2.0-Pro Seed Team (2026), YOLOE-26 Chakrabarty

(2026), SAM3 Carion et al. (2026), GPT-5.2 OpenAI (2025), GPT-5.6-Sol OpenAI (2026a), Rex-Omni Jiang et al. (2026), and Nano-Banana-Pro Google DeepMind (2025).

These models cover open-vocabulary detection, vision-language reasoning, and generative visual paradigms. The comparison is intended to characterize the transferability and limitations of existing visual recognition paradigms on PVTA rather than to establish a universal ranking among foundation models.

D Implementation Details

D.1 Workflow Configuration

Our workflow orchestration agent consists of an MLLM-based Planner and candidate modules containing SAM3, Seed2.0-Pro, Rex-Omni, and a Fuser. The Planner determines the workflow and invokes the visual tools according to the current target and feedback.

We denote the full configuration with TCC and all available visual tools as PLANNER(TCC, MODULES). To isolate the contribution of heterogeneous tool orchestration from target calibration, we also consider PLANNER(TCC, SEED), which restricts the candidate module to Seed2.0-Pro.

D.2 Model and Prompt Configuration

We describe the prompt templates and inference protocol for each component below. We present the core structure of each prompt; full templates (in both Chinese and English) are provided in the supplementary material.

Detection Models. The candidate modules include three detection models with distinct prompting interfaces.

Seed2.0-Pro is a vision-language model that accepts a natural-language target description. The prompt enforces a normalized 1000×1000 coordinate system and a strict output format `label confidence x0 y0 x1 y1`. The core prompt structure is:

```
You are a visual object-detection expert. Detect targets in
the images and output bounding boxes.
- User target description: {description}
- Output label must equal exactly: "{label}" (no rewrite /
expand / translate)
- Coordinate system: 1000×1000; Output format: label
confidence x0 y0 x1 y1
```

SAM3 is an open-vocabulary detector that takes a list of English phrases as text input:

```
Detect {phrase1,phrase2,...}. Output bounding boxes in [x0,
y0, x1, y1] format.
```

An optional confidence threshold can be specified as a hyperparameter.

Rex-Omni is another open-vocabulary detector invoked programmatically as:

```
predictor(source=image_path, text=phrases)
```

where `phrases` is a non-empty list of English short phrases.

Planner. The Planner is an MLLM that outputs a complete workflow as a JSON object at each round. Its system prompt establishes three key mechanisms: (1) image-centric input—round 1 receives clean images, rounds $k>1$ receive feedback-overlaid images; (2) Annotated demonstration images-visibility strategy—approximately 50% of images are sampled per round to reveal TP/FP/FN, and the Planner must not hallucinate errors on unsampled images; (3) tool-deprecation rules—a tool is deprecated after ≥ 2 consecutive poor rounds, unless it ranks first on current FP/FN.

The output JSON contains a `reasoning` field and an ordered `steps` array, where each step specifies a tool (`seedpro_detect`, `sam3_detect`, `rex_omni_detect`, or `fuse`) and its arguments. The core prompt structure is:

```
You are an object-detection orchestration Planner. Inputs
are image-centric: turn 1 inspects clean images; from turn
2, inspect previous Feeder boxed feedback overlays (green TP
/ red FP #n / purple FN #n). Output one complete workflow
JSON and stop.

Annotated demonstration images visibility: sample ∼50%
of images; expose TP/FP/FN only on sampled images. Do not
invent missing IDs.

Output JSON: reasoning (string) + steps (array of {id, tool,
args}).
Tools: seedpro_detect (zeroshot/fewshot) | sam3_detect |
rex_omni_detect | fuse.
Deprecation: poor for ≥2 consecutive turns → deprecate,
unless ranks #1 on FP/FN this turn.
```

TCC Feedback Prompts. The TCC uses two separate MLLM prompts for semantic and linguistic feedback.

The *semantic feedback* prompt (refined target description) asks the model to act as a target-definition calibrator. It receives the original description, visibility constraints, and a structured TP/FP/FN summary, then outputs a JSON object with six fields: `aligned` (whether the original description matches visible targets), `issue` (brief mismatch description), `refined_description` (corrected description for the next round), `include_visuals` (what to detect), `exclude_visuals` (what not to detect), and `planner_hint` (one-sentence hint). The core instruction is:

```
You are the target-definition calibrator. Calibrate
strictly to the visible scope; do not invent unseen
error types. Output one JSON: aligned, issue,
refined_description, include_visuals, exclude_visuals,
planner_hint.
```

The *linguistic feedback* prompt (error description with correction guidance) asks the model to act as a human reviewer and provide at most three sentences of natural-language correction advice. It must reference specific FP/FN indices from the overlaid images, stay within the visible subset, and only discuss how to adjust bounding boxes—it must not rewrite the target description. The core instruction is:

```
You are a human-like reviewer. Write at most three
sentences of box-revision advice. Use FP #n / FN #n to
reference specific errors. Obey visibility constraints. Do
not rewrite the target description.
```

D.3 Interaction Setting and Feedback Sampling

Interaction Setting. We configure the maximum interaction budget to 10 rounds. At each round, the Planner invokes the available candidate modules and updates the current workflow according to the feedback returned by the TCC. This budget is chosen to be sufficient for identifying an effective workflow while remaining practical for deployment. As analyzed in Appendix E.1, only 26.5% of targets attain their best observed workflow in a single round, while 73.5% require two or more rounds of interaction. The empirical CDF shows that the first three rounds cover 52.1% of targets, increasing to 70.9% by round 5 and 88.9% by round 8, with the remaining targets extending through round 10. This distribution demonstrates that best-workflow exploration requires a non-trivial interaction budget, and the 10-round setting is sufficient, reasonable, and effective under our proposed task and benchmark. The substantial variation in interaction needs across targets further validates the motivation of PVTA: a single-round specification is often inadequate for application-specific visual targets.

Feedback Sampling. At each round, the TCC samples 50% of the demonstration images and their corresponding feedback, which are provided to the Planner as input for the next round. This sampling strategy is motivated by three considerations. First, each round should have a focused calibration direction for the Planner's orchestration; sampled feedback prevents the Planner from being overwhelmed by an excessively large context and encourages it to prioritize the most salient errors rather than diluting attention across all cases. Second, this design mirrors the attentional allocation in real human multi-round feedback: human reviewers naturally focus on a limited set of issues per round and cannot guarantee uniform feedback quality across every image. As shown in Appendix E.3, human-guided orchestration operates under the same feedback protocol and achieves competitive performance, confirming that the sampling design aligns with realistic human feedback patterns. Third, sampling avoids over-exposing feedback information to the Planner. Without sampling, the TCC would effectively provide the full ground-truth annotations of the demonstration images as input, which constitutes a form of information leakage that does not correspond to real application scenarios where complete ground truth is unavailable at inference time.

D.4 Target Information Ablation Settings

For the experiments in Table 4, we compare three target-information configurations for *Seed2.0-Pro*: the baseline receives only the final target name; *Examples + Captions + Seed2.0-Pro* additionally receives a small set of target examples with corresponding captions describing their intended characteristics; the *Oracle + Seed2.0-Pro* setting provides complete and ideal target examples that fully specify the desired visual boundary, serving as a reference under complete target information without iterative calibration or tool orchestration.

D.5 Fuser Implementation

When a workflow invokes multiple detection models, the `fuse` module merges their outputs through a deterministic pipeline:

1. **Normalization:** All detections are converted to a normalized 1000×1000 coordinate system.
2. **Low-confidence filtering:** Detections with confidence < 0.05 are discarded.
3. **First NMS:** Boxes are sorted by confidence descending; overlapping boxes with IoU $> nms_iou$ are suppressed.
4. **Weighted merging:** Remaining boxes with IoU $\geq merge_iou$ are clustered into groups. Each group produces one merged box: coordinates are the confidence-weighted average of group members, confidence is the maximum in the group, and the source is recorded as `merged(model1, model2, ...)`.
5. **Second NMS:** A final NMS pass with threshold $\max(0.3, nms_iou)$ removes any overlaps reintroduced by merging.

The `fuse` step takes a list of preceding detection step IDs as input and produces the final bounding-box predictions for the round.

D.6 Nano-Banana-Pro Evaluation Protocol

Table 10: **Nano-Banana-Pro subset evaluation protocol.**

Item	Setting
Number of scenarios	43
Images per scenario	20
Random seed	42
Model	`gemini-3-pro-image-preview`
Aspect ratio	16:9

Due to the substantially higher inference cost of Nano-Banana-Pro, we evaluate it on a fixed subset of AST-50K instead of the complete benchmark. The subset contains 20 images from each of the

43 scenarios, resulting in 860 images in total. The subset is constructed using a fixed random seed of 42. For each image, Nano-Banana-Pro is prompted to generate a flat multi-class color mask. The prompt template and subsequent color-based decoding protocol are adapted from Liu et al. (2026); Han et al. (2026):

```
Turn this image into a flat segmentation mask using only
solid colors.  Paint {{category_color_list}}, and everything
else solid black.  Each object must be filled completely as
one single-color region from edge to edge --- no internal
details should be visible.  No textures, no gradients.
```

where the placeholder {{category_color_list}} is instantiated by concatenating one entry per target class using the following format:

```
{phrase} solid {color_name} RGB({r}, {g}, {b})
```

Each target class is represented using a fixed solid color, while non-target regions are rendered as background. The resulting image is subsequently converted into detections through color-based decoding and connected-component analysis.

The target classes within each scenario are assigned to a fixed palette consisting of red, lime, blue, and yellow. Pixels are assigned to the nearest palette color using Euclidean RGB distance, and pixels whose distance from all palette colors exceeds 45 are treated as background. Connected components with width or height smaller than 5 pixels or an area smaller than 50 pixels are removed. All remaining detections are assigned a confidence score of 1.0.

Nano-Banana-Pro is evaluated using the same detection metrics as the other methods, including mAP, mF1, precision, and recall at $\mathrm{IoU} = 0.15$ and $\mathrm{IoU} = 0.50$. To control for differences caused by image composition, Seed2.0-Pro and SAM3 are also evaluated on the identical 860-image subset. These results are reported separately from their full-benchmark results in Table 3.

E Supplementary Results and Analysis

E.1 Effect of Interaction Rounds

We further investigate how many interaction rounds are needed for the Planner to identify its best observed workflow. After K rounds, the final workflow $\mathcal{W}^*$ is selected by maximizing AP at $\mathrm{IoU} = 0.15$ against ground truth (Eq. 6). Rather than reporting a per-round performance table, we classify each of the 117 AST-50K targets by the round at which this selected workflow first occurs (ties assigned to the earliest round): a target is *single-round* if $\mathcal{W}^*$ is attained at $k{=}1$, and *multi-round* otherwise.

Figure 7(a) shows the distribution of k^*, the earliest round at which the best observed workflow is identified. The concentration at $k^*{=}1$ and $k^*{=}3$, together with the single- versus multi-round comparison in Figure 7(b), shows that only a minority of targets attain their best observed workflow from the initial specification. For the remaining 73.5%, the best observed workflow first appears after additional interaction. This indicates that a single interaction is often insufficient to identify the most effective workflow, while the amount of interaction needed varies across application-specific targets.

As shown in Figure 7(b), only 31 targets (26.5%) attain their best observed workflow in a single round, whereas 86 targets (73.5%) attain it at $k{\geq}2$. This distribution reveals substantial variation in the interaction budget across targets: some targets can be addressed effectively from the initial specification, while others benefit from additional interaction before the best observed workflow is identified.

No target selects its best workflow at round 2. The interaction traces suggest that the early rounds primarily serve as exploratory stages, during which the Planner probes the available visual tools and accumulates scenario-specific evidence. The subsequent concentration at round 3 is consistent with a transition from tool exploration to workflow consolidation, where the accumulated evidence can be incorporated into a more effective workflow. This pattern suggests that interaction is not uniformly distributed across rounds, but instead reflects different stages of workflow construction.

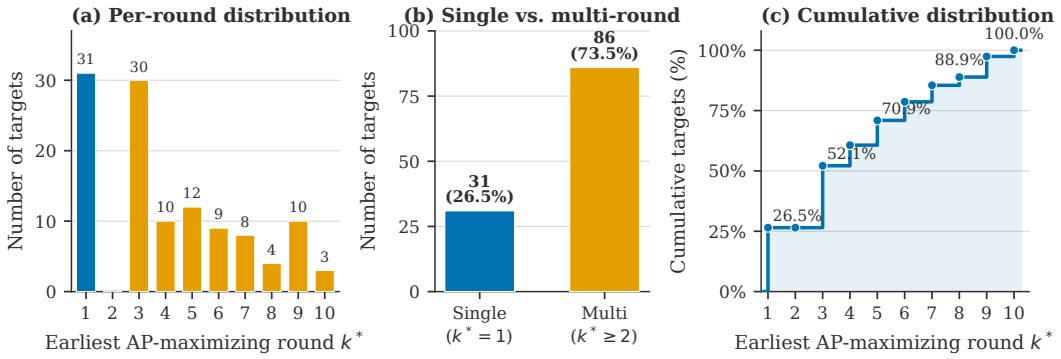


Figure 7: **Distribution of the selected best workflow $\mathcal{W}^*$ over interaction rounds.** (a) Number of targets whose earliest AP-maximizing workflow occurs at each round k^*. (b) Single-round (k^*=1) versus multi-round ($k^* \geq 2$) selection. (c) Empirical CDF of k^*. A slight majority of targets are covered by the first three rounds, while the remaining targets extend through round 10, revealing substantial variation in the interaction budget needed to identify the best observed workflow. For each target, $\mathcal{W}^*$ is chosen by maximizing AP at IoU $= 0.15$ (Eq. 6); ties are assigned to the earliest round. Results use PLANNER(TCC, MODULES) with a budget of ten rounds.

The CDF in Figure 7(c) further characterizes the variation in interaction budget across targets. The first three rounds identify the best observed workflow for a slight majority of targets (61/117, 52.1%), while coverage increases to 70.9% by round 5 and 88.9% by round 8, with residual targets extending through round 10. The target-level interaction records show a corresponding spectrum of calibration difficulty. Visually distinctive targets such as *ballpoint pen*, *plastic bottle*, and *black digit on a light background* tend to appear among the early-round cases, whereas targets with subtler appearance or stronger contextual dependence—including *thin linear mark*, *fibrous board wood*, *drilled bead jewel*, and *freight truck*—occur more frequently in the later-round groups. Overall, these results reveal substantial variation in the interaction budget needed to identify an effective workflow across application-specific targets.

E.2 PLANNER MODEL ANALYSIS

We further examine whether the choice of Planner model affects the performance of the workflow. To isolate the effect of the underlying MLLM, we keep the candidate modules, interaction budget, and feedback protocol fixed, while replacing both the Planner and the TCC with the same reasoning-capable MLLM backbone. That is, when a different Planner model is used, the TCC's MLLM is correspondingly switched to the same model to ensure a consistent model configuration throughout the framework. The evaluated MLLM backbones include GPT-5.6-Sol OpenAI (2026a), Qwen3.8-27B Bai et al. (2025), Claude Opus-5 Anthropic (2026), and GPT-6.0-Astra OpenAI (2026b), spanning both open-weight and proprietary models of varying scales and generations.

As shown in Table 11, the performance difference across Planner models is surprisingly small. The best-performing Planner (Claude Opus-5) achieves 0.5665 mAP at IoU=0.15 and 0.4361 mAP at IoU=0.50, while the weakest (Qwen3.8-27B) achieves 0.5399 and 0.4088 respectively—a gap of only 0.027 in mAP at both IoU thresholds. Notably, even the latest GPT-6.0-Astra yields only marginal improvements over GPT-5.6-Sol (0.5550 vs. 0.5488 at IoU=0.15, and 0.4293 vs. 0.4255 at IoU=0.50), suggesting that raw Planner capability is not the primary bottleneck in our framework.

This observation carries two implications. First, the performance gains of our method stem primarily from the core design of the framework—the TCC's multimodal feedback generation, the iterative workflow orchestration mechanism, and the configuration of complementary candidate modules—rather than from the intrinsic reasoning power of any specific Planner model. The framework is robust to the choice of Planner backend, which enhances its practical deployability. Second, the relatively uniform performance across strong MLLMs suggests that the proposed PVTA task and AST-50K benchmark present a genuine challenge that is not trivially solved by adopting a more powerful Planner model alone. Existing multimodal large language models and other visual detec-

Table 11: **Effect of the Planner model.** All Planner variants use the same candidate modules, TCC, and interaction protocol. Results are reported on the representative evaluation subset.

	IoU = 0.15				IoU = 0.50			
Planner	mAP	mF1	Prec.	Rec.	mAP	mF1	Prec.	Rec.
GPT-5.6-Sol	0.5488	0.6784	0.6633	0.7702	0.4255	0.5694	0.5714	0.6276
Qwen3.8-27B	0.5399	0.6748	0.6913	0.7278	0.4088	0.5470	0.5807	0.5680
Claude Opus-5	0.5665	0.6898	0.6587	0.7953	0.4361	0.5734	0.5589	0.6475
GPT-6.0-Astra	0.5550	0.6794	0.6667	0.7749	0.4293	0.5668	0.5721	0.6264

tion paradigms have paid relatively little attention to this class of application-specific, progressively-calibrated detection tasks, indicating a promising direction for future research.

E.3 Comparison with Human-Guided Orchestration

To further characterize the remaining gap between automatic and human-guided workflow construction, we compare Planner(TCC, Modules) with a human-guided counterpart, Planner(Human, Modules). Both configurations use the same underlying candidate modules, Planner, interaction budget, and best-workflow selection protocol; the only difference is whether target calibration and corrective feedback are generated by the TCC or supplied through human guidance.

We recruit two domain experts with application-specific visual inspection experience to conduct the human-guided experiments. In this configuration, the TCC is removed from the framework, and the human expert provides multimodal feedback at each interaction round. The human feedback follows the same protocol as the TCC: the expert inspects the current prediction results on the demonstration images, identifies false-positive and false-negative cases, and provides corrective guidance to the Planner. The interaction budget is set to a maximum of 10 rounds. If the expert judges that the current workflow already satisfies the detection requirement, the interaction may terminate early, and the workflow at the stopping round is selected as the best workflow. If no satisfactory workflow is identified within 10 rounds, the interaction terminates and the best workflow is selected by maximizing the detection metric on the demonstration set (Eq. 6).

As shown in Table 12, Planner(TCC, Modules) achieves performance comparable to Planner(Human, Modules) across all evaluation metrics. This result indicates that the multimodal feedback generated by our TCC effectively models the corrective feedback that human experts provide during iterative target calibration. While human experts may bring richer domain knowledge and more nuanced visual interpretation, the TCC's structured feedback pipeline—combining error visualization, target description refinement, linguistic correction guidance, and per-model error verification—captures the essential elements of human feedback sufficiently well to guide effective workflow orchestration. This finding supports the practical value of the TCC as a scalable and consistent alternative to human feedback in application-specific detection scenarios.

Table 12: **Comparison between automatic and human-guided orchestration.** Both configurations use the same visual modules. Results are reported on the representative evaluation subset.

Configuration	$\text{mAP}_{0.50}$	$\text{mF1}_{0.50}$	$\text{Prec.}_{0.50}$	$\text{Rec.}_{0.50}$
Planner(TCC, Modules)	0.4255	0.5694	0.5714	0.6276
Planner(Human, Modules)	0.4071	0.5629	0.5965	0.6011

E.4 Per-Scenario Tool Performance

The main paper analyzes whether heterogeneous visual tools provide complementary capabilities across application scenarios. We provide additional per-scenario results here to expose both the overall performance distribution and the complete metric breakdown of individual tools.

E.4.1 PER-SCENARIO MAP OVERVIEW

We first provide a compact overview of tool performance across all AST-50K scenarios. Table 13 reports mAP at IoU=0.50 for each scenario and each visual tool. This table is intended to provide a high-level view of scenario-dependent tool performance, allowing the relative strengths of different tools to be inspected without the additional detail of the other evaluation metrics.

Table 13: **Per-scenario mAP of candidate visual tools on AST-50K** (IoU**=0.50).** Results are reported without additional background information.

Scenario	Ours	Seed2.0-Pro	SAM3	Rex-Omni	Qwen3.6-Flash	GPT-5.2	GPT-5.6-Sol
Actions	0.6410	0.2316	0.0000	0.2305	0.0478	0.0005	0.1586
Aerial-Pool	0.3702	0.3464	0.3755	0.3245	0.0724	0.0008	0.1690
Asbestos	0.3097	0.1895	0.3087	0.1562	0.0016	0.0016	0.1205
Bccd	0.6939	0.4745	0.0807	0.2494	0.1005	0.0721	0.4370
Billboard	0.3150	0.2064	0.4520	0.1695	0.1513	0.0525	0.1902
Book	0.3620	0.0966	0.4192	0.1663	0.0953	0.0711	0.0696
Butterfly	0.4381	0.2520	0.2444	0.2298	0.2171	0.1401	0.2215
Cable-Damage	0.0706	0.0264	0.0000	0.0010	0.0004	0.0004	0.0434
Coconut	0.3213	0.0692	0.1466	0.0645	0.0820	0.0733	0.0332
Coins	0.8749	0.4408	0.3428	0.5629	0.3485	0.0826	0.3883
Cola	0.1115	0.0320	0.2270	0.0792	0.0205	0.0250	0.0135
ComputerKeyboard	0.6146	0.3648	0.5957	0.3800	0.2331	0.1362	0.3031
Coral	0.0821	0.0509	0.0292	0.0381	0.0213	0.0036	0.0242
Cotton	0.3372	0.2814	0.2237	0.1336	0.0771	0.0204	0.1545
Crab	0.3363	0.1296	0.1376	0.0956	0.0928	0.0408	0.0972
Deer	0.4560	0.2259	0.2286	0.2522	0.0925	0.0477	0.0902
Egg	0.4809	0.2038	0.3248	0.1908	0.0517	0.0162	0.1340
Excavators	0.7897	0.3877	0.6951	0.4374	0.3148	0.2618	0.3048
Fruit Juice	0.4722	0.1607	0.6521	0.2172	0.0568	0.0118	0.0584
Fruitjes	0.4372	0.3343	0.4383	0.2994	0.2485	0.0645	0.3234
Goose	0.6051	0.2329	0.7510	0.1395	0.1612	0.1564	0.1700
Videogame	0.4747	0.4608	0.3397	0.1640	0.0805	0.0507	0.4093
Jewel	0.2387	0.0488	0.2475	0.0296	0.0234	0.0131	0.0456
Leaf-Disease	0.3858	0.0970	0.1118	0.0139	0.0875	0.0612	0.1142
Mask-Wearing	0.3658	0.0184	0.0011	0.0021	0.0018	0.0000	0.1175
Microphone	0.3826	0.0527	0.1413	0.0567	0.0403	0.0011	0.0460
DefectsWood	0.2072	0.1819	0.1883	0.1912	0.0088	0.0054	0.1227
Orionproducts	0.2621	0.0749	0.3243	0.0840	0.0131	0.0001	0.0158
Parasites	0.3462	0.2765	0.3195	0.2280	0.0404	0.0404	0.2881
Peixos-Fish	0.2036	0.1541	0.2465	0.0996	0.0308	0.0003	0.0649
Pen	0.3656	0.2416	0.5486	0.1946	0.1043	0.0466	0.2029
Pig-Detection	0.6175	0.4519	0.0596	0.3165	0.2570	0.0134	0.3253
Potato	0.3011	0.1001	0.1126	0.0445	0.0611	0.0548	0.0274
Recode-Waste	0.5131	0.1801	0.2584	0.1649	0.0179	0.0115	0.1014
Signatures	0.4173	0.2939	0.3906	0.3258	0.0090	0.0052	0.2912
Smoke	0.3228	0.3162	0.0013	0.1999	0.1775	0.1458	0.2835
Soccer-Players	0.8187	0.4771	0.5937	0.4470	0.0064	0.0001	0.2272
Stomata-Cells	0.1854	0.1654	0.0000	0.0795	0.0244	0.0141	0.1387
Taco-Trash	0.2262	0.1619	0.0365	0.0932	0.0932	0.0065	0.1095
ValentinesChocolate	0.7072	0.3381	0.5970	0.3540	0.2100	0.0091	0.2591
Vehicles	0.3699	0.1778	0.5065	0.2729	0.0178	0.0012	0.0949
Water-Meter	0.4574	0.4384	0.1771	0.1310	0.0006	0.0021	0.1529
X-Ray-Id	0.3276	0.0244	0.0704	0.2673	0.0008	0.0045	0.0005
Overall	0.4084	0.2107	0.2764	0.1864	0.0878	0.0396	0.1530

The results show substantial variation in the relative performance of individual tools across scenarios. In particular, the strongest-performing tool is not consistent across application scenarios, indicating that the utility of a visual tool depends strongly on the target and application context.

Table 14: **Detailed per-scenario metrics of individual visual tools on AST-50K.** All tools are evaluated independently without additional background information. Results are reported under both IoU thresholds. All metrics are target-macro averages.

		IoU = 0.15				IoU = 0.50			
Scenario	Tool	mAP	mF1	Prec.	Rec.	mAP	mF1	Prec.	Rec.
Overall	SAM3	0.3009	0.2829	0.2842	0.5350	0.2764	0.2623	0.2648	0.4903
	Rex-Omni	0.2358	0.3668	0.3942	0.4614	0.1864	0.3150	0.3375	0.3929
	Seed2.0-Pro	0.2678	0.3967	0.3674	0.6276	0.2107	0.3370	0.3121	0.5383
	Qwen3.6-Flash	0.1634	0.2916	0.2745	0.4542	0.0878	0.1848	0.1718	0.3028
	GPT-5.2	0.1455	0.2669	0.2679	0.3762	0.0396	0.1019	0.0956	0.1569
	GPT-5.6-Sol	0.2370	0.3538	0.2971	0.6712	0.1530	0.2642	0.2207	0.5136
Actions	SAM3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	Rex-Omni	0.2572	0.4303	0.4991	0.4170	0.2305	0.4002	0.4699	0.3816
	Seed2.0-Pro	0.2639	0.4510	0.5232	0.5597	0.2316	0.4219	0.4860	0.5280
	Qwen3.6-Flash	0.0708	0.1926	0.5245	0.1770	0.0478	0.1520	0.4451	0.1346
	GPT-5.2	0.0774	0.2323	0.3580	0.2177	0.0005	0.0179	0.0299	0.0161
	GPT-5.6-Sol	0.2464	0.4297	0.5252	0.5379	0.1586	0.3446	0.4194	0.4335
Egg	SAM3	0.3349	0.3183	0.2605	0.8943	0.3248	0.3133	0.2573	0.8480
	Rex-Omni	0.2007	0.3176	0.2792	0.5900	0.2003	0.3078	0.2721	0.5520
	Seed2.0-Pro	0.2249	0.3223	0.3129	0.7967	0.2038	0.3064	0.2972	0.7397
	Qwen3.6-Flash	0.0839	0.2026	0.1862	0.4161	0.0517	0.1573	0.1457	0.3077
	GPT-5.2	0.1112	0.2502	0.2956	0.3979	0.0162	0.0887	0.1020	0.1553
	GPT-5.6-Sol	0.1981	0.2761	0.2286	0.8888	0.1340	0.2279	0.1865	0.7655
Coins	SAM3	0.3428	0.3308	0.5989	0.3451	0.3428	0.3308	0.5989	0.3451
	Rex-Omni	0.5678	0.7125	0.6955	0.7373	0.5629	0.7098	0.6929	0.7345
	Seed2.0-Pro	0.4570	0.5197	0.3582	0.9724	0.4408	0.5073	0.3498	0.9478
	Qwen3.6-Flash	0.4576	0.5879	0.4625	0.8100	0.3485	0.4918	0.3862	0.6792
	GPT-5.2	0.1620	0.2661	0.2080	0.3892	0.0826	0.0920	0.0684	0.1485
	GPT-5.6-Sol	0.4131	0.4691	0.3181	0.9535	0.3883	0.4514	0.3059	0.9184
Smoke	SAM3	0.0048	0.0354	0.1117	0.0210	0.0013	0.0169	0.0534	0.0101
	Rex-Omni	0.4959	0.6165	0.6549	0.5957	0.1999	0.3598	0.3982	0.3334
	Seed2.0-Pro	0.6666	0.7155	0.6520	0.7927	0.3162	0.4452	0.4067	0.4917
	Qwen3.6-Flash	0.5192	0.6405	0.6489	0.6333	0.1775	0.3421	0.3430	0.3417
	GPT-5.2	0.5094	0.5673	0.4688	0.7187	0.1458	0.2878	0.2385	0.3630
	GPT-5.6-Sol	0.6977	0.7319	0.6658	0.8130	0.2835	0.4368	0.3991	0.4826

E.4.2 Detailed Per-Scenario Metrics

To provide the complete quantitative evidence underlying the tool analysis, Table 14 reports mAP, mF1, precision, and recall for every evaluated tool under both IoU thresholds. Unlike the preceding overview, which is designed for rapid inspection of scenario-level performance, this table provides the full metric breakdown used to characterize individual tool behavior.