CountEx: Fine-Grained Counting via Exemplars and Exclusion

Yifeng Huang[1], Gia Khanh Nguyen[2], and Minh Hoai[2]

[1] Stony Brook University, Stony Brook, NY, USA
[2] Australian Institute for Machine Learning, Adelaide University, SA, Australia

Abstract. This paper presents CountEx, a discriminative visual counting framework designed to address a key limitation of existing prompt-based methods: the inability to explicitly exclude visually similar distractors. While current approaches allow users to specify what to count via inclusion prompts, they often struggle in cluttered scenes with confusable object categories, leading to ambiguity and overcounting. CountEx enables users to express both inclusion and exclusion intent, specifying what to count and what to ignore, through multimodal prompts including natural language descriptions and optional visual exemplars. At the core of CountEx is a novel Discriminative Query Refinement module, which jointly reasons over inclusion and exclusion cues by first identifying shared visual features, then isolating exclusion-specific patterns, and finally applying selective suppression to refine the counting query. To support systematic evaluation of fine-grained counting methods, we introduce CoCount, a benchmark comprising 1,780 videos and 10,086 annotated frames across 97 category pairs. Experiments show that CountEx achieves substantial improvements over state-of-the-art methods for counting objects from both known and novel categories. The data, code, and model are available at `https://github.com/bbvisual/CountEx`.

Keywords: Object Counting · Class-agnostic · Few-shot Counting

1 Introduction

Visual counting is a fundamental task in computer vision, with applications ranging from crowd monitoring to medical imaging, and more recently, a benchmark for evaluating the reasoning abilities of vision-language models. While current models can estimate object counts from textual prompts or annotated exemplars, fine-grained counting in scenes with multiple coexisting object categories remains challenging. Models often misinterpret user intent, overcount distractors, or default to dominant classes, underscoring the need for more precise and controllable mechanisms to specify both what to count and what to ignore.

We propose here a more expressive interface for visual counting that enables users to explicitly indicate both what they want to count and what they want to exclude, as illustrated in Fig. 1. Users can specify their intent through natural language descriptions alone, or optionally provide exemplar bounding

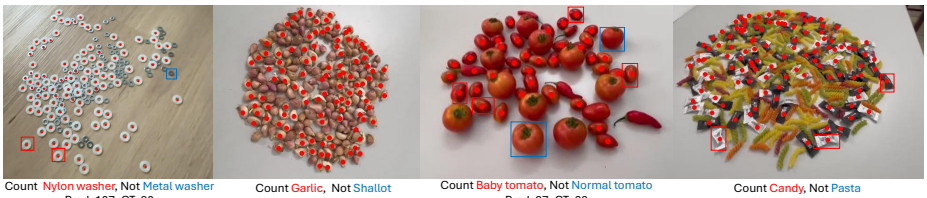


Fig. 1: Given a cluttered scene containing multiple object categories, our method allows users to specify both inclusion and exclusion intent, *e.g.*, "count nylon washers, not metal washers," via language prompts and optional visual exemplars (red and blue bounding boxes), enabling more precise and controllable counting.

boxes for additional visual guidance. For instance, a user might specify, "Count Nylon washers, not Metal washers" or "Count white poker chips, not the blue ones." When visual disambiguation is needed, a few example bounding boxes (e.g., three per category) can complement the text descriptions, as shown in Fig. 1. By allowing explicit specification of negative examples, this approach reduces ambiguity and improves counting accuracy in complex scenes with visually similar objects. For brevity, we refer to the objects the user wants to count as **positive**, and those they explicitly want to exclude as **negative**. Note that scenes may also contain **distractor objects**, other categories not mentioned by the user, that should neither be counted nor treated as explicit negatives.

Although the idea of explicitly specifying what not to count is intuitive and seemingly straightforward, it remains largely unexplored in the literature. While it is reasonable to expect that negative examples can help disambiguate user intent, it is still unclear how much they help and how best to integrate them into the counting pipeline. Current counting methods are typically rigid in design, accepting user intent for only one object category at a time and lacking the flexibility to interpret both inclusion and exclusion intent. A naive workaround might involve treating positive and negative exemplars separately, computing individual counts or density maps for each, and subtracting one from the other. However, such disjoint treatment ignores the relational context between what should and shouldn't be counted, and often yields poor results.

In this paper, we propose a novel counting architecture that jointly reasons about both positive and negative intent throughout the entire pipeline. Our method encodes multimodal prompts (text and optional exemplars) to extract positive queries and negative-exclusive references, then applies discriminative query refinement to suppress distractor patterns before final counting. By considering inclusion and exclusion simultaneously, the model faithfully captures user intent in scenes with visually similar or co-occurring categories.

To develop a method capable of fine-grained counting using both positive and negative exemplars, we introduce a new dataset, referred to as **CoCount**. Unlike existing counting datasets, which typically feature only a single object category in sufficient quantity, such as [6, 32, 38], or multiple categories with too few annotated examples, such as [7, 31]. CoCount contains 1,780 videos and 10,086 annotated frames spanning 97 category pairs, including both *inter-category pairs* (such as bolts and nuts) and *intra-category pairs* (such as hex nuts and square

nuts). Each image includes two co-occurring object types with ample instances, supporting both training and evaluation of fine-grained counting methods that must reason over inclusion and exclusion intent. This design addresses a key limitation of prior datasets, which often lead models to ignore user prompts and instead count the dominant category due to dataset bias. CoCount is the first dataset to enable large-scale, exemplar-based fine-grained counting in such complex multi-object scenarios.

In summary, we make three contributions. First, we formulate the task of counting with explicit exclusion cues, allowing users to specify both what to count and ignore. Second, we propose CountEx, a new architecture that jointly reasons over inclusion and exclusion signals, achieving the state-of-the-art performance on multiple benchmarks. Third, we introduce CoCount, a fine-grained benchmark with 10,086 annotated images across 97 category pairs, supporting scalable research on exemplar-based counting.

2 Related Work

Visual counting. Early visual counting methods were predominantly class-specific, requiring extensive labeled data for each target category, e.g., [1,5,11,15, 18–20,22,25,26,29,37,39,40,43,45,48]. To enable generalization across arbitrary categories, class-agnostic counting methods emerged, regressing density maps by computing spatial correlations between query images and visual exemplars [12, 13,21,27,32,36,38,41,50]. Subsequent works introduced prototype matching and attention refinement to improve performance [33,34,41,44,47,51]. Exemplar-free approaches [10,36] reduce annotation burden but lack control in multi-category scenarios.

Counting with language prompts. Vision-language models enabled text-guided counting, where natural language descriptions replace or complement visual exemplars [2,3,14,16,23,42,49,52,54]. ZSC [49] pioneered this by constructing visual prototypes from text, while CLIP-Count [14] and CounTX [2] leveraged CLIP [35] for cross-modal alignment. Recent methods extended GroundingDINO [3,6,23,24,46]. However, these methods share a fundamental limitation: they specify only *what to count* through positive prompts, lacking mechanisms to explicitly exclude visually similar distractors, leading to ambiguity in dense scenes with co-occurring categories.

Counting with negative prompts. While language-guided methods specify target objects through positive prompts, they cannot explicitly exclude visually similar distractors. Recent work [7] tackles this issue via test-time adaptation, synthesizing confounding categories with diffusion models and applying hard negative supervision with empty attention maps. T-Rex-Omni [53] further introduces negative visual prompts into generic object detection by jointly encoding positive and negative prompts and suppressing negative responses during probability computation. In contrast, our work studies negative prompts in few-shot object counting, using them as lightweight inference-time cues for

target–distractor disambiguation without per-category optimization, synthetic data generation, or detector fine-tuning.

3 CoCount Dataset

CoCount is created to enable the systematic development and evaluation of counting models that must discriminate fine-grained visual distinctions to perform well. CoCount is organized into five supercategories (*Food*, *Game*, *Home*, *Desk*, and *Misc*), encompassing 55 distinct categories. Each category is further divided into up to two subcategories, consisting of visually similar variants that differ in attributes such as color, size, or shape. The supercategories, categories, and subcategories are listed in Fig. 2.

Video recording. CoCount consists of images depicting scenes in which two object categories co-occur in substantial quantities. The final images are extracted from recorded videos, beginning with the identification of 100 object category pairs. We first create 50 inter-category pairs by pairing categories within the same supercategory, such as pasta and candy in Food, or chess pieces and poker chips in *Game*. We then form 50 intra-category pairs, composed of subcategories within the same category to evaluate fine-grained attribute discrimination, such as black versus white peppercorn, or AA versus AAA batteries.

For each of the 100 category pairs, we record 20 videos per pair, organized into 10 distinct count settings, each filmed twice with objects re-scrambled and reoriented. We start from the maximum feasible counts for both categories and progressively decrease the counts across five successive groups. Within each group, the two companion videos have closely matched counts (5–10% difference), while each subsequent group exhibits a larger decrement. This grouped design intentionally creates near-duplicate scenes with controlled object removals, enabling the data to serve as a verification or contrastive resource for training. For every video, we first determine the number of objects from each category and create a scene by spreading them across a surface (e.g., table, bed, floor, or tablecloth). To increase scene diversity and realism, we also add distractor objects not belonging to either target category. After quality inspection, we remove three intra-category pairs due to insufficient video quality or visual distinction, resulting in 97 final category pairs (50 inter-category and 47 intra-category).

Data split and annotation. These videos vary in object categories, counts, backgrounds, and layouts. Each video contains multiple frames annotated with two dominant object categories and their counts. This makes the dataset a valuable resource for training and evaluation. To facilitate experimentation, we divide the videos into four disjoint subsets: **Train**, **Train2**, **Val**, and **Test**, and extract representative frames for annotation. The number of videos, along with additional statistics such as the number of annotated frames and available annotation types for each subset, are summarized in Fig. 2 and described below.

For the **Train** set, where dense annotations are needed to train counting models, we adopt a semi-automated annotation pipeline designed to balance quality

and efficiency. Specifically, we manually annotate one clear frame per video with a dot at the center of each object instance, along with three exemplar bounding boxes for each of the two categories in the category pair. These annotations are then propagated to the remaining frames using CoTracker3 [17], an off-the-shelf pixel tracker. After temporal sampling and quality filtering, we retain 7,417 annotated frames for training, covering a wide range of object configurations, occlusion levels, and scale variations.

Due to the labor-intensive nature of dot annotation—even at a rate of one frame per video—we did not annotate the **Train2** set and therefore exclude it from the experiments in this paper. Nonetheless, we report its availability and will release it as part of the full dataset. This subset could be annotated in the future or used for semi- or weakly-supervised learning, as the total object counts per video are known and can provide useful weak supervision.

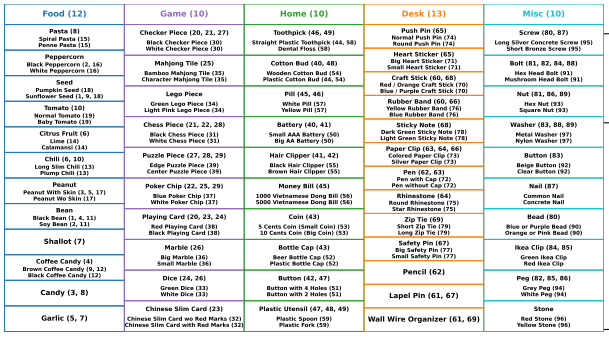
Supported experiment settings. The five supercategories enable two complementary evaluation protocols for fine-grained counting. The first is the **Novel-Category Setting (NC-setting)**, which evaluates a model's ability to generalize to object categories not seen during training. This setting takes advantage of the dataset's supercategory structure: Test-Food, Test-Game, Test-Home, Test-Desk, and Test-Misc. For each test subset, all training data containing the same categories are excluded. For example, when evaluating on Test-Food, the model is trained using Train-Game, Train-Home, Train-Desk, and Train-Misc, while Train-Food is omitted entirely.

The second protocol is the **Known-Category Setting (KC-setting)**, in which the training set includes categories that also appear in the test set. Note that while object categories may overlap, the test images are extracted from videos that are strictly disjoint from those used to extract the training images. In this setting, a single model is trained on the full training data (i.e., Train-Food, Train-Game, Train-Home, Train-Desk, and Train-Misc) and evaluated across all test subsets. This setting reflects scenarios where some labeled data is available for each target category.

For the **Val** set, we randomly sample three frames per video (1,335 frames in total) to provide a balanced subset for hyperparameter tuning. For the **Test** set, we manually select three high-quality frames per video based on clarity (minimal motion blur and good lighting) and visual diversity (viewpoint and layout), resulting in 1,334 test frames after discarding one low-quality example. To support both exemplar-based and text-guided methods, we provide three manually annotated exemplar bounding boxes per object variant in all test frames.

4 CountEx

We develop a model for fine-grained open-vocabulary counting that can leverage explicit negative prompts when available as follows. Given an image I and a positive text prompt T^{pos} describing the objects the user wants to count, the model may also receive any combination of the following optional inputs: positive exemplar bounding boxes E^{pos}, a negative text prompt T^{neg}, and negative



	Train	Train2	Val	Test
#videos	445	445	445	445
#frames	7417	0	1335	1334
#objects	1,159K	–	204K	204K
Available annotation				
Counts for two cat.	✓	✓	✓	✓
Boxes for six obj.	✓			✓
Dots for all obj.	✓			

Fig. 2: Left: Object categories and their variants across five supercategories in CoCount. The number(s) next to each category name indicate the pair ID(s) with which the category is paired. Right: statistics and available annotations for each dataset split.

exemplar bounding boxes E^{neg}. The goal is to predict the total number of instances belonging to the positive object category specified by T^{pos}. The model is expected to flexibly handle varying input configurations, whether none, one, or all optional inputs are provided.

4.1 Key challenge and approach overview

A key question is how to effectively use the negative prompt when it is available. This is not as straightforward as it may seem, since the negative prompt is not simply the opposite of the positive one. Naively subtracting one from the other can lead to suboptimal results, as will be seen in our experiments below. For example, in the prompt "count red apples, not green ones," the user intends to exclude green apples, but both red and green apples belong to the same category. Subtracting their embeddings may shift focus toward color differences rather than reinforcing the notion of apples as the relevant object class.

This nuance is important because scenes often contain other, unrelated object types. The model must distinguish between positive, negative, and irrelevant categories, not just contrast subtypes within a category. Designing mechanisms that handle this kind of exclusion without collapsing the broader object context is therefore essential for fine-grained counting.

To address this challenge, we introduce CountEx, which features a novel *Discriminative Query Refinement (DQR)* module. CountEx builds on a query-based detection model [8], which begins with an image-and-prompt–conditioned query set $\mathbf{Q}$, predicts detection boxes and scores (B_i, s_i), and outputs the total count by enumerating detections with scores above a confidence threshold: $N = \sum_{i=1}^{n} \delta(s_i > \tau)$. CountEx extends this model by encoding the input image along with positive and negative prompts into separate query sets $\mathbf{Q}^{\text{pos}}$ and $\mathbf{Q}^{\text{neg}}$ (Sec. 4.2), and applying DQR (Sec. 4.3) to refine $\mathbf{Q}^{\text{pos}}$ by isolating negative-exclusive features and suppressing them while preserving shared,

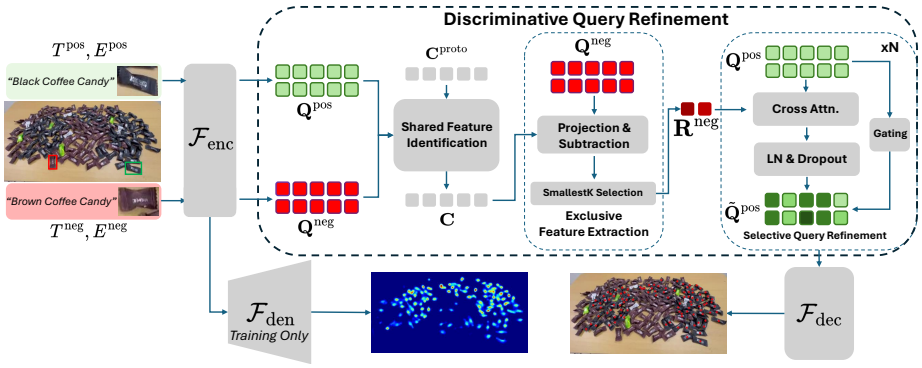


Fig. 3: Overview of CountEx. Given an image and multimodal prompts (positive and negative text with optional visual exemplars), we encode them into query sets $\mathbf{Q}^{pos}$ and $\mathbf{Q}^{neg}$. Our Discriminative Query Refinement (DQR) module consists of three stages: **(1)** Shared Feature Identification learns prototypes $\mathbf{C}$ capturing common features between both query sets; **(2)** Exclusive Feature Extraction isolates negative-exclusive patterns $\mathbf{R}^{neg}$ by projecting $\mathbf{Q}^{neg}$ onto $\mathbf{C}$ and filtering residuals; **(3)** Selective Query Refinement produces refined queries $\tilde{\mathbf{Q}}^{pos}$ by selectively suppressing negative patterns via attention. The refined queries are fed to detection heads for final predictions. An auxiliary density prediction branch provides additional supervision during training.

category-relevant cues. This enables accurate and discriminative counting in the presence of subtle attribute-level differences. See Fig. 3 for an overview.

4.2 Prompt-Conditioned Query Encoding

Our approach is compatible with any open-vocabulary detector that supports multimodal conditioning, such as Grounding DINO [23] or similar architectures. In this work, we adopt the query encoder from LLMDet [8], denoted as $\mathcal{F}_{enc}$, due to its strong open-vocabulary capabilities. The encoder $\mathcal{F}_{enc}$ is a transformer encoder-decoder architecture where learnable queries cross-attend to vision and language features to produce a set of n queries encoding potential object instances. To incorporate both text and visual exemplars, we construct multimodal conditioning sequences as follows: text descriptions are encoded via the language encoder to obtain token embeddings. For visual exemplars, each crop $e \in E^{pos} \cup E^{neg}$ is encoded by extracting patch features, average-pooling to obtain $\mathbf{v}_e \in \mathbb{R}^d$, projecting through a linear layer, and concatenating with text embeddings. We then condition the query encoder separately on the positive and negative prompts (T_{pos}, E_{pos}) and (T_{neg}, E_{neg}) to obtain two query sets:

$$\mathbf{Q}^{pos} = \mathcal{F}_{enc}(I, T^{pos}, E^{pos}), \quad \mathbf{Q}^{neg} = \mathcal{F}_{enc}(I, T^{neg}, E^{neg}),$$

where $\mathbf{Q}^{pos}, \mathbf{Q}^{neg} \in \mathbb{R}^{n \times d}$. Each query set encodes object candidates aligned with its corresponding prompt, forming the foundation for discriminative refinement. Importantly, by generating separate positive and negative query sets from the

same image, we obtain two complementary representations of the scene: one focused on the target objects and the other on visually similar distractors. This dual encoding enables our DQR module to explicitly reason about the distinction between what to count and what to exclude.

4.3 Discriminative Query Refinement

Given the positive and negative query sets from Sec. 4.2, we refine $\mathbf{Q}^{\text{pos}}$ using a discriminative process that preserves category-relevant features while selectively suppressing those unique to the negative prompt. Rather than relying on naive subtraction, which can eliminate critical features of the target class, we adopt a three-stage refinement strategy: (1) extract shared features, (2) identify negative-exclusive components, and (3) apply targeted suppression to remove only the truly discriminative negatives.

Shared Feature Identification Given $\mathbf{Q}^{\text{pos}}, \mathbf{Q}^{\text{neg}} \in \mathbb{R}^{n\times d}$, this stage identifies shared features by learning r prototypes that capture visual attributes common to both positive and negative queries. The intuition is that these prototypes should respond equally to queries encoding shared attributes such as object shape, texture, or category-level properties. By explicitly modeling a shared feature subspace, we can subsequently decompose negative queries into components that overlap with positive queries (which should be preserved) and components unique to negatives (which should be suppressed).

We introduce r learnable prototype embeddings $\mathbf{C}^{\text{proto}} \in \mathbb{R}^{r\times d}$, initialized as normalized random vectors. To make these prototypes context-aware, we fuse them with information from both query sets via cross-attention:

$$\mathbf{H} = \mathit{MHA}(\mathbf{C}^{\text{proto}}, [\mathbf{Q}^{\text{pos}}; \mathbf{Q}^{\text{neg}}], [\mathbf{Q}^{\text{pos}}; \mathbf{Q}^{\text{neg}}]), \quad \mathbf{C} = \mathit{LN}(\mathit{Linear}(\mathbf{H})), \tag{1}$$

where *MHA* denotes the multi-head attention block, *LN* stands for the layer normalization operation, and *Linear* refers to the linear layer. The concatenation $[\mathbf{Q}^{\text{pos}}; \mathbf{Q}^{\text{neg}}] \in \mathbb{R}^{2n\times d}$ merges both sets along the sequence dimension, and $\mathbf{C} \in \mathbb{R}^{r\times d}$ represents the shared query prototypes.

The loss for prototype learning is $\mathcal{L}_{\text{proto}} = \lambda_{\text{share}}\mathcal{L}_{\text{share}} + \lambda_{\text{div}}\mathcal{L}_{\text{div}}$, where:

$$\mathcal{L}_{\text{share}} = -\frac{1}{r}\sum_{j=1}^{r}\left[\max_i \langle \mathbf{c}_j, \mathbf{q}_i^{\text{pos}}\rangle + \max_i \langle \mathbf{c}_j, \mathbf{q}_i^{\text{neg}}\rangle\right]; \mathcal{L}_{\text{div}} = \|\mathbf{C}^{\text{proto}}\mathbf{C}^{\text{proto}^\top} - \mathbf{I}\|_F^2.$$

The $\mathcal{L}_{\text{share}}$ loss encourages each prototype to be similar to at least one query from each set, while the diversity loss prevents collapse by encouraging orthogonal prototype embeddings. In the above, $\langle\cdot,\cdot\rangle$ denotes cosine similarity and $\mathbf{c}_j$ denotes the j-th prototype.

Exclusive Feature Extraction. Given shared prototypes $\mathbf{C} \in \mathbb{R}^{r\times d}$, this module extracts features unique to the negative prompt through projection-based

decomposition followed by exclusivity-based filtering. This two-stage design extracts negative-exclusive patterns while discarding shared features that would harm positive detection if naively subtracted.

We first identify negative queries that are most distant from the shared feature space. For each negative query $\mathbf{q}_i^{\text{neg}}$, we compute an exclusivity score based on its maximum similarity to any shared prototype: $\sigma_i = \max_j \langle \mathbf{q}_i^{\text{neg}}, \mathbf{c}_j \rangle$, where lower scores indicate further distance from the shared space, suggesting features more exclusive to the negative category. Let $\mathcal{I}_{\text{excl}}$ denote the set of indices i corresponding to the m smallest σ_i values.

For each selected query, we then decompose it into shared and exclusive components through subspace projection. The exclusive component is obtained by projecting onto the subspace spanned by $\mathbf{C}$ and extracting the residual. This projection removes the shared component, retaining only the negative-exclusive residual. Assembling the residuals from selected queries yields the negative-exclusive reference set:

$$\mathbf{R}^{\text{neg}} = \{\mathbf{q}_i^{\text{excl}} \mid i \in \mathcal{I}_{\text{excl}}\} \in \mathbb{R}^{m \times d}, \text{where } \mathbf{q}_i^{\text{excl}} = \mathbf{q}_i^{\text{neg}} - \mathbf{q}_i^{\text{neg}} \mathbf{C}^\top \mathbf{C}. \quad (2)$$

This two-step design, that first selecting queries with high exclusivity, then extracting their exclusive components, yields a compact set of distinctive negative patterns for subsequent refinement, free from shared features that characterize both categories.

Selective Query Refinement. Given negative-exclusive references $\mathbf{R}^{\text{neg}}$ from the previous stage, this module selectively refines positive queries by suppressing negative patterns while preserving discriminative information. As illustrated in Fig. 3, positive queries $\mathbf{Q}^{\text{pos}}$ are refined through cross-attention to negative-exclusive features: queries attend to $\mathbf{R}^{\text{neg}}$ to identify alignment with distractor patterns, and the attended features are then subtracted via a gated residual connection. Specifically, we apply cross-attention where $\mathbf{Q}^{\text{pos}}$ serves as queries and $\mathbf{R}^{\text{neg}}$ serves as keys and values, followed by layer normalization and dropout to obtain $\tilde{\mathbf{Q}}^{\text{pos}}$. A learnable gating parameter controls the suppression strength, enabling the model to adaptively balance between aggressive negative filtering and preservation of positive instances. This design follows residual learning principles: queries with low attention to $\mathbf{R}^{\text{neg}}$ pass through largely unchanged, while queries strongly aligned with negative patterns are selectively suppressed. Since $\mathbf{R}^{\text{neg}}$ contains only negative-exclusive features by construction, this refinement avoids removing shared visual attributes and prevents false negatives. The refined queries $\tilde{\mathbf{Q}}^{\text{pos}}$ retain discriminative information about positive instances while being cleaned of distractor patterns. These refined queries are then fed to the dot decoder $\mathcal{F}_{\text{dec}}$ with standard detection heads for classification and box regression to produce final predictions $\{(b_i, s_i)\}_{i=1}^n$.

4.4 Training Objective

Following common practice in point-supervised object counting [3, 6, 46], our model is trained end-to-end using a multi-component loss that combines clas-

sification, localization, and density prediction terms, along with the prototype learning loss described above: $\mathcal{L} = \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}} + \lambda_{\text{den}}\mathcal{L}_{\text{den}} + \mathcal{L}_{\text{proto}}$.

The Classification Loss applies focal loss over the matched pairs: $\mathcal{L}_{\text{cls}} = \sum_{i=1}^{n} \text{FocalLoss}(s_i, y_i)$, where s_i is the predicted score for bounding box B_i, and y_i is the assigned ground-truth label.

The Localization Loss supervises the centers of the matched predicted bounding boxes using the ground-truth point annotations: $\mathcal{L}_{\text{loc}} = \sum_{i=1}^{k} \|\text{center}(B_i) - C_i\|_1$, where B_i is the predicted box for the i-th query, C_i is the ground-truth point coordinate, k indicates the number of matched queries.

The Density Prediction Loss provides complementary dense supervision to promote spatially-aware feature learning. We extract text-fused visual tokens from the encoder (prior to query decoding) and pass them through an FPN-based density head to produce a predicted density map $\hat{D}$. For supervision, we construct pseudo ground-truth density maps D from point annotations by placing a 2D Gaussian kernel at each annotated point location. The density prediction is then supervised using mean squared error loss: $\mathcal{L}_{\text{den}} = |\hat{D} - D|_2^2$.

5 Experiments

This section describes the experiments conducted to evaluate CountEx across various settings. We benchmark CountEx on CoCount under both novel and known category settings, then assess its generalization on LOOKALIKES [7], PairTally [31], and FSC-147 [38]. We also conduct ablations on key components and discuss limitations.

5.1 Results on CoCount

We evaluate CountEx on CoCount under two settings: counting Novel Categories (NC) and Known Categories (KC), as outlined in Sec. 3. We compare against four strong baselines (LLMDet [8], CAD-GD [46], GroundingREC [6], and CountGD [3]) trained under the same protocols. We also compare against three alternative approaches for leveraging negative prompts and exemplars, built on LLMDet: Detection-Suppression, which detects each category separately and suppresses spatially overlapping predictions; Density-Subtraction, which predicts per-category density maps and subtracts the negative from the positive with zero-flooring; and Count-All-then-Halve, which counts all objects and divides the result by two. Following standard practice [27, 32, 38, 50], we use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for evaluation. Unlike baselines that only use inclusion prompts, CountEx leverages both inclusion and exclusion cues (textual and visual).

Table 1 presents results on both evaluation protocols. On the task of counting novel object categories (NC-setting), where models train on four supercategories and test on the held-out fifth, our method achieves 26.61 MAE and 38.86 RMSE. Compared to our base architecture LLMDet (33.22 MAE), we achieve a substantial 19.9% error reduction. On the task of counting objects from known

Method	NC-setting		KC-setting	
	MAE	RMSE	MAE	RMSE
CAD-GD [46]	34.08	50.39	16.00	27.52
GroundingREC [6]	29.29	42.43	17.54	27.41
CountGD [3]	33.78	48.29	15.55	28.32
LLMDet [8]	33.22	47.66	16.82	29.23
LLMDet + Detection-Suppression	28.93	42.17	16.94	29.71
LLMDet + Density-Subtraction	47.71	74.05	24.19	38.93
LLMDet + Count-All-then-Halve	31.51	46.80	19.77	31.66
LLMDet + DQR (the proposed CountEx)	**26.61**	**38.86**	**12.72**	**23.99**

Table 1: Performance comparison on the CoCount test set. All methods are trained using the same data under either the known-category (KC) or novel-category (NC) setting. CountEx outperforms all baselines across both settings.

categories (KC-setting), where all five categories are available during training, our method achieves 12.72 MAE and 23.99 RMSE, outperforming the best baseline (CountGD at 15.55 MAE) by 18%. Notably, while all methods perform better when trained with access to all categories, our method maintains consistent advantages across both settings. The relative improvement over LLMDet is substantial in both protocols (19.9% on NC-setting and 24.4% on KC-setting), indicating that our approach benefits both zero-shot and supervised scenarios.

5.2 Evaluation on Other Datasets

To assess generalization beyond CoCount, we evaluate on the LOOKALIKES benchmark [7], a test-only dataset with 1,037 images across 27 fine-grained subcategories. For CountEx and the starred baselines, models are trained on the full CoCount training split before zero-shot evaluation on LOOKALIKES. Other baselines are evaluated without additional training or adaptation, following prior reports. As shown in Table 2a, CountEx achieves an MAE of 18.53 under zero-shot transfer without fine-tuning, setting a new state-of-the-art among zero-shot methods. D'Alessandro et al. [7] report a lower MAE of 10.00, but their method relies on synthetic data generation and test-time adaptation per category, incurring 5–7 minutes of processing time per class.

We further evaluate our proposed method on the PairTally benchmark [31], comparing it against both specialist counting models and general vision-language models as reported in [31]. Results are presented in Table 2b, reporting MAE and Normalized Absolute Error (NAE) for both inter-scene and intra-scene settings, following the evaluation protocol in [31]. CountEx achieves superior performance across all metrics, outperforming not only pre-trained specialist counters and general vision-language models, but also models trained on the same CoCount dataset used for training CountEx. Another popular benchmark is FSC-147 [38], though it primarily contains a single object category per image and therefore offers no opportunity to leverage exclusion prompts. For fair comparison with other methods trained specifically on this dataset, we fine-tune CountEx on the FSC-147 training set and evaluate it using inclusion prompts (from [2]) only. On the

Method	MAE	RMSE
D'Alessandro et al. [7]	10.00	16.95
(per category data synthesis + adaptation)		
Zero-shot - no training/adaptation		
OWLv2 [30]	37.25	55.01
GroundingDINO [23]	33.89	59.51
PseCo [13]	59.82	72.79
DAVE [34]	56.10	73.34
CountGD [3]	22.34	33.90
LLMDet* [8]	22.86	33.29
CAD-GD* [46]	28.14	41.30
GroundingREC* [6]	27.24	41.57
CountGD* [3]	23.55	36.89
CountEx* (proposed)	18.53	30.46

(a) Results on LOOKALIKES [7]. * denotes training on CoCount before zero-shot evaluation. CountEx achieves best zero-shot performance. D'Alessandro et al. [7] obtain lowest MAE but require synthetic data generation and per-category test-time adaptation (5–7 minutes per category).

Model	MAE ↓		NAE ↓	
	Inter	Intra	Inter	Intra
Pre-trained specialist counting models				
DAVE [34]	46.27	46.75	0.779	0.797
GeCo [33]	45.05	54.80	0.777	0.935
LoCA [44]	71.89	57.45	1.177	0.950
FamNet [38]	66.97	74.75	1.363	1.440
Count GD [3]	39.78	56.54	0.673	0.906
Count GD (Text) [3]	50.23	53.93	0.712	0.841
LLMDet [8]	78.72	142.08	0.661	1.060
Pre-trained large vision language models				
Ovis2 [28]	56.87	74.24	0.711	0.736
Qwen2.5-VL [4]	46.35	67.86	0.598	0.712
LLaMA-3.2 [9]	49.14	58.73	0.730	0.740
InternVL3 [55]	55.89	71.47	0.667	0.721
Models trained on CoCount				
LLMDet [8]	18.78	21.06	0.225	0.368
CAD-GD [46]	19.47	16.98	0.243	0.259
GroundingREC [6]	19.71	17.91	0.260	0.349
CountGD [3]	19.67	15.67	0.263	0.318
CountEx (proposed)	15.61	12.57	0.197	0.229

(b) Results on PairTally [31]. CountEx achieves the best performance on both MAE and NAE, outperforming pre-trained specialist counters, general vision-language models, and models trained on the same CoCount dataset.

Table 2: Evaluation on external benchmarks.

test set, CountEx achieves an MAE of 8.63, outperforming several recent methods, including DAVE (8.66), CACViT (9.13), LOCA (10.79), CountTR (11.95), VLCounter (17.05), GroundingREC (10.12), and CAD-GD (10.35), but remaining behind the state-of-the-art CountGD (5.74). This gap highlights a limitation of CountEx on single-category counting benchmarks, where its exclusion-prompt design cannot be fully exploited.

5.3 Error Analysis

Fig. 4 provides a detailed error breakdown on the KC-setting. As shown in (a), MAE increases with ground-truth count, rising from 5.0 for counts in the 1–20 range to 43.1 for counts above 200. This is expected, as higher counts involve denser and more occluded scenes. Interestingly, (b) reveals that the relative error is highest for low-count images (35.8%) and decreases to around 15–17% for counts above 50, indicating that the model handles dense scenes proportionally well. The mean relative error across all ranges is 21.1%. The hardest category pairs shown in (c) involve objects with highly similar shapes and textures, such as craft sticks vs. rubber bands (MAE 179) and nails vs. screws (MAE 99), highlighting that extreme visual similarity remains the primary challenge for fine-grained discriminative counting.

5.4 Efficiency Analysis

We evaluate the inference efficiency of DQR on a single NVIDIA A5000 GPU using batch size 1. Latency is averaged over 1000 distinct images from the CoCount-

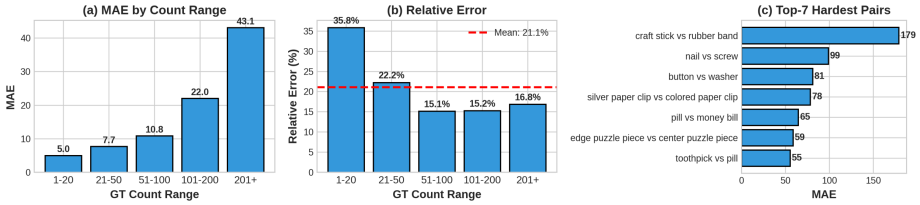


Fig. 4: Error analysis on CoCount (KC-setting). (a) MAE increases with ground-truth count, from 5.0 for counts 1–20 to 43.1 for counts above 200. (b) Relative error is highest (35.8%) for low counts and stabilizes around 15–17% for counts above 50, with a mean of 21.1%. (c) The hardest category pairs involve objects with highly similar shapes and textures (*e.g.*, craft stick vs. rubber band, nail vs. screw).

(a) Effect of swapping positive and negative prompts

(b) In-the-wild results on images outside the training and test sets, demonstrating CountEx's generalization to diverse real-world scenarios.

Fig. 5: Qualitative results on CoCount test set and in-the-wild data.

test set. CountEx achieves an average latency of 194.0 ms per image with DQR, compared to 190.2 ms without DQR. Therefore, DQR introduces only 3.8 ms additional latency, corresponding to approximately 2% overhead.

5.5 Ablation Studies

Prompt combinations. Table 3b evaluates performance across different combinations of input modalities on the test data of CoCount. We find that negative text prompts provide substantial improvements: on the KC-setting the MAE decreased from 15.96 to 13.22.

Using irrelevant negative prompts. To further analyze the role of exclusion prompts, we conduct an additional experiment using irrelevant negative prompts that specify distractors not present in the scene. For instance, in a scene containing only black and brown candies where the goal is to count black ones, we replace the correct exclusion prompt ("not brown") with an unrelated one ("not

Variant	$\mathcal{L}_{\text{den}}$	$\mathcal{L}_{\text{proto}}$	MAE	RMSE
w/o both			15.20	28.33
w/o $\mathcal{L}_{\text{proto}}$	✓		13.00	25.22
w/o $\mathcal{L}_{\text{share}}$	✓	($\mathcal{L}_{\text{div}}$)	13.64	25.07
w/o $\mathcal{L}_{\text{div}}$	✓	($\mathcal{L}_{\text{share}}$)	14.05	26.28
Full	✓	✓	12.72	23.99

(a) Loss components (KC).

T_{pos}	E_{pos}	T_{neg}	E_{neg}	MAE	RMSE
✓				15.96	28.76
✓	✓			15.75	27.59
✓		✓		13.22	26.23
✓	✓	✓	✓	12.72	23.99

(b) Prompt modality (KC).

Variant	MAE	RMSE
DQR components		
Full DQR	12.72	23.99
w/o Shared Feature Id.	14.18	26.48
w/o Exclusive Feature Ex.	13.40	25.30
w/o Selective Query Ref.	16.10	30.49
Negative queries m		
$m = 16$	13.87	26.26
$m = 32$	13.18	24.37
$m = 64$ (default)	12.72	23.99
$m = 128$	12.99	23.70

(c) Architecture ablations (KC).

Table 3: Ablation studies on the KC-setting. (a) Loss component ablation. (b) Prompt modality ablation. (c) DQR component and negative query count ablation.

red"). Using such mismatched negatives, the model achieves an MAE of 28.43 and RMSE of 42.29 in the NC-setting, worse than using relevant negatives (MAE 26.61) but still better than using no exclusion prompt at all (MAE 32.22). This result highlights two points: (1) relevant exclusion prompts are crucial for optimal performance, and (2) even irrelevant exclusion cues can offer modest benefits due to CountEx's discriminative query embedding and selection mechanisms.

Impact of loss components. Table 3a evaluates the contribution of different loss components. Adding the density prediction loss $\mathcal{L}_{\text{den}}$ improves KC-setting MAE from 15.20 to 13.00, demonstrating that dense supervision helps the encoder learn spatially-aware representations. Incorporating the prototype losses $\mathcal{L}_{\text{proto}}$ further reduces MAE to 12.72, showing that explicit guidance for learning shared and negative-exclusive patterns is crucial for discriminative query refinement. We further ablate $\mathcal{L}_{\text{share}}$ and $\mathcal{L}_{\text{div}}$: removing $\mathcal{L}_{\text{share}}$ degrades MAE from 12.72 to 13.64, and removing $\mathcal{L}_{\text{div}}$ to 14.05.

DQR component ablation. Table 3c isolates the contribution of each DQR stage and the number of negative queries m on the KC-setting. Among DQR components, removing Selective Query Refinement causes the largest degradation (MAE 12.72 $\rightarrow$ 16.10), as the model loses its ability to suppress negative-exclusive patterns from positive queries. Removing Shared Feature Identification increases MAE to 14.18, since without explicit shared prototypes, the exclusive feature extraction cannot reliably separate shared from negative-only components. Removing Exclusive Feature Extraction yields MAE 13.40, confirming that the projection-based decomposition contributes meaningfully to isolating distractor patterns. Performance is also stable across the number of selected negative queries m.

5.6 Qualitative Results

Figure 1 demonstrates accurate counting across visually similar categories. Figure 5a shows results on identical images when swapping positive and negative

text prompts. Fig. 5b further shows in-the-wild results on images outside the training and test data, where CountEx successfully distinguishes fine-grained categories such as blueberries vs. strawberries and white vs. blue boats.

5.7 Prototype Interpretability

To better understand the learned shared prototypes, we visualize the queries most strongly associated with one representative prototype. Specifically, for each prototype, we retrieve the top-16 queries by cosine similarity across all CoCount-test queries and crop image patches around their reference points. As shown in Fig. 6, the retrieved patches form a coherent visual concept: elongated rod-like structures shared across diverse categories such as nails, screws, safety pins, paper clips, and craft sticks. Quantitatively, the 1024 nearest queries to the 64 prototypes contain positives with a ratio of 49.67% on average (std 4.52%) per image, close to the 50% expected when prototypes capture features shared between the positive and negative branches.

Fig. 6: Visualization of learned shared prototypes. We retrieve the top-16 queries most similar to a representative prototype and crop patches around their reference points. The retrieved patches correspond to a coherent shared visual concept, namely elongated rod-like structures, across different object categories.

6 Conclusions

We introduced **CountEx**, a discriminative counting framework that leverages multimodal prompts to specify both inclusion and exclusion intent, enabling accurate fine-grained object counting. To support rigorous evaluation, we presented **CoCount**, a benchmark with dense annotations across 97 fine-grained category pairs. CountEx achieves state-of-the-art performance on CoCount and strong generalization to LOOKALIKES and PairTally, while remaining competitive on FSC-147 where exclusion cues are largely unavailable.

Acknowledgment. This work was funded by the Australian Institute for Machine Learning (Adelaide University) and the Centre for Augmented Reasoning, an initiative by the Department of Education, Australian Government. The authors would also like to thank Viet Bach Tran for assistance with data collection.

References

1. Abousamra, S., Hoai, M., Samaras, D., Chen, C.: Localization in the crowd with topological constraints. In: Proceedings of AAAI Conference on Artificial Intelligence (2021)
2. Amini-Naieni, N., Amini-Naieni, K., Han, T., Zisserman, A.: Open-world text-specified object counting. arXiv:2306.01851 (2023)
3. Amini-Naieni, N., Han, T., Zisserman, A.: Countgd: Multi-modal open-world counting. Advances in Neural Information Processing Systems (2024)
4. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-VL technical report (2025)
5. Bai, S., He, Z., Qiao, Y., Hu, H., Wu, W., Yan, J.: Adaptive dilated network with self-correction supervision for counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
6. Dai, S., Liu, J., Cheung, N.M.: Referring expression counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2024)
7. D'Alessandro, A., Mahdavi-Amiri, A., Hamarneh, G.: FiGO: Fine-grained object counting without annotations. arXiv preprint arXiv:2504.11705 (2025)
8. Fu, S., Yang, Q., Mo, Q., Yan, J., Wei, X., Meng, J., Xie, X., Zheng, W.S.: Llmdet: Learning strong open-vocabulary object detectors under the supervision of large language models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2025)
9. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The Llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
10. Hobley, M., Prisacariu, V.: Learning to count anything: Reference-less class-agnostic counting with weak supervision. arXiv preprint arXiv:2205.10203 (2022)
11. Hu, Y., Jiang, X., Liu, X., Zhang, B., Han, J., Cao, X., Doermann, D.: Nas-count: Counting-by-density with neural architecture search. In: Proceedings of the European Conference on Computer Vision (2020)
12. Huang, Y., Ranjan, V., Hoai, M.: Interactive class-agnostic object counting. In: Proceedings of the International Conference on Computer Vision (2023)
13. Huang, Z., Dai, M., Zhang, Y., Zhang, J., Shan, H.: Point segment and count: A generalized framework for object counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2024)
14. Jiang, R., Liu, L., Chen, C.: Clip-count: Towards text-guided zero-shot object counting. In: Proceedings of the ACM Multimedia Conference (2023)
15. Jiang, X., Zhang, L., Xu, M., Zhang, T., Lv, P., Zhou, B., Yang, X., Pang, Y.: Attention scaling for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
16. Kang, S., Moon, W., Kim, E., Heo, J.P.: Vlcounter: Text-aware visual representation for zero-shot object counting. In: Proceedings of AAAI Conference on Artificial Intelligence (2024)
17. Karaev, N., Makarov, Y., Wang, J., Neverova, N., Vedaldi, A., Rupprecht, C.: Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In: Proceedings of the International Conference on Computer Vision (2025)
18. Laradji, I.H., Rostamzadeh, N., Pinheiro, P.O., Vazquez, D., Schmidt, M.: Where are the blobs: Counting by localization with point supervision. In: Proceedings of the European Conference on Computer Vision (2018)

19. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
20. Lian, D., Li, J., Zheng, J., Luo, W., Gao, S.: Density map regression guided detection network for rgb-d crowd counting and localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
21. Liu, C., Zhong, Y., Zisserman, A., Xie, W.: Countr: Transformer-based generalised visual counting. arXiv preprint arXiv:2208.13721 (2022)
22. Liu, C., Weng, X., Mu, Y.: Recurrent attentive zooming for joint crowd counting and precise localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
23. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: Proceedings of the European Conference on Computer Vision (2024)
24. Liu, S., Zhang, P., Zhang, S., Ke, W.: CountSE: Soft exemplar open-set object counting. In: Proceedings of the International Conference on Computer Vision (2025)
25. Liu, X., Yang, J., Ding, W., Wang, T., Wang, Z., Xiong, J.: Adaptive mixture regression network with local counting map for crowd counting. In: Proceedings of the European Conference on Computer Vision (2020)
26. Liu, Y., Shi, M., Zhao, Q., Wang, X.: Point in, box out: Beyond counting persons in crowds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
27. Lu, E., Xie, W., Zisserman, A.: Class-agnostic counting. In: Proceedings of the Asian Conference on Computer Vision (2018)
28. Lu, S., Li, Y., Chen, Q.G., Xu, Z., Luo, W., Zhang, K., Ye, H.J.: Ovis: Structural embedding alignment for multimodal large language model. arXiv:2405.20797 (2024)
29. Miao, Y., Lin, Z., Ding, G., Han, J.: Shallow feature based dense attention network for crowd counting. In: Proceedings of AAAI Conference on Artificial Intelligence (2020)
30. Minderer, M., Gritsenko, A., Houlsby, N.: Scaling open-vocabulary object detection. Advances in Neural Information Processing Systems (2023)
31. Nguyen, G.K., Huang, Y., Hoai, M.: Can current AI models count what we mean, not what they see? a benchmark and systematic evaluation. In: Proceedings of International Conference on Digital Image Computing: Techniques and Applications (2025)
32. Nguyen, T., Pham, C., Nguyen, K., Hoai, M.: Few-shot object counting and detection. In: Proceedings of the European Conference on Computer Vision (2022)
33. Pelhan, J., Lukezic, A., Zavrtanik, V., Kristan, M.: A novel unified architecture for low-shot counting by detection and segmentation. Advances in Neural Information Processing Systems (2024)
34. Pelhan, J., Zavrtanik, V., Kristan, M., et al.: Dave-a detect-and-verify paradigm for low-shot counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2024)
35. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the International Conference on Machine Learning (2021)

36. Ranjan, V., Hoai, M.: Exemplar free class agnostic counting. In: Proceedings of the Asian Conference on Computer Vision (2022)
37. Ranjan, V., Le, H., Hoai, M.: Iterative crowd counting. In: Proceedings of the European Conference on Computer Vision (2018)
38. Ranjan, V., Sharma, U., Nguyen, T., Hoai, M.: Learning to count everything. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2021)
39. Ranjan, V., Wang, B., Shah, M., Hoai, M.: Uncertainty estimation and sample selection for crowd counting. In: Proceedings of the Asian Conference on Computer Vision (2020)
40. Sam, D.B., Peri, S.V., Sundararaman, M.N., Kamath, A., Babu, R.V.: Locate, size and count: Accurately resolving people in dense crowds via detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
41. Shi, M., Lu, H., Feng, C., Liu, C., Cao, Z.: Represent, compare, and learn: A similarity-aware framework for class-agnostic counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2022)
42. Shi, Z., Sun, Y., Zhang, M.: Training-free object counting with prompts. In: Proceedings of the IEEE Workshop on Applications of Computer Vision (2024)
43. Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Wu, Y.: Rethinking counting and localization in crowds: A purely point-based framework. In: Proceedings of the International Conference on Computer Vision (2021)
44. Đukić, N., Lukežič, A., Zavrtanik, V., Kristan, M.: A low-shot object counting network with iterative prototype adaptation. In: Proceedings of the International Conference on Computer Vision (2023)
45. Wang, B., Liu, H., Samaras, D., Nguyen, M.H.: Distribution matching for crowd counting. Advances in Neural Information Processing Systems (2020)
46. Wang, Z., Pan, Z., Peng, Z., Cheng, J., Xiao, L., Jiang, W., Cao, Z.: Exploring contextual attribute density in referring expression counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2025)
47. Wang, Z., Xiao, L., Cao, Z., Lu, H.: Vision transformer off-the-shelf: A surprising baseline for few-shot class-agnostic counting. In: Proceedings of AAAI Conference on Artificial Intelligence (2024)
48. Xu, C., Qiu, K., Fu, J., Bai, S., Xu, Y., Bai, X.: Learn to scale: Generating multipolar normalized density maps for crowd counting. In: Proceedings of the International Conference on Computer Vision (2019)
49. Xu, J., Le, H., Nguyen, V., Ranjan, V., Samaras, D.: Zero-shot object counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2023)
50. Yang, S.D., Su, H.T., Hsu, W.H., Chen, W.C.: Class-agnostic few-shot object counting. In: Proceedings of the IEEE Workshop on Applications of Computer Vision (2021)
51. You, Z., Yang, K., Luo, W., Lu, X., Cui, L., Le, X.: Few-shot object counting with similarity-aware feature enhancement. In: Proceedings of the IEEE Workshop on Applications of Computer Vision (2023)
52. Zhang, S., Zhou, Q., Ke, W.: Enhancing zero-shot object counting via text-guided local ranking and number-evoked global attention. In: Proceedings of the International Conference on Computer Vision (2025)
53. Zhou, J., Jiang, Q., Chen, K., Jiang, L., Lyu, Y., Chen, Y.C., Zhang, L.: T-Rex-Omni: Integrating negative visual prompt in generic object detection. In: Proceedings of AAAI Conference on Artificial Intelligence (2026)

54. Zhu, H., Yuan, J., Yang, Z., Guo, Y., Wang, Z., Zhong, X., He, S.: Zero-shot object counting with good exemplars. In: Proceedings of the European Conference on Computer Vision (2024)
55. Zhu, J., Wang, W., Chen, Z., Liu, Z., Ye, S., Gu, L., Tian, H., Duan, Y., Su, W., Shao, J., et al.: Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. arXiv preprint arXiv:2504.10479 (2025)