COUNTING BEYOND INSTANCES: BUNCHCOUNT FOR GROUP-INDIVIDUAL OBJECT COUNTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual counting is commonly formulated at the instance level, aiming to estimate how many objects of a queried category appear in an image. However, real-world counting often involves higher-level semantic units formed by multiple instances, such as a bunch of grapes, a stack of plates, or a pair of shoes. This exposes a key limitation of existing counting formulations, which mainly focus on *what to count*, while largely overlooking *at which semantic unit to count*. We introduce **Group-Individual Object Counting** (GIC), a new setting that requires models to count both individual objects and semantic groups within a unified framework. To support this new task, we present **BunchCount**, a real-world benchmark with 1,330 images, 89,254 individual annotations, and 11,065 group annotations. BunchCount provides paired individual-group annotations within the same image and explicitly records containment relations between each group and its constituent individuals. Experiments on BunchCount show that current advanced counting models perform well on individual instances but fail to count semantic groups more accurately. To mitigate semantic granularity conflict, we propose a counting-unit guided relational counting framework, which exploits group-individual containment relations to regularize cross-granularity representations during training. Our method substantially improves group-level counting while better preserving individual-level counting ability, establishing a strong baseline for counting beyond instances.

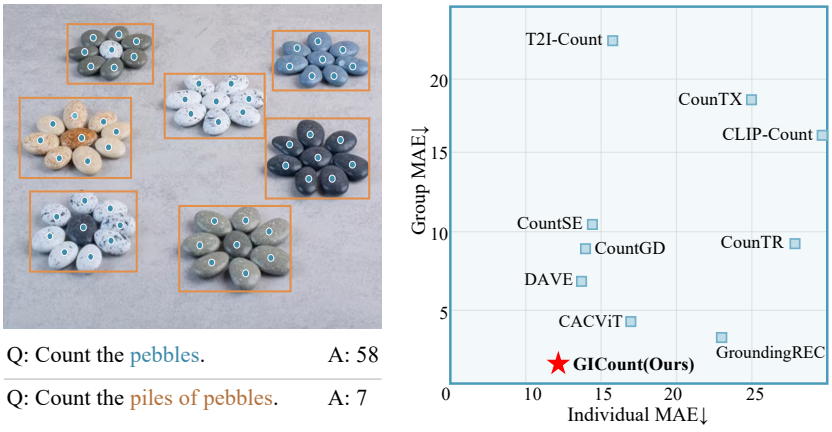
1 INTRODUCTION

Object counting aims to estimate the number of instances of target objects in an image (Ranjan et al., 2021). Recent studies have extended counting from fixed categories to more flexible target specifications, such as visual exemplars (Ranjan et al., 2021; Chang et al., 2022) or text descriptions (Amini-Naieni et al., 2023; Jiang et al., 2023). Despite these advances, most existing counting paradigms still treat individual instances as the basic counting units. They have substantially expanded *what to count*, but rarely examine *which semantic unit* should be counted.

In many practical scenarios, object counting often involves beyond individual instances for the same set of objects, which can be counted either individually or according to the semantic groups they form. For example, grapes may be counted as individual grapes or as bunches, shoes as individual shoes or as pairs, and plates as individual plates or as stacks. These two levels of counting are both meaningful and serve different practical needs (Pérez-Zavala et al., 2018). Motivated by this observation, we introduce a new counting paradigm, **Group-Individual Object Counting (GIC)**, as shown in Figure 1(a). Given an image and a prompt, a model is required to count either individual objects or the semantic groups, according to the counting unit specified by the prompt.

The GIC task is distinct from existing counting formulations. Class-agnostic (Ranjan et al., 2021) and fine-grained counting (Dai et al., 2024; Nguyen et al., 2025) extend the range of countable categories or attributes, but still typically regard individual instances as the basic counting units. Related vision tasks have explored group structure. Group-aware perception (Zhang & Hung, 2016) aims to recover group structures or uses groups as intermediate regions, while Locount (Cai et al., 2021) localizes regions of heavily overlapping same-category objects and estimates the number of constituent instances within each region. Under this background, **GIC formulates counting-unit selection as an explicit component of object counting**, allowing the prompt to specify either indi-

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.



(a) (b)

Figure 1: Illustration of group-individual object counting. (a) Group-individual object counting. (b) Counting model performance comparisons on the BunchCount.

vidual objects or the semantic groups they form. However, there is no suitable benchmark to study this problem. Existing counting datasets usually annotate targets at a single granularity (Ranjan et al., 2021), and therefore cannot evaluate whether a model can switch between individual-level and group-level counting for the same category in the same image. Moreover, since each semantic group is composed of specific constituent instances, group or individual annotations alone are insufficient to fully characterize the task. To fill this gap, we construct **BunchCount**, a real-world benchmark containing 1,330 images, 89,254 individual annotations, and 11,065 group annotations. Each image provides paired individual and group annotations, together with explicit group-individual containment relations that capture real-world group structures, including structural arrangement, physical connection, fixed cardinality, and spatial aggregation.

Furthermore, we examine whether existing counting models can serve as direct solutions to the new GIC task. Representative visual-exemplar (Chang et al., 2022), text-guided (Amini-Naieni et al., 2023), and grounding-based counters (Amini-Naieni et al., 2024) can count repeated target instances, making them natural baselines for GIC. However, their established individual-counting capability does not reliably transfer to semantic groups, as illustrated in Figure 1(b). Naive joint fine-tuning further exposes a semantic granularity conflict that improving group counting may degrade individual counting. It derives from the fact that a group is a relational unit formed by its group structures rather than a fixed-appearance object. Thus, simply mixing annotations from both granularities is insufficient to learn a unified counter. To address this challenge, we propose **GICount**, a counting-unit guided relational counting framework. GICount models the queried target as a counting unit, and exploits group-individual containment relations to regularize cross-granularity representations. This enables the model to count the intended individual or group unit while preserving structural consistency across counting granularities. Our main contributions are as follows:

- We introduce GIC, a new counting task that extends object counting from target-category selection to semantic counting-unit selection, requiring a model to count either individual objects or the semantic groups.
- We construct BunchCount, a real-world benchmark for GIC, containing paired individual- and group-level annotations in the same images, together with explicit containment relations between the group and its instances.
- We conduct a systematic benchmark study on BunchCount and show that individual-counting capability does not naturally transfer to group counting, revealing semantic granularity conflict across counting units.
- We propose GICount, a counting-unit guided relational counting framework. It models group-unit prompt and exploits group-individual containment relations to improve group counting while preserving individual counting.

2 Related Work

2.1 Object Counting Benchmarks

Early counting benchmarks are largely developed for specific domains, such as crowds (Zhang et al., 2016; Sindagi et al., 2019; Wang et al., 2021), cars (Hsieh et al., 2017) and cells (Lempitsky & Zisserman, 2010), with predefined target categories. FSC-147 (Ranjan et al., 2021) establishes the widely adopted benchmark for class-agnostic counting, where models estimate the number of objects from a target category. Subsequent benchmarks broaden this paradigm through target localization (Nguyen et al., 2022), multi-category annotation (Mondal et al., 2025), and counting question answering (Deitke et al., 2025), expanding both the range of countable targets and the ways in which they can be specified.

More recent benchmarks provide finer control over target selection. REC-8k (Dai et al., 2024) uses referring expressions to select instances according to attributes and contextual cues, while PairTally (Nguyen et al., 2025) evaluates counting in the presence of visually or semantically similar distractors. KubriCount (Liu et al., 2026) further supports target descriptions at multiple semantic abstraction levels. These benchmarks refine the semantics of a counting query and determine which object instances should be counted, but still treat each selected instance as the implicit counting unit. BunchCount addresses this missing dimension by allowing objects of the same category within an image to be counted either as individual objects or the semantic groups formed by multiple instances. Its paired individual and group annotations, together with their containment relations, enable counting units to be studied while controlling for visual content and target category.

2.2 Group Understanding and Relation Modeling

Group-level visual understanding studies how multiple people can be aggregated into a coherent whole for collective activity recognition and social group analysis (Ibrahim et al., 2016; Gavrilyuk et al., 2020; Li et al., 2021; Kim et al., 2022). Related application (Cai et al., 2021) has also represented clusters of same-category objects as localized regions and estimated the number of instances they contain. In relational modeling, Relationformer (Shit et al., 2022) introduces a dedicated relation token to model relations among DETR-style object queries, within a broader line of transformer-based scene graph methods (Im et al., 2024; Cong et al., 2023; Li et al., 2022; Hayder & He, 2024). Together, these works provide complementary precedents for aggregating multiple instances and reasoning over their relations. In object counting, GIC introduces counting granularity as an explicit query dimension, treating individuals and semantic groups as alternative counting units, with BunchCount supporting the task through paired individual and group annotations in each image and containment relations linking groups to their constituent instances.

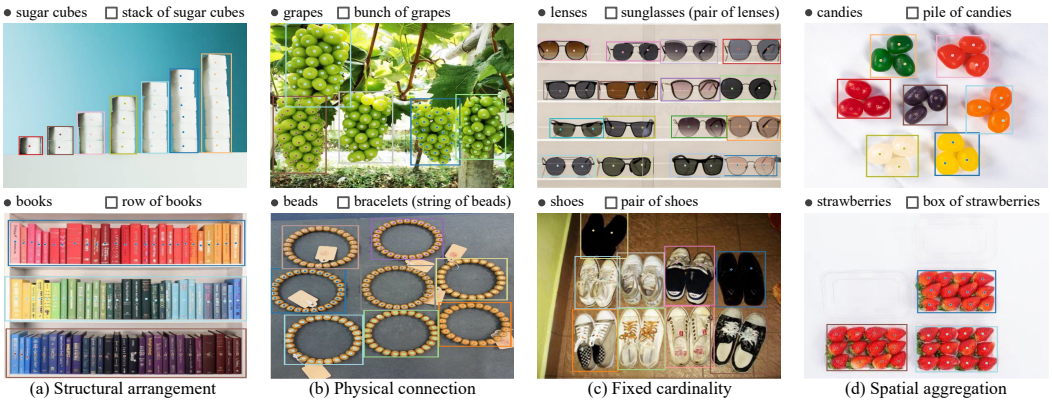
3 Group–Individual Object Counting

Object counting is typically defined at the individual-instance level. We extend it to Group-Individual Object Counting (GIC), where the prompt specifies both the target category and the semantic counting unit. Given an image I and a target category c, let $\mathcal{I}_c = \{i_1, \ldots, i_{N_{\text{ind}}}\}$ denote the set of individual instances. A semantic group is a countable higher-level unit formed by multiple instances through connection, aggregation, arrangement, or fixed cardinality. We denote the group set as $\mathcal{G}_c = \{g_1, \ldots, g_{N_{\text{grp}}}\}$. Each group g_j is associated with its constituent instances under a group-individual containment relation. Each individual belongs to at most one group, while some individuals may remain ungrouped.

Given a prompt P, the target category is c and the specified counting unit is $u \in \{\text{ind}, \text{grp}\}$. The target count N_u is

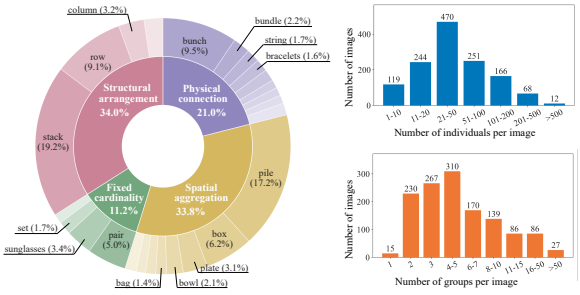
$$N_u = \begin{cases} |\mathcal{I}_c|, & u = \text{ind}, \\ |\mathcal{G}_c|, & u = \text{grp}. \end{cases}$$

Thus, the same image and category may have different valid counts under different prompts, e.g., counting individual grapes versus counting bunches of grapes. GIC requires the model to follow the specified semantic unit and count either individual instances or semantic groups accordingly.



(a) Structural arrangement (b) Physical connection (c) Fixed cardinality (d) Spatial aggregation

Figure 2: Representative samples from BunchCount, covering (a) structural arrangement, (b) physical connection, (c) fixed cardinality, and (d) spatial aggregation. Dots and boxes indicate individual and group annotations, respectively.



(a) (b)

(c)

Dataset	Images	Classes	Annotations	
			Ind.	Grp.
UCF_CC_50	50	1	1279	–
CountBench	540	–	6	–
PairTally	681	54	160	–
CARPK	1448	1	62	–
FSC-147	6135	147	56	–
REC-8k	8011	–	36	–
BunchCount	**1330**	**86**	**67**	**8**

Figure 3: Statistics and comparison of BunchCount. (a) Distribution of group structure types and descriptions. (b) Distributions of the numbers of individual and group annotations per image. (c) Comparison with existing counting datasets. Ind. and Grp. denote the average numbers of individual and group annotations per image, respectively.

4 BunchCount Dataset

To support GIC, we construct BunchCount, a real-world benchmark that provides paired individual- and group-level annotations within the same image.

4.1 Data Collection and Annotation

We collect images from FSC-147 (Ranjan et al., 2021), Locount (Cai et al., 2021), COCO (Lin et al., 2014), OpenImages (Kuznetsova et al., 2020), and the web, and re-annotate them under a unified GIC protocol. BunchCount focuses on real-world scenes involving structural arrangement, physical connection, fixed cardinality and spatial aggregation as shown in Figure 2, while retaining only images with identifiable instances, clear group boundaries, and reliable group relations. Each image is annotated at two semantic units: individual objects are labeled by center points, and semantic groups are labeled by bounding boxes enclosing their constituent instances. We further record group-individual containment relations by linking each group box to its member points, also allowing explicitly ungrouped individuals. For prompt-based counting, we additionally provide visual exemplars, including three individual boxes and one to three group boxes per image. The annotation was conducted by six annotators over approximately 183 hours, followed by three rounds of correction and consistency checking.

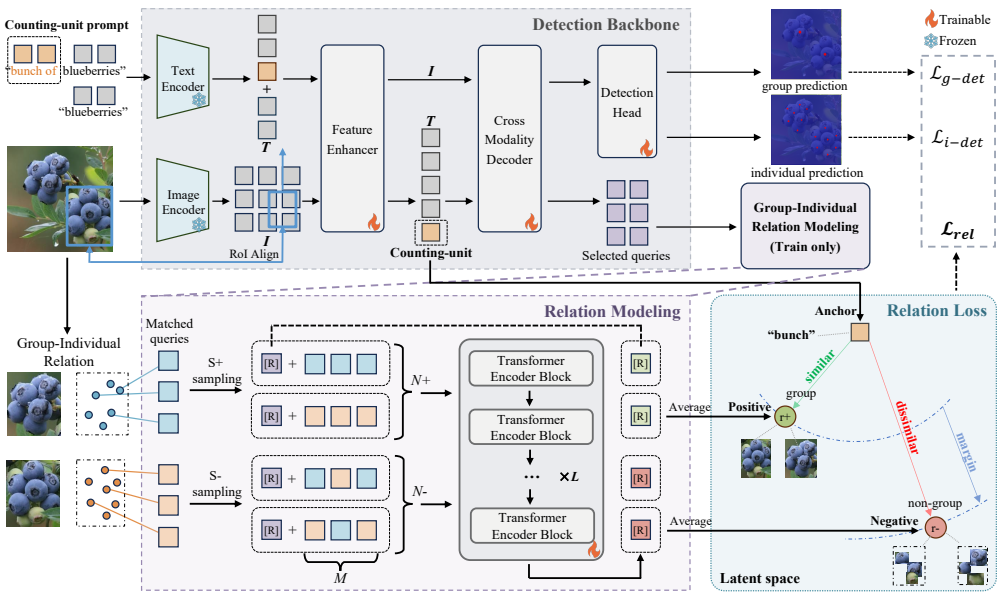


Figure 4: Overview of the GICount framework. A shared detection backbone receives a counting-unit prompt and predicts the group or individual counts, while the training-only group-individual relation modeling aggregates matched queries.

4.2 Dataset Statistics and Split

BunchCount contains 1,330 real-world images, 89,254 individual point annotations, and 11,065 group bounding-box annotations. Each image includes 4–941 individuals and 1–253 groups, with averages of 67 individuals and 8 groups, respectively. The dataset covers diverse group structures and count distributions, as shown in Figure 3(a) and (b). Figure 3(c) compares BunchCount with representative counting benchmarks, including UCF_CC_50 (Idrees et al., 2013), CountBench (Paiss et al., 2023), PairTally (Nguyen et al., 2025), CARPK (Hsieh et al., 2017), FSC-147 (Ranjan et al., 2021), and REC-8k (Dai et al., 2024). Furthermore, we split BunchCount by target category into mutually disjoint training, validation, and test sets, containing 63, 9, and 14 categories with 869, 151, and 310 images, respectively. For categories overlapping with FSC-147, we follow its original split. Each image provides paired individual- and group-level counting samples. Group-structure descriptions are shared across splits to evaluate whether models can transfer learned group-structure semantics to unseen object categories. Additional dataset analyses are provided in Appendix A.

5 GICount Framework

We present GICount, a counting-unit guided relational counting framework for GIC in Figure 4. It contains two key components: the counting-unit prompt and group-individual relation modeling.

5.1 Overview

GICount follows a detection-based counting architecture built on GroundingDINO (Liu et al., 2025a), a vision-language detector that localizes open-vocabulary targets from textual prompts. This architecture is suitable for GIC because the model localizes and enumerates the semantic counting unit specified by the prompt, rather than regressing a global count from local repetitions. As shown in Figure 4, given an $I \in \mathbb{R}^{H\times W\times 3}$, a textual prompt t, and visual exemplars, GICount first encodes visual and textual features with the GroundingDINO backbone and then decodes target regions corresponding to the queried unit.

To adapt this architecture to the GIC task, GICount introduces two key components. First, counting-unit prompt distinguishes category semantics from group-structure semantics, enabling the model to

represent either individual instances or semantic groups as counting units. Second, group-individual relation modeling uses containment relations to align group representations with the aggregated representations of their constituent individuals during training. The final count is obtained by enumerating localized predictions, allowing GICount to improve group-level counting while preserving individual-level counting capability.

5.2 COUNTING-UNIT PROMPT

In the GIC task, the counting-unit prompt specifies both the target category and the semantic counting unit. For individual-level queries, the unit is the object instance itself, and the prompt mainly encodes category semantics, e.g., "apples". For group-level queries, the unit is a semantic group, whose prompt contains a category component describing the constituent object type and a group-structure component describing how multiple instances form a countable whole, such as aggregation, arrangement, physical connection, or fixed cardinality. For explicit group prompts, we separate it from the category term, e.g., "pile of" and "apples" in "pile of apples". For holistic group names with implicit structure, e.g., "trains" composed of multiple carriages, we insert two learnable group-unit token to represent the latent group structure. The complete prompt is used for text encoding, cross-modal fusion, and decoding, while the group-unit representation is extracted for subsequent group-individual relation modeling during training. This design enables GICount to ground the queried target as a semantic counting unit.

5.3 GROUP–INDIVIDUAL RELATION MODELING

Group-level counting requires the model to associate the textual group-unit semantics with the visual organization among individuals. For example, in the prompt "pile of apples", the relation among apples belonging to the same pile should correspond to the structural semantics of "pile," whereas an arbitrary combination of apples drawn from different piles should not express the same group structure. To impose this distinction, we construct positive and negative sets of individual queries, aggregate the relations within each set, and employ metric learning to align textual group-structure semantics with visual relations among individuals.

Positive and Negative Relation-Set Sampling. For a paired group-level sample and individual-level sample from the same image, we obtain the group-structure semantic representation s from the counting-unit prompt. In the individual branch, predicted points are matched to ground-truth individual points through bipartite matching, and the decoder queries are denoted as $\{q_i\}$. According to the annotated containment relations, each matched query is assigned to its associated group if it belongs to one. We then sample positive relation sets from individuals within the same group. Negative relation sets are constructed by mixing individuals from different groups and/or ungrouped individuals. For each paired sample, we sample N^+ positive sets and N^- negative sets, each containing at most M individual queries. When fewer than M queries are available, the remaining positions are padded with padding masks.

Relation Aggregation. To encode the organization of each sampled individual set, we prepend a learnable $[\mathrm{REL}]$ token to its query sequence and feed the sequence into a Transformer-based relation aggregator with L blocks. The encoded $[\mathrm{REL}]$ token is used as the relation representation of the corresponding set, yielding positive set representations r_i^+ and negative set representations r_j^-. Finally, mean pooling is applied separately over the N^+ positive and the N^- negative representations to obtain the sample-level positive and negative relation representations r^+ and r^-.

Relation Loss. We use cosine distance $d(\cdot,\cdot)$ to align the group-unit representation with valid individual-set relations. For the k-th paired sample, we compute the positive distance $d(s_k, r_k^+)$ and negative distance $d(s_k, r_k^-)$. The relation objective is defined as

$$\mathcal{L}_{\mathrm{rel}} = \frac{1}{K}\sum_{k=1}^{K}[\alpha \max(0, d(s_k, r_k^+) - d(s_k, r_k^-) + m) + (1-\alpha)d(s_k, r_k^+)], \tag{1}$$

where s is the semantic anchor, K denotes the number of paired samples, m is the triplet margin, and $\alpha \in [0,1]$ balances the triplet loss and the pairwise positive-alignment loss. As illustrated in Figure 4, the objective pulls together the anchor and positive relations formed by individuals from the same group, while pushing apart the anchor and negative relations constructed from different

groups. This supervision establishes an explicit correspondence between textual group-structure semantics and visual patterns of individual organization.

5.4 Training and Inference

For each individual sample, we follow the objective design of CountGD and optimize a detection loss consisting of a localization term $\mathcal{L}_{\text{loc}}$ and a classification term $\mathcal{L}_{\text{cls}}$.

Localization Loss. The localization loss measures the ℓ_1 discrepancy between the predicted point coordinate $\hat{c}_i$ and its matched GT coordinate c_i over the N predictions matched to GT targets:

$$\mathcal{L}_{\text{loc}} = \sum_{i=1}^{N} \|\hat{c}_i - c_i\|_1 , \tag{2}$$

Classification Loss. The classification loss measures the discrepancy between predicted classes and their matched labels:

$$\mathcal{L}_{\text{cls}} = \text{FocalLoss}(\hat{L}, Y), \tag{3}$$

where $\hat{L}$ denotes the predicted classification confidence and Y denotes the class labels. For each paired individual- and group-level sample, the overall objective combines the detection losses of the two samples with the relation loss:

$$\mathcal{L} = \mathcal{L}_{\text{det}}^{\text{ind}} + \lambda_2 \mathcal{L}_{\text{det}}^{\text{grp}} + \lambda_3 \mathcal{L}_{\text{rel}}, \tag{4}$$

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{loc}} + \lambda_1 \mathcal{L}_{\text{cls}}. \tag{5}$$

λ_1, λ_2, and λ_3 balance the classification, group-level detection, and relation objectives, respectively.

During training, the individual- and group-level samples associated with the same image are paired and fed into the model, enabling relation aggregation and computation of the relation loss. At inference time, each sample is processed independently through the underlying detection pipeline, which predicts the locations of all queried targets and obtains the count from the number of predictions.

6 Experiments

6.1 Experimental Setup

Implementation Details. We adopt GroundingDINO-B as the backbone and initialize it with the publicly released CountGD checkpoint, while randomly initializing the proposed relation-modeling module. The model is then trained on BunchCount for 30 epochs. During training, the Swin Transformer image encoder and the BERT text encoder are frozen, and all remaining trainable parameters are optimized with Adam. The initial learning rate is set to 1×10^{-4} and multiplied by 0.1 every 10 epochs. Training is conducted on a single NVIDIA RTX 5090 GPU with a batch size of 4 and takes approximately 10 hours. We set the numbers of sampled positive and negative relation sets, N^+ and N^-, to 2, the maximum relation-set size M to 5, and the loss weights λ_2 and λ_3 to 1. Due to space constraints, complete implementation details are provided in Appendix B.1.

Evaluation Metrics. Each test image in BunchCount gives rise to one individual-level counting sample and one group-level counting sample. Since the two counting units differ substantially in numerical scale, we report their performance separately. Following standard practice in visual counting, we evaluate all methods using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), with detailed definitions provided in Appendix C.

6.2 Results on BunchCount

We compare against several representative class-agnostic counting methods spanning visual-exemplar, text, and multimodal prompting, and additionally report results from representative vision-language models (VLMs) in Table 1. The counting methods are fine-tuned on BunchCount by uniformly sampling from the merged individual- and group-level training samples, with implementation, evaluation details and cost analysis provided in Appendix B.2, B.3 and D. CountGD achieves the lowest individual-level MAE among the counting baselines (13.54), but records a group-level

Table 1: Main results on the individual- and group-level validation and test sets of BunchCount. T, V, and T+V denote text, visual, and text-plus-visual prompts, respectively. The best and second-best results are shown in **bold** and underlined, respectively. Results marked with * are computed after excluding samples with anomalous model output; details are provided in Appendix B.2.

Method	Ind. Val		Grp. Val		Ind. Test		Grp. Test	
	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓
Qwen3-VL-8B [**T**] (2025)	34.25	83.37	1.15	5.30	40.07	86.50	10.67	32.21
LLaVA-NeXT-Mistral-7B [**T**] (2024)	48.59	92.51	3.21	7.17	55.27*	99.42*	11.85	34.51
InternVL3.5-8B [**T**] (2025)	38.23	85.11	1.56	5.09	46.56	95.97	9.31	30.47
Qwen3.5-27B [**T**] (2026b)	40.00	91.39	0.87	5.33	38.28	82.26	5.84	19.86
GLM-4.5V [**T**] (2026)	29.90	75.22	0.56	1.52	44.50	131.16	8.83	27.93
Qwen3.7-Max [**T**] (2026a)	36.66	90.50	**0.38**	**1.15**	31.02	84.96	5.55	22.74
Claude Opus 5 [**T**] (2026)	30.09	82.05	0.52	1.47	22.05	49.16	10.26	33.76
GPT-5.6 Sol [**T**] (2026)	27.22	81.47	0.62	2.59	15.94	33.54	5.46	18.57
CounTR [**V**] (2022)	19.82	35.70	4.01	10.51	28.97	48.33	9.16	18.98
DAVE [**V**] (2024)	15.91	29.67	4.52	10.87	13.83	**22.07**	6.94	12.06
CACViT [**V**] (2024)	18.44	34.80	1.51	2.67	17.21	35.98	4.46	12.97
CounTX [**T**] (2023)	20.28	37.05	4.02	6.79	25.01	61.94	18.58	48.72
CLIP-Count [**T**] (2023)	22.22	36.11	2.05	3.10	38.17	82.22	16.47	54.23
GroundingREC [**T**] (2024)	**10.03**	26.89	0.97	2.02	23.17	60.48	3.59	10.60
T2ICount [**T**] (2025)	11.59	**25.43**	12.71	53.06	16.17	41.83	23.53	68.03
CountSE [**T**] (2025b)	13.20	41.60	0.76	1.62	14.70	29.18	10.35	31.75
CountGD [**T+V**] (2024)	13.05	34.05	1.77	7.79	13.54	38.68	9.21	28.30
GICount (Ours) [T+V]	13.07	31.98	2.18	9.31	**12.20**	28.89	**1.85**	**5.49**

Table 2: Effect of relation modeling and counting-unit prompt on the GICount framework.

Method	Individual Test		Group Test	
	MAE↓	RMSE↓	MAE↓	RMSE↓
w/o relation modeling	13.54	38.68	9.21	28.30
w/o counting-unit prompt	18.36	57.39	1.93	6.79
full (Ours)	**12.20**	**28.89**	**1.85**	**5.49**

MAE of 9.21. In contrast, GroundingREC obtains the strongest baseline group-level result (3.59 MAE) despite a considerably higher individual-level MAE of 23.17, possibly because its fine-grained attribute partially align with the structural semantics in group prompts. This ranking reversal shows that conventional instance-counting performance does not reliably predict the ability to count semantic groups, highlighting the distinct capability evaluated by BunchCount.

GICount achieves 1.85 and 5.49 of MAE and RMSE on the group test set, reducing the errors of GroundingREC by 48.5% and 48.2%. It also obtains an individual-level MAE of 12.20 and an RMSE of 28.89. Overall, these results establish GICount as a strong unified baseline that substantially advances semantic-group counting while retaining strong individual-level counting performance. As illustrated in Figure 5, GICount places one prediction on each intended group and shifts its predictions to the constituent instances when given the corresponding individual-level prompt.

6.3 Ablation Study on GICount

We ablate the two key components of GICount in Table 2. w/o relation modeling uses only individual- and group-level detection losses, while w/o counting-unit prompt applies relation supervision with the mean-pooled full group prompt as the anchor. Adding relation modeling sharply reduces group-level MAE from 9.21 to 1.93, showing that containment-based supervision is crucial for semantic-group counting. However, using the full prompt as the relation anchor entangles category and counting-unit semantics, increasing individual-level MAE from 13.54 to 18.36. By isolating group-structure semantics as the anchor, the full model preserves the group-level gain and

Table 3: Comparison under different training settings on the individual- and group-level test sets.

Method	Train Setting	Individual Test		Group Test	
		MAE↓	RMSE↓	MAE↓	RMSE↓
CountGD	Ind.-only	10.92	26.01	61.73	100.66
CountGD	Grp.-only	73.95	115.76	1.38	3.98
CountGD	Joint	13.54	38.68	9.21	28.30
GICount (Ours)	Joint	**12.20**	**28.89**	**1.85**	**5.49**

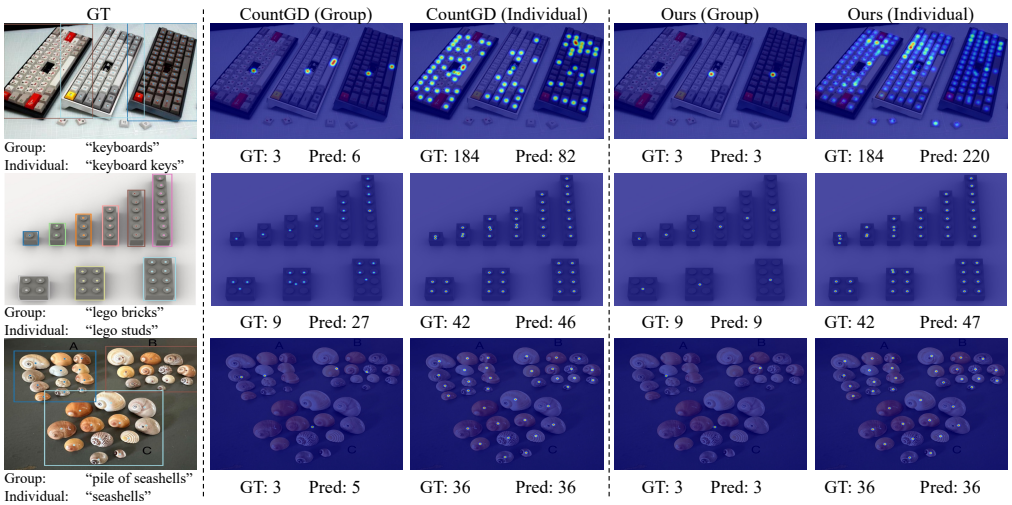


Figure 5: Qualitative results of CountGD and GICount on BunchCount under paired group- and individual-level prompts.

restores individual counting, achieving 1.85 group MAE and 12.20 individual MAE. These results show that relation modeling enhances group counting, and counting-unit prompts are necessary to avoid cross-granularity interference.

6.4 Counting-units Analysis

We examine whether mixing individual- and group-level samples is sufficient to learn a unified counter in Table 3. We train three CountGD variants using individual samples only, group samples only, or uniformly mixed samples, and report GICount under the same joint training setting. The two single-unit models show clear specialization. Individual-only and group-only achieve 10.92 and 1.38 MAE on their respective units, while their cross-unit errors reach 61.73 and 73.95. Since BunchCount pairs individual and group queries for the same image and category, this reversal shows that category identity alone does not specify the intended counting granularity and establishes the counting unit as an essential conditioning factor. Directly mixing both units enables a single CountGD model to process both query types, but its group MAE increases from 1.38 under group-only training to 9.21, exposing semantic granularity conflict in joint training.

7 Conclusions

In this paper, we present GIC, a new counting task that requires counting either individuals or the semantic groups they form according to the specified counting unit. Furthermore, we construct BunchCount, a real-world benchmark with paired individual-group annotations and explicit containment relations, and propose GICount, a counting-unit guided relational framework. Systematic experiments establish GICount as a strong baseline for this task. This work moves visual counting beyond individual instances and provides a foundation for semantic unit-aware counting in practical scenarios such as retail inventory, agriculture, warehousing, and logistics.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods and assisting with translation. We have not used generative AI tools for other tasks with required disclosure, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for editing software code and editing the research paper to improve readability. We have reviewed all AI-assisted work. AI-assisted method implementations were verified and tested for correctness by 2 authors, and AI-assisted translations and textual edits were reviewed by the authors for accuracy. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

BunchCount is constructed from images collected from existing public datasets and publicly accessible web sources. The annotations were produced by the authors and members of their research laboratory. The dataset is intended for research use, and its release will respect the usage conditions of the corresponding image sources.

REFERENCES

Niki Amini-Naieni, Kiana Amini-Naieni, Tengda Han, and Andrew Zisserman. Open-world text-specifed object counting. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA, 2023. URL https://papers.bmvc2023.org/0510.pdf.

Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. Countgd: Multi-modal open-world counting. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 48810–48837. Curran Associates, Inc., 2024. doi: 10.52202/079017-1547. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/57c56985d9afe89bf78a8264c91071aa-Paper-Conference.pdf.

Anthropic. Claude Opus 5 System Card. https://www.anthropic.com/claude-opus-5-system-card, July 2026. Accessed: 2026-07-29.

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, 2025. URL https://arxiv.org/abs/2511.21631.

Yuanqiang Cai, Longyin Wen, Libo Zhang, Dawei Du, and Weiqiang Wang. Rethinking object detection in retail stores. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2):947–954, May 2021. doi: 10.1609/aaai.v35i2.16178. URL https://ojs.aaai.org/index.php/AAAI/article/view/16178.

Liu Chang, Zhong Yujie, Zisserman Andrew, and Xie Weidi. Countr: Transformer-based generalised visual counting. In *British Machine Vision Conference (BMVC)*, 2022.

Yuren Cong, Michael Ying Yang, and Bodo Rosenhahn. Reltr: Relation transformer for scene graph generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):11169–11183, 2023. doi: 10.1109/TPAMI.2023.3268066.

Siyang Dai, Jun Liu, and Ngai-Man Cheung. Referring expression counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16985–16995, June 2024.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 91–104, June 2025.

Kirill Gavrilyuk, Ryan Sanford, Mehrsan Javan, and Cees G. M. Snoek. Actor-transformers for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Zeeshan Hayder and Xuming He. Dsgg: Dense relation transformer for an end-to-end scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28317–28326, June 2024.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2026. URL https://arxiv.org/abs/2507.01006.

Meng-Ru Hsieh, Yen-Liang Lin, and Winston H. Hsu. Drone-based object counting by spatially regularized regional proposal network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Mostafa S. Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

Jinbae Im, JeongYeon Nam, Nokyung Park, Hyungmin Lee, and Seunghyun Park. Egtr: Extracting graph from transformer for scene graph generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24229–24238, June 2024.

Ruixiang Jiang, Lingbo Liu, and Changwen Chen. Clip-count: Towards text-guided zero-shot object counting. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pp. 4535–4545, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701085. doi: 10.1145/3581783.3611789. URL https://doi.org/10.1145/3581783.3611789.

Dongkeun Kim, Jinsung Lee, Minsu Cho, and Suha Kwak. Detector-free weakly supervised group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20083–20093, June 2022.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper_files/paper/2010/file/fe73f687e5bc5280214e0486b273a5f9-Paper.pdf.

Rongjie Li, Songyang Zhang, and Xuming He. Sgtr: End-to-end scene graph generation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19486–19496, June 2022.

Shuaicheng Li, Qianggang Cao, Lingbo Liu, Kunlin Yang, Shinan Liu, Jun Hou, and Shuai Yi. Groupformer: Group activity recognition with clustered spatial-temporal transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13668–13677, October 2021.

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 David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

Chang Liu, Haoning Wu, and Weidi Xie. Count anything at any granularity, 2026. URL https://arxiv.org/abs/2605.10887.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 38–55, Cham, 2025a. Springer Nature Switzerland. ISBN 978-3-031-72970-6.

Shuai Liu, Peng Zhang, Shiwei Zhang, and Wei Ke. Countse: Soft exemplar open-set object counting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21536–21546, October 2025b.

Anindya Mondal, Sauradip Nag, Xiatian Zhu, and Anjan Dutta. Omnicount: Multi-label object counting with semantic-geometric priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19537–19545, 2025.

Gia Khanh Nguyen, Yifeng Huang, and Minh Hoai. Can current ai models count what we mean, not what they see? a benchmark and systematic evaluation. In *2025 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, 2025. doi: 10.1109/DICTA68720.2025.11302471.

Thanh Nguyen, Chau Pham, Khoi Nguyen, and Minh Hoai. Few-shot object counting and detection. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 348–365, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20044-1.

OpenAI. GPT-5.6 System Card, July 2026. URL https://deploymentsafety.openai.com/gpt-5-6. Includes GPT-5.6 Sol, Terra, and Luna.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3170–3180, October 2023.

Jer Pelhan, Alan Lukeži?, Vitjan Zavrtanik, and Matej Kristan. Dave - a detect-and-verify paradigm for low-shot counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23293–23302, June 2024.

Rodrigo Pérez-Zavala, Miguel Torres-Torriti, Fernando Auat Cheein, and Giancarlo Troni. A pattern recognition strategy for visual grape bunch detection in vineyards. *Computers and Electronics in Agriculture*, 151:136–149, 2018. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2018.05.019. URL https://www.sciencedirect.com/science/article/pii/S0168169917310682.

Yifei Qian, Zhongliang Guo, Bowen Deng, Chun Tong Lei, Shuai Zhao, Chun Pong Lau, Xiaopeng Hong, and Michael P. Pound. T2icount: Enhancing cross-modal understanding for zero-shot counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25336–25345, June 2025.

Qwen Team. Qwen3.7-Max. Official model release, May 2026a. URL https://www.qwencloud.com/models/qwen3.7-max. Accessed: 2026-07-28.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026b. URL https://qwen.ai/blog?id=qwen3.5.

Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3394–3403, June 2021.

Suprosanna Shit, Rajat Koner, Bastian Wittmann, Johannes Paetzold, Ivan Ezhov, Hongwei Li, Jiazhen Pan, Sahand Sharifzadeh, Georgios Kaissis, Volker Tresp, and Bjoern Menze. Relationformer: A unified framework for image-to-graph generation. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 422–439, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19836-6.

Vishwanath A. Sindagi, Rajeev Yasarla, and Vishal M. Patel. Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting and localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43 (6):2141–2149, 2021. doi: 10.1109/TPAMI.2020.3013269.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025. URL https://arxiv.org/abs/2508.18265.

Zhicheng Wang, Liwen Xiao, Zhiguo Cao, and Hao Lu. Vision transformer off-the-shelf: A surprising baseline for few-shot class-agnostic counting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(6):5832–5840, Mar. 2024. doi: 10.1609/aaai.v38i6.28396. URL https://ojs.aaai.org/index.php/AAAI/article/view/28396.

Lu Zhang and Hayley Hung. Beyond f-formations: Determining social involvement in free standing conversing groups from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

A Additional Details of BunchCount

A.1 Semantic Groups and Annotation Scope

A group is primarily a semantic and perceptual unit rather than one determined by a single geometric rule. In common visual understanding, multiple objects of the same category can be perceived as a coherent and countable whole through cues such as regular arrangement, physical connection, fixed cardinality, spatial aggregation, or shared containment. In some cases, this perception also coincides with a part–whole organization, where repeated objects jointly form a familiar whole, such as beads forming a bracelet or lenses forming sunglasses. Although these groups differ substantially in appearance and organization, they share the same counting interpretation: the visible objects can be counted either individually or according to the higher-level groups they form.

To reduce ambiguity and ensure reliable annotations, BunchCount adopts a conservative image-selection protocol. We retain scenes in which the constituent objects are individually identifiable, the perceived groups have relatively clear boundaries, and group membership can be determined with reasonable consistency. Such boundaries may be supported by visible containers or physical connections, regular spatial organization, or clear separation between neighboring groups, for example, when the distance between groups is noticeably larger than the spacing among objects within each group. Cases with highly ambiguous group existence, boundaries, or constituent membership are excluded. For each retained image, we annotate individual objects with center points, semantic groups with bounding boxes, and explicitly associate each group with its constituent individuals, while allowing some individuals to remain ungrouped.

A.2 Taxonomy of Group Structures

To characterize the diverse ways in which individual objects form semantic groups, we organize the groups in BunchCount into four structure types according to their most representative organizing

characteristics: structural arrangement, physical connection, fixed cardinality, and spatial aggregation. A group may exhibit multiple perceptual cues, while its assigned type captures the cue that most directly supports its interpretation as a coherent and countable unit. This assignment provides each group with a single structural label and supports consistent dataset annotation and analysis.

Structural arrangement. This type describes groups formed through a regular or ordered spatial configuration among their constituent objects. The relative positions and ordering of the objects provide the primary cue for perceiving them as a coherent unit. Representative forms include stacks, rows, and columns, such as a stack of sugar cubes or a row of books. The same structure can appear across object categories with substantially different appearances, while preserving a recognizable organization.

Physical connection. This type describes groups whose constituent objects form a unified whole through visible attachment, linkage, or binding. The physical relation among the objects provides the primary cue for identifying the group and its members. Typical examples include grapes connected as a bunch, objects tied into a bundle, and beads linked into a string or bracelet. These groups often have flexible shapes, while their connections maintain a stable group identity.

Fixed cardinality. This type describes groups defined primarily by a conventional or functional number of constituent objects. The prescribed cardinality determines how the visible objects are combined into countable units across different spatial configurations. Representative examples include a pair of shoes and two lenses forming a pair of sunglasses. The objects may vary in position, orientation, and appearance, while their cardinal relation provides a consistent counting interpretation.

Spatial aggregation. This type describes groups formed through local spatial clustering or a shared container or supporting region. Spatial coherence provides the primary cue for associating the constituent objects with the same countable unit. Representative forms include piles and collections placed in boxes, bowls, bags, or plates, such as a pile of candies or a box of strawberries. Shared containment is included in this type because the containing region provides a clear spatial organization and group boundary.

A.3 Data Collection and Image Selection

We first identified object categories that commonly form countable groups. Candidate images were collected from FSC-147, Locount, COCO, OpenImages, and the web. We manually inspected images from the existing datasets. For web collection, we searched using either category names or combinations of group-structure terms and category names, such as "stack of plates," "pair of shoes," and "pile of stones." **The dataset and annotations will be made publicly available upon publication**.

We selected real-world images containing multiple objects of the same category and clearly distinguishable groups, allowing both individual- and group-level annotation. We excluded low-resolution images, images with severe occlusion or object density that prevented individual identification, and images in which group boundaries or member assignments were ambiguous. Synthetic and AI-generated images were also excluded.

A.4 Data Annotation

For each image, annotators first specified the individual-object category and the corresponding group description. All visible individual objects were annotated with center points, while each semantic group was annotated with a tight bounding box covering its visible constituent objects. Each individual point was then explicitly associated with at most one group, and individuals not belonging to any group were retained as ungrouped instances.

Groups without an explicit enclosing boundary were required to contain at least two visible individuals. A group with only one visible constituent was retained only when its identity was independently supported by a clear boundary, such as a shared container, or by a recognizable whole-object structure whose remaining constituents were occluded or outside the image. For fixed-cardinality groups, membership additionally followed the corresponding cardinal constraint. To support exemplar-based counting, three individual instances and one to three semantic groups were randomly selected in



Figure 6: Additional samples from BunchCount.

each image. Tight instance boxes were annotated for the selected individual exemplars, while the corresponding group annotations were used as group exemplars.

The annotations were produced by six annotators using our custom annotation platform and subsequently cross-checked for textual consistency, duplicated or missing point and box annotations, and incorrect group–individual associations. Samples whose group description, boundary, or membership remained ambiguous during cross-checking were removed rather than resolved through an arbitrary assignment. This process filtered out 58 candidate images, resulting in the final set of 1,330 images.

A.5 Dataset Split and Statistics

We split BunchCount by target category into mutually disjoint training, validation, and test sets. For categories overlapping with FSC-147, we follow its original split to maintain compatibility. The resulting splits contain 63, 9, and 14 categories with 869, 151, and 310 images, respectively.

Group-structure descriptions are allowed to recur across splits. Unlike object categories, which cover diverse appearances and semantics, semantic groups are formed through a limited set of recurring structures, including arrangement, physical connection, fixed cardinality, and spatial aggregation. This design evaluates whether group-structure semantics learned from training categories can transfer to unseen object categories. The average annotation counts and the numbers of samples for the four group-structure types in each split are summarized in Tables 4 and 5. Additional samples from BunchCount are shown in Figure 6.

Table 4: Split-wise statistics of BunchCount.

Split	Classes	Images	Ind. avg.	Grp. avg.
Train	63	869	58	6
Val	9	151	69	5
Test	14	310	90	16

Table 5: Distribution of group-structure types across the dataset splits.

Split	Images	Group Structure			
		Physical connection	Spatial aggregation	Fixed cardinality	Structural arrangement
Train	869	194	237	98	340
Val	151	47	42	6	56
Test	310	39	171	45	55

B Additional Implementation Details

B.1 Additional GICount Details

Architecture. GICount is built on CountGD and retains its detection backbone, RoI-based exemplar injection, and detection head. For counting-unit prompting, predefined prompt annotations separate the group descriptor from the object category. The BERT-encoded group descriptor is used as the semantic anchor; for holistic group names without an explicit descriptor, two learnable counting-unit embeddings are inserted before the feature enhancer. Relation sets are encoded by an independent two-layer Transformer encoder with a hidden dimension of 256, eight attention heads, an FFN dimension of 2048, GELU activation, 0.1 dropout, pre-layer normalization, and a final LayerNorm. Each set contains at most five individual queries preceded by a learnable `[REL]` token, whose output is used as the relation representation.

Training. GICount is initialized from the publicly released CountGD checkpoint, while the relation encoder, [REL] token, and learnable counting-unit embeddings are randomly initialized. Each image forms paired individual- and group-level samples, which are processed through two forward passes of the shared model. For group-level detection supervision, each annotated group box is represented by its center point, while the original box is retained as a visual exemplar. CountGD takes three exemplars per target; when fewer than three group exemplars are available, the last exemplar is repeated for padding. For relation supervision, queries from the final decoder layer of the individual-level pass are matched to the ground-truth points and organized using the annotated group–individual relations. We sample two positive sets from individuals within the same group and two negative sets from different groups, with at most five queries per set. As described in the main text, the relation loss combines a triplet loss and a pairwise positive-alignment loss. We search their relative weights over $\{0.1{:}0.9, 0.3{:}0.7, 0.5{:}0.5, 0.7{:}0.3, 0.9{:}0.1\}$ and set both weights to 0.5, corresponding to $\alpha = 0.5$. The triplet margin is set to 0.2. The individual- and group-level detection losses are also equally weighted. We freeze the Swin image encoder and BERT text encoder and optimize all remaining parameters using Adam with a learning rate of 1×10^{-4}, weight decay of 1×10^{-4}, and global gradient clipping of 0.1. The learning rate is reduced by a factor of ten every ten epochs. The maximum number of sampled label phrases is set to 40. The model is trained for 30 epochs with a batch size of 4 on one NVIDIA RTX 5090 GPU, taking approximately 10 hours. The relation module is disabled during inference, which follows the original CountGD localization and confidence-threshold counting procedure.

B.2 VLM Evaluation

We evaluate all vision-language models without finetuning on BunchCount. Each model receives an image and a text prompt specifying either the individual or group counting target. We disable

reasoning mode and set the temperature to zero when supported, with the maximum output length set to 32 tokens.

We designed three prompt formats—*direct*, *exclude*, and *detailed*—with progressively increasing instruction specificity, evaluated them using Qwen3-VL-8B, and selected the second format for all VLM evaluations. The individual- and group-level prompts are defined as follows:

Individual prompt: *Count the visible individual instances matching "{target_text}" in this image. Count individual items only, not groups or clusters of items. Return only one non-negative integer and no other text.*

Group prompt: *Count the visible groups matching "{target_text}" in this image. Count groups only, not the individual items inside the groups. Return only one non-negative integer and no other text.*

We extract a numerical count from each model response whenever possible. Responses from which no valid number can be parsed are treated as invalid and excluded from evaluation. Across all evaluated models, only LLaVA-NeXT-Mistral-7B produced such invalid responses, resulting in the exclusion of six samples.

B.3 Adaptation of Counting Methods

We use the official implementations and publicly released checkpoints for all counting baselines, and continue training on BunchCount following their official training recipes. Each image yields one individual-level sample and one group-level sample, which are randomly mixed with equal sampling probability. Individuals are represented by their annotated points, while group bounding boxes are converted to center points. For methods requiring three visual exemplars, available group exemplars are duplicated when necessary. All other settings follow the official implementations.

For GroundingREC, we preserve its paired referring-expression training instead of treating the two counting levels as independent samples. Each image is associated with an individual-level and a group-level attribute expression, and the original same-image contrastive objective is retained, such that the two counting-unit expressions serve as negatives to each other.

C Additional Evaluation Metrics

Following standard practice in visual counting, we evaluate counting accuracy using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE):

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}_i - y_i\right|, \ \mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2}, \tag{6}$$

where N is the number of evaluation samples, and $\hat{y}_i$ and y_i denote the predicted and ground-truth counts of the i-th sample, respectively. We compute these metrics separately for individual- and group-level samples.

For model selection, we use an average normalized MAE to balance performance across the two counting units. Specifically, we evaluate the candidate model weights on the validation set and normalize the MAE of each counting unit by the lowest corresponding MAE among all candidates:

$$\mathrm{NMAE}_{\mathrm{avg}} = \frac{1}{2}\left(\frac{\mathrm{MAE}_{\mathrm{ind}}}{\mathrm{MAE}_{\mathrm{ind}}^{\min}} + \frac{\mathrm{MAE}_{\mathrm{grp}}}{\mathrm{MAE}_{\mathrm{grp}}^{\min}}\right), \tag{7}$$

where $\mathrm{MAE}_{\mathrm{ind}}^{\min}$ and $\mathrm{MAE}_{\mathrm{grp}}^{\min}$ denote the lowest individual- and group-level MAEs among the candidate weights, respectively. We select the weight with the lowest $\mathrm{NMAE}_{\mathrm{avg}}$, while reporting the original MAE and RMSE for final evaluation.

D Computational Cost

Table 6 compares the model size and per-sample computation of representative counting methods. GICount contains 239.94M parameters, only 2.63M, or 1.1%, more than its CountGD backbone.

Table 6: Computational cost of representative counting methods under their reference input configurations. The two FLOPs values of GICount correspond to individual- and group-level queries, respectively.

Method	Parameters	Input	FLOPs per Forward
CounTR	99.69M	384^2	168.66G
CLIP-Count	166.09M	384^2	54.41G
GroundingREC	173.10M	800×1333	1.621T
CountGD	237.31M	800×1333	1.368T
GICount (Ours)	239.94M	800×1333	1.375T / 1.377T

Table 7: Group-level counting results across different group structures on the BunchCount test set. Each entry reports MAE / RMSE, and the number of samples in each subset is shown in parentheses.

Method	Physical Connection (39)	Spatial Aggregation (171)	Fixed Cardinality (45)	Structural Arrangement (55)
CountGD	1.64 / 4.11	0.64 / 1.98	58.22 / 74.05	1.11 / 1.93
GICount (Ours)	1.18 / 1.95	0.74 / 2.24	8.11 / 13.55	0.62 / 1.18

Table 8: Additional ablations on the components of relation supervision.

Method	Individual Test		Group Test	
	MAE↓	RMSE↓	MAE↓	RMSE↓
mean-pooling aggregator	16.02	44.13	2.55	8.62
group-query anchor	21.63	69.05	1.51	4.63
triplet loss only	20.37	64.33	1.63	4.49
positive-alignment loss only	18.80	57.10	1.85	5.41
full (Ours)	**12.20**	**28.89**	1.85	5.49

The group-individual relation module is used only during training and therefore introduces no additional relation-modeling branch at inference. Under an 800×1333 input, GICount requires 1.375 and 1.377 TFLOPs for individual- and group-level queries, respectively, indicating nearly identical inference costs across the two counting units. Training the full model for 30 epochs takes approximately 10 hours on a single NVIDIA RTX 5090 GPU.

E Results across Group Structures

We further break down the group-level results according to the four group structures in BunchCount, as reported in Table 7. CountGD and GICount produce similar errors on spatial aggregation, whereas GICount records lower errors on physical connection and structural arrangement. Fixed cardinality remains the most difficult subset for both models.

E.1 Additional Ablation Study

We further ablate the aggregation, anchor, and loss designs within relation supervision. All variants follow the same training and evaluation settings as the full model, with only the specified component modified. For relation aggregation, we replace the Transformer-based relation aggregator with mean pooling over the sampled individual-query features. For the relation anchor, we replace the group-structure representation extracted from the counting-unit prompt with the group-query representation. For the loss objective, we separately retain only the triplet term or only the pairwise positive-alignment term. Results are shown in Table 8

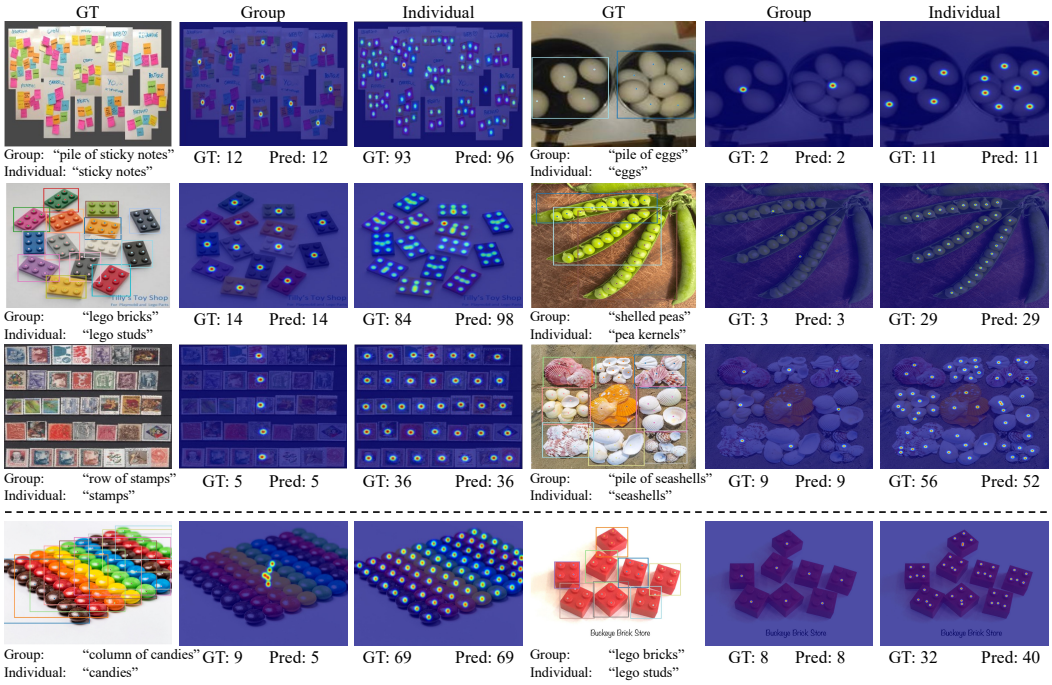


Figure 7: Additional qualitative results of GICount on BunchCount.

F Additional Qualitative Results

Figure 7 presents additional qualitative results of GICount on diverse object categories and group structures. For each example, the ground-truth group annotations are shown together with the predicted group- and individual-level response maps under the corresponding counting-unit prompts. The examples above the dashed line illustrate representative predictions, while those below show several failure cases.