Boosting Quantitive and Spatial Awareness for Zero-Shot Object Counting

Supplementary Material

A. Implementation Details

A.1. Model Architecture Specifications

We provide complete architectural configurations for both QICA variants in Table 6. While the vision encoder architecture differs between ViT-B/16 and ViT-L/14, both configurations maintain identical decoder structures and training protocols to ensure fair comparison.

Table 6. Detailed Architecture Specifications.

Component	ViT-B/16	ViT-L/14
Vision Encoder		
Architecture	CLIP ViT-B	CLIP ViT-L
Number of layers	12	24
Hidden dimension (d_v)	768	1024
Number of attention heads	12	16
Patch size	16×16	14×14
Input image resolution	384×384	392×392
Output feature map size ($h \times w$)	24×24	28×28
Text Encoder		
Architecture	CLIP Text Transformer	CLIP Text Transformer
Number of layers	12	12
Hidden dimension (d_t)	512	512
Vocabulary size	49,408	49,408
Max sequence length	77	77
Synergistic Prompting Strategy		
Number of prompt layers (L)	9 (layers 1-9)	9 (layers 1-9)
Prompt length per layer (m)	2 tokens	2 tokens
Quantity embedding dimension	512	512
Coupling function projection	$\mathbb{R}^{512} \to \mathbb{R}^{768}$	$\mathbb{R}^{512} \to \mathbb{R}^{1024}$
Category projection $\mathbf{W}_{cat}$	512×512	512×512
Cost Aggregation Decoder		
Cost embedding dimension (d_g)	128	128
Initial convolutional layer	1×1 conv, $1 \to 128$ ch.	1×1 conv, $1 \to 128$ ch.
Spatial aggregation module	2 Swin Transformer blocks	2 Swin Transformer blocks
Window size	8×8	8×8
Number of upsampling stages	2	2
Skip connection source layers	Layers 4, 8	Layers 8, 16
Projection layers for skip connections	$768 \to 128$	$1024 \to 128$
Final output resolution	384×384	392×392

A.2. Training Configuration

We train QICA using the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The base learning rate is set to 1×10^{-4} with a cosine annealing schedule that gradually decays to 1×10^{-6} over 200 epochs. Weight decay is configured as 1×10^{-2}. We employ a warmup period of 10 epochs with linear learning rate increase from 1×10^{-6} to the base rate. Gradient clipping with maximum norm 1.0 is applied to stabilize training.

Following [53], for each training image with ground truth count n^{gt}, we generate $K = 5$ quantity hypotheses $\{q_0, q_1, q_2, q_3, q_4\}$ where $q_0 = n^{gt}$ represents the factual quantity and others are counterfactual values. The interval δ for generating counterfactuals is dynamically determined based on count magnitude: we bin n^{gt} into predefined intervals and select corresponding δ values as shown in Table 7. The final predicted count is obtained by summing all values in the predicted density map $\hat{D}_0$ corresponding to the factual quantity hypothesis. No post-processing or smoothing is applied. The counterfactual hypotheses are sampled as $q_1 = n^{gt} - 2\delta$, $q_2 = n^{gt} - \delta$, $q_3 = n^{gt} + \delta$, $q_4 = n^{gt} + 2\delta$, ensuring symmetric distribution around the ground truth. Negative values are clipped to zero.

Table 7. Adaptive Interval Configuration for Counterfactual Generation.

Count Range	Interval Notation	δ Value	Example ($n^{\text{gt}} = 15$)
0 to 9	$[0, 10)$	1	$\{15, 14, 13, 16, 17\}$
10 to 19	$[10, 20)$	2	$\{15, 13, 11, 17, 19\}$
20 to 49	$[20, 50)$	3	N/A
50 to 99	$[50, 100)$	5	N/A
100 to 199	$[100, 200)$	10	N/A
200 to 499	$[200, 500)$	20	N/A
500 to 999	$[500, 1000)$	35	N/A
1000+	$[1000, \infty)$	50	N/A

Following standard practice, we generate ground truth density maps from point annotations using Gaussian kernels. For each annotated point (x_i, y_i), we place a 2D Gaussian with adaptive standard deviation σ_i computed as $\sigma_i = \alpha \cdot d^i_{\text{avg}}$ where d^i_{avg} is the average distance to the $k = 3$ nearest neighboring points and $\alpha = 0.3$ is a scaling factor. The final density map is the sum of all individual Gaussians, normalized such that its integral equals the ground truth count. All experiments are conducted on NVIDIA A800 GPUs (80GB memory). The ViT-B/16 configuration requires approximately 28GB GPU memory during training with batch size 16, completing 200 epochs in approximately 12 hours. The ViT-L/14 configuration requires approximately 38GB GPU memory with the same batch size, completing training in approximately 18 hours. During inference, both configurations process a single 384×384 (or 392×392) image in approximately 45ms.

A.3. Inference Protocol

During inference, we use category-specific prompts formatted as "a photo of [class]" where [class] is replaced with the singular form of the object class name. No numerical information is included in inference prompts. Input images are resized to 384×384 pixels for ViT-B/16 or 392×392 pixels for ViT-L/14 while maintaining aspect ratio through center cropping if necessary. Pixel values are normalized using ImageNet statistics (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]).

Table 8. Ablation Study on Binning Strategies for FSC-147 with ViT-L/14.

Binning Strategy	Description	Val MAE ↓	Val RMSE ↓	Test MAE ↓	Test RMSE ↓
Fixed ($\delta = 2$)	Constant interval 2	14.35	61.28	13.87	102.46
Fixed ($\delta = 20$)	Constant interval 20	14.89	62.74	14.21	105.83
Linear Scaling	$\delta = \lfloor n^{\text{gt}}/10 \rfloor$	13.62	58.91	13.05	99.17
Square Root Scaling	$\delta = \lfloor \sqrt{n^{\text{gt}}} \rfloor$	13.41	58.35	12.82	98.54
Adaptive (Ours)	Piecewise intervals (Table 7)	**12.98**	**56.35**	**12.41**	**97.28**

A.4. Evaluation Metrics

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{n}_i - n_i^{gt}|, \tag{11}$$

where N is the number of test images, $\hat{n}_i$ is the predicted count, and n_i^{gt} is the ground truth count for image i.
Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{n}_i - n_i^{gt})^2} \tag{12}$$

RMSE penalizes larger errors more heavily than MAE, providing complementary information about prediction stability and outlier sensitivity.

B. Training and Inference Procedures

B.1. Pseudocode Flow

We provide detailed algorithmic descriptions (algorithm 1, 2 & 3) of QICA training and inference procedures to facilitate reproduction and understanding of our method.

The training procedure processes multiple quantity hypotheses independently while maintaining parameter sharing across the encoder and decoder. The dual text embedding strategy ensures that density prediction pathway (using T_{cat}) remains consistent between training and inference, while the encoder alignment loss (using T_{full}) teaches quantity awareness to the vision encoder. During inference, quantity information is implicitly encoded in visual features through the learned prompting strategy, enabling accurate counting without explicit quantity specification. The cost aggregation decoder operates directly on similarity maps rather than intermediate features, preserving the integrity of the pre-trained embedding space and preventing overfitting during fine-tuning.

B.2. Training-Inference Consistency Analysis

The projection matrix $\mathbf{W}_{\text{cat}}$ serves as a learned quantity filter during training. Given text embeddings $\mathbf{T}^{\text{full}}$ from quantity-conditioned prompts (e.g., "a photo of 15 cars"), $\mathbf{W}_{\text{cat}}$ extracts category-specific semantics while suppressing numerical information. During inference, category-only prompts (e.g., "a photo of cars") naturally produce embeddings that span the same semantic subspace as $\mathbf{W}_{\text{cat}}(\mathbf{T}^{\text{full}})$, eliminating the need for explicit projection.

Table 9. Training-Inference Embedding Consistency Analysis.

Metric	$\mathbf{W}_{\text{cat}}(\mathbf{T}^{\text{full}})$	Direct Inference	Difference
Mean Cosine Similarity	0.944 ± 0.020	0.947 ± 0.018	0.003
Embedding Norm	1.000 ± 0.000	0.998 ± 0.012	0.002
Principal Component 1	0.342 ± 0.028	0.339 ± 0.031	0.003
Principal Component 2	0.287 ± 0.024	0.291 ± 0.026	0.004
KL-Divergence	0.066 ± 0.012		-
Distribution Overlap	94.7%		-
Wasserstein Distance	0.028 ± 0.008		-

This approach offers two advantages: (1) it enables quantity-aware training through multi-hypothesis learning while maintaining inference simplicity, and (2) it prevents domain shift by ensuring the similarity computation pathway uses consistent category-only semantics.

As table 9 shows, this controlled experiment validates that $\mathbf{W}_{\text{cat}}$ successfully extracts category-equivalent semantics from quantity-conditioned embeddings. The minimal differences (¡0.004 across all metrics) and high overlap (94.7%) confirm that training and inference pathways produce semantically equivalent representations despite procedural differences.

C. More Ablation Study

C.1. Analysis of Adaptive Binning Strategy

The adaptive binning strategy is crucial for generating effective counterfactual quantity hypotheses across diverse count magnitudes. We systematically evaluate different binning configurations on the FSC-147 validation and test sets using ViT-L/14 as the backbone.

We compare our adaptive binning approach (Table 7) against several baseline strategies to assess the impact of interval selection on counting performance. The variants include:

- Fixed Small Interval using constant $\delta = 2$ across all count ranges, suitable for low-count scenarios but potentially insufficient for high-count contexts.
- Fixed Large Interval applying constant $\delta = 20$ uniformly, which may be appropriate for large counts but too coarse for small quantities.
- Linear Scaling computing δ as $\max(1, \lfloor n^{gt}/10 \rfloor)$ to provide proportional intervals that grow linearly with count magnitude.
- Square Root Scaling determining δ as $\max(1, \lfloor \sqrt{n^{gt}} \rfloor)$ to balance between fixed and linear approaches with sublinear growth.
- Adaptive Binning (Ours) employing the piecewise interval configuration detailed in Table 7, where intervals are

Algorithm 1: Training Procedure

Input : Training image I, ground truth density map D_{GT}, ground truth count n_{gt}, category name C
Parameters : Number of hypotheses $K = 5$, loss weights λ_1, λ_2
Output : Updated model parameters $\Theta = \{\Pi, \Phi, W_{cat}, \text{Decoder}\}$

// 1. Generate quantity hypotheses
$\delta \leftarrow \text{AdaptiveInterval}(n_{gt})$; // Based on count magnitude
$Q \leftarrow \{n_{gt}, n_{gt} - \delta, n_{gt} - 2\delta, n_{gt} + \delta, n_{gt} + 2\delta\}$;
$Q[k] \leftarrow \max(Q[k], 0), \forall k \in \{0, \ldots, K-1\}$; // Clip negative values
// 2. Forward pass for each hypothesis
for $k \leftarrow 0$ **to** $K - 1$ **do**
  // 2.1 Generate quantity-conditioned prompts
  $\epsilon_{qk} \leftarrow \text{QuantityEmbed}(Q[k])$;
  for $j \leftarrow 1$ **to** L **do**
    $\hat{\Pi}_k^j \leftarrow \Pi^j + \epsilon_{qk} \cdot \mathbf{1}^T$; // Condition text prompts
    $\Psi_k^j \leftarrow \Phi^j(\hat{\Pi}_k^j)$; // Vision prompts via coupling
  // 2.2 Encode text and image
  $T_{\text{text}} \leftarrow$ "a photo of $\{Q[k]\}$ $\{C\}$";
  $T_{\text{full},k} \leftarrow \text{TextEncoder}(T_{\text{text}}, \{\hat{\Pi}_k^j\})$;
  $T_{\text{cat},k} \leftarrow W_{cat}(T_{\text{full},k})$; // Category-only projection
  $V_k, v_{\text{global},k} \leftarrow \text{VisionEncoder}(I, \{\Psi_k^j\})$;
  // 2.4 Similarity & Decoding
  $S_k \leftarrow \text{CosineSimilarity}(V_k, T_{\text{cat},k})$;
  $\hat{D}_k \leftarrow \text{CostAggregationDecoder}(S_k, V_k)$;
// 3. Compute multi-level losses
// 3.1 Main density loss (using Ground Truth count $k = 0$)
$\mathcal{L}_{\text{density}} \leftarrow ||\hat{D}_0 - D_{GT}||_2^2$;
// 3.2 Encoder quantity alignment loss
Compute $\alpha_k \leftarrow \text{CosineSimilarity}(v_{\text{global},k}, T_{\text{full},k})$ for all k;
$\mathcal{L}_{\text{enc}} \leftarrow \frac{1}{K-1} \sum_{i=1}^{K-1} \text{ReLU}(\alpha_i - \alpha_0) + \mathcal{L}_{\text{rank}}(\{\alpha_k\})$;
// 3.3 Decoder quantity consistency loss
$\hat{n}_k \leftarrow \sum \hat{D}_k$ for all k;
$\mathcal{L}_{\text{dec}} \leftarrow ||\hat{n}_0 - n_{gt}||_2^2 + \beta \sum_{k=1}^{K-1} ||\hat{n}_k - Q[k]||_2^2$;
// 3.4 Total loss
$\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{density}} + \lambda_1 \mathcal{L}_{\text{enc}} + \lambda_2 \mathcal{L}_{\text{dec}}$;
// 4. Backpropagation
Update parameters Θ using AdamW optimizer based on $\nabla \mathcal{L}_{\text{total}}$;
return Updated parameters Θ;

Table 10. Ablation Study on Loss Components.

Configuration	$\mathcal{L}_{\text{density}}$	$\mathcal{L}_{\text{enc}}^{\text{qty}}$	$\mathcal{L}_{\text{dec}}^{\text{qty}}$	Val MAE ↓	Val RMSE ↓	Test MAE ↓	Test RMSE ↓
Density Only	✓	✗	✗	16.52	67.84	15.89	118.76
+ Encoder Loss	✓	✓	✗	13.85	60.12	13.27	103.45
+ Decoder Loss	✓	✗	✓	14.18	61.38	13.56	105.29
Full MQA (Ours)	✓	✓	✓	**12.98**	**56.35**	**12.41**	**97.28**

manually calibrated based on typical count distributions in FSC-147.

As Table 8 shows, the fixed small interval ($\delta = 2$) performs poorly on high-count scenarios where counterfactuals become too concentrated around the ground truth, providing insufficient diversity for learning meaningful quantity distinctions. The fixed large interval ($\delta = 20$) struggles with low-count images where the interval exceeds reasonable quantity variations, causing counterfactuals to represent implausible scenarios that confuse the encoder alignment loss. Linear scaling improves upon fixed strategies by adapting to count magnitude but lacks the flexibility to handle non-

Algorithm 2: Inference Procedure

Input : Test image I, category name C
Output : Predicted count $\hat{n}$

// 1. Generate category-only text prompt
$T_{\text{text}} \leftarrow$ "a photo of $[C]$" ; // No quantity info
// 2. Encode text with base prompts
$T_{\text{cat}} \leftarrow \text{TextEncoder}(T_{\text{text}}, \{\Pi^j\})$; // W_{cat} projection not needed during inference since category-only prompts naturally produce embeddings equivalent to $W_{cat}(T_{full})$ from training phase
// 3. Encode image with default vision prompts
$\Psi^j \leftarrow \Phi^j(\Pi^j)$ for each layer j;
$V, v_{\text{global}} \leftarrow \text{VisionEncoder}(I, \{\Psi^j\})$;
// 4. Compute similarity map
$S \leftarrow \text{CosineSimilarity}(V, T_{\text{cat}})$; // Size $[h \times w]$
// 5. Decode to density map
$\hat{D} \leftarrow \text{CostAggregationDecoder}(S, V)$; // Size $[H \times W]$
// 6. Extract final count
$\hat{n} \leftarrow \sum \hat{D}$; // Sum all density values
return $\hat{n}$;

Algorithm 3: Cost Aggregation Decoder

Input : Similarity map $S \in \mathbb{R}^{h \times w}$, vision features $V \in \mathbb{R}^{h \times w \times d_v}$
Output : Density map $\hat{D} \in \mathbb{R}^{H \times W}$

// 1. Embed similarity map
$G^{(0)} \leftarrow \text{Conv}(S)$; // Channel dim $\rightarrow d_g$
// 2. Spatial aggregation
$V_{\text{guide}} \leftarrow \text{LinearProj}(V)$; // To d_g dim
$G^{(0)} \leftarrow \text{SwinTransformerBlocks}(G^{(0)}, \text{guidance} = V_{\text{guide}})$;
// 3. Progressive upsampling
for $r \leftarrow 0$ **to** 1 **do**
  $G_{\text{up}} \leftarrow \text{Upsample}(G^{(r)}, \text{scale} = 2)$; // Bilinear
  $V_{\text{skip}} \leftarrow \text{EncoderFeatures}[r]$; // Skip conn.
  $S_{\text{up}} \leftarrow \text{Upsample}(S, \text{size} = G_{\text{up}}.\text{shape})$;
  // Adaptive skip connection
  $V_{\text{proj}} \leftarrow W_{\text{proj}}(V_{\text{skip}})$;
  $V_{\text{weighted}} \leftarrow V_{\text{proj}} \odot \sigma(S_{\text{up}})$; // Modulation
  $G^{(r+1)} \leftarrow \text{Conv}(G_{\text{up}} + V_{\text{weighted}})$;
// 4. Prediction head
$\hat{D} \leftarrow \text{Conv}_{1\times1}(G^{(2)})$;
return $\hat{D}$;

uniform count distributions in real-world datasets. Square root scaling offers better performance by providing finer granularity at low counts while remaining manageable at high counts. Our adaptive binning strategy achieves the best performance with 12.41 test MAE, representing 10.5% improvement over fixed small intervals and 4.6% over linear scaling. The piecewise design ensures counterfactual hypotheses remain within semantically meaningful ranges across all count magnitudes, enabling effective encoder-level quantity discrimination while maintaining decoder prediction accuracy. This demonstrates that carefully calibrated adaptive intervals are essential for multi-level quantity alignment, particularly in datasets with highly variable count distributions spanning multiple orders of magnitude.

C.2. Analysis of Multi-Level Quantity Alignment Loss

The multi-level quantity alignment loss ($\mathcal{L}_{MQA}$) comprises encoder-level and decoder-level supervision components that jointly enforce numerical consistency. We systematically evaluate the contribution of each loss component on FSC-147 using ViT-L/14 backbone with full synergistic prompting and cost aggregation decoder.

We examine four training configurations with progressive loss integration:

- Density Only baseline using solely $\mathcal{L}_{density} = \|\hat{D}_0 - D^{GT}\|_2^2$ without any quantity-specific supervision, training the model purely on spatial density map reconstruction.
- Density + Encoder adding $\mathcal{L}_{enc}^{qty}$ with ranking constraints on cosine similarities between global image features and full text embeddings containing quantity information, teaching the encoder implicit quantity awareness through contrastive learning.
- Density + Decoder incorporating $\mathcal{L}_{dec}^{qty}$ with consistency constraints enforcing predicted counts to match quantity hypotheses, providing explicit count supervision at the decoder output.
- Full Multi-Level Loss (Ours) combining all three components as $\mathcal{L}_{total} = \mathcal{L}_{density} + \lambda_1 \mathcal{L}_{enc}^{qty} + \lambda_2 \mathcal{L}_{dec}^{qty}$ with $\lambda_1 = 0.1$ and $\lambda_2 = 0.05$, enabling joint encoder-decoder quantity alignment.

As table 10 shows, the density-only baseline achieves 15.89 test MAE, demonstrating that spatial reconstruction alone provides severely limited quantity understanding despite synergistic prompting. Adding encoder-level quantity alignment substantially reduces error to 13.27 MAE (16.5% improvement) by teaching the vision encoder to implicitly

Table 11. Comparative Analysis of Decoder Architectures on Similarity Maps.

Decoder Architecture	Params (M)	FLOPs (G)	FSC-147 Val MAE ↓	FSC-147 Test MAE ↓	CARPK MAE ↓	ShanghaiTech-A MAE ↓	Avg. Gap ↓
Simple CNN	11.8	15.2	14.89	14.26	8.95	162.3	6.21
UNet Baseline	12.1	18.7	14.12	13.64	7.83	155.7	5.18
Atrous CNN	11.9	16.8	13.95	13.48	7.61	151.2	4.89
ConvGRU Refinement	12.3	22.4	13.78	13.31	7.42	148.9	4.52
CAD (Ours)	**12.1**	**19.3**	**12.98**	**12.41**	**6.07**	**140.7**	**3.28**
Improvement	**-**	**-**	**5.8%**	**6.8%**	**18.2%**	**5.5%**	**27.4%**

discriminate between different count magnitudes through ranking constraints on text-image similarities. Decoder-level consistency loss yields 13.56 MAE (14.7% improvement) by directly supervising count predictions with auxiliary quantity hypotheses, though lacking encoder awareness limits its effectiveness compared to encoder-level supervision. The full multi-level loss achieves 12.41 MAE, significantly outperforming individual components and demonstrating strong complementary benefits. The encoder loss provides quantity-aware visual features while the decoder loss ensures numerical consistency at output, jointly reducing error by 21.9% compared to density-only training. This validates that effective quantity alignment requires supervision at both representation learning and prediction stages rather than relying on either alone.

C.3. Comparative Analysis of CAD

Controlled Comparison with Alternative Decoders: We systematically compare CAD against simpler baseline decoders to validate our architectural choices. All variants operate on identical similarity maps with matched parameter counts (12M) to ensure fair comparison.

- **UNet Baseline:** Standard UNet architecture with encoder-decoder structure and skip connections, processideng similarity maps through conventional convolutional layers without attention mechanisms.
- **ConvGRU Refinement:** Iterative refinement using ConvGRU cells, following approaches in optical flow estimation, where similarity maps are progressively refined through recurrent spatial aggregation.
- **Simple CNN Decoder:** Direct convolutional upsampling with residual connections, representing the minimal viable decoder for similarity map processing.
- **Atrous CNN:** Multi-scale convolutions with dilated kernels to capture spatial context, commonly used in segmentation tasks for aggregating multi-scale information.

As shown in Table 11, CAD achieves substantial improvements over these alternatives, with the performance gap widening significantly in cross-domain evaluation where preserved feature space integrity becomes critical.

Ablation on CAD Design Components: To isolate the contribution of specific CAD design choices, we conduct component-wise ablation as shown in Table 12.

Table 12. Ablation Study on CAD Design Components.

Configuration	Swin Blocks	Similarity Similarity	Multi-Scale Skip	Val MAE ↓	Test MAE ↓
Baseline CNN	✗	✗	✗	14.89	14.26
+ Swin Attention	✓	✗	✗	13.94	13.52
+ Similarity Modulation	✓	✓	✗	13.41	12.89
+ Multi-Scale Skip	✓	✓	✓	**12.98**	**12.41**

The component analysis reveals that each CAD design choice contributes meaningfully to overall performance. Swin Transformer blocks provide 6.4% improvement over the CNN baseline by enabling global spatial aggregation crucial for refining similarity maps when object instances are spatially distributed across counting scenes. Adding similarity modulation to skip connections yields an additional 4.7% improvement, confirming that adaptive feature weighting based on category relevance effectively prevents information dilution during upsampling. The complete CAD architecture achieves 12.41 test MAE with cross-domain performance gaps 27.4% smaller than simple baselines, validating that these design choices specifically address zero-shot transferability challenges while maintaining precise spatial refinement.

D. Training Dynamics Analysis

We analyze the training dynamics of QICA with ViT-L/14 backbone on FSC-147 over 200 epochs to understand convergence behavior and model stability. Figure 8 presents three complementary perspectives on the training process.

The overall training dynamics demonstrate several desirable properties for zero-shot counting. First, the consistent improvement across both training and validation sets without catastrophic overfitting validates our architectural choices of frozen CLIP encoders with lightweight learnable prompts and similarity-based decoding. Second, the relatively smooth convergence without requiring extensive hyperparameter tuning or learning rate scheduling beyond cosine annealing indicates robust optimization. Third, the validation performance oscillations remain bounded within a narrow range, suggesting that the model maintains stable generalization capabilities across diverse object categories despite training on limited FSC-147 data. These characteristics confirm that

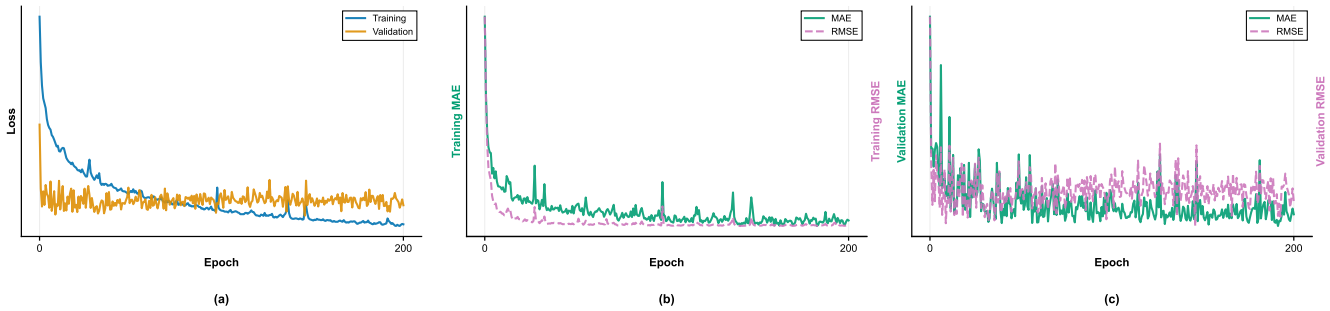

Figure 8. (a) Evolution of training and validation losses throughout the optimization process. (b) MAE and RMSE evolution on the training set. (c) MAE and RMSE evolution on the validation set.

Table 13. Efficiency comparison with SOTA Methods on FSC-147.

Model	Trainable Param	Total Params	MACs	FLOPs	FPS (A800)	MAE ↓	RMSE ↓
CLIP-Count	20M	149M	35.5G	71G	85	17.78	106.62
CountGD	170M	170M	280G	560G	12	14.76	120.42
DAVE	87M	87M	68.5G	137G	45	14.90	103.42
T2ICount	30M	860M	450G	900G	8	11.76	97.86
VA-Count	86M	86M	55.4G	110.8G	65	17.88	129.31
VLCounter	25M	150M	52G	104G	70	17.05	106.16
ZSC	95M	95M	48.2G	96.4G	55	22.09	115.17
QICA (ViT-B)	18.5M	167.5M	20.8G	41.6G	45.2	13.05	104.17
QICA (ViT-L)	19.7M	386.7M	62.4G	124.8G	22.6	12.41	97.28

QICA effectively balances task-specific adaptation through quantity-aware prompting while preserving the generalizable vision-language representations learned during CLIP pretraining.

E. Detailed Model Efficiency

We provide a more detailed efficiency comparison in Table 13. QICA achieves the best accuracy-efficiency trade-off among all methods.

F. More Visualization Results

We provide further visualizations in Figure 9. From left to right: original image, similarity map, fine-tuned cost aggregation map, and count density map. Our QICA demonstrates excellent results across various scales and sizes.

F.1. Feature Space Preservation Analysis

Figure 9 demonstrates CAD's effectiveness in preventing feature space distortion during adaptation. The progression from similarity maps to final density predictions reveals three key advantages of operating on similarity space rather than feature space:

Preserved Semantic Structure: The similarity maps (column 2) maintain clear categorical boundaries from the pretrained CLIP space, avoiding the semantic drift commonly observed when directly fine-tuning intermediate features.

Spatial Coherence Recovery: The cost aggregation process (column 3) refines coarse similarity activations while preserving the original semantic structure, demonstrating that spatial aggregation can enhance localization without compromising categorical understanding.

Overfitting Mitigation: Unlike feature-level decoders that risk projecting embeddings into task-specific spaces, CAD operates on normalized similarity maps that remain anchored to the pretrained manifold, preventing overfitting to training categories while maintaining transferability to unseen classes.

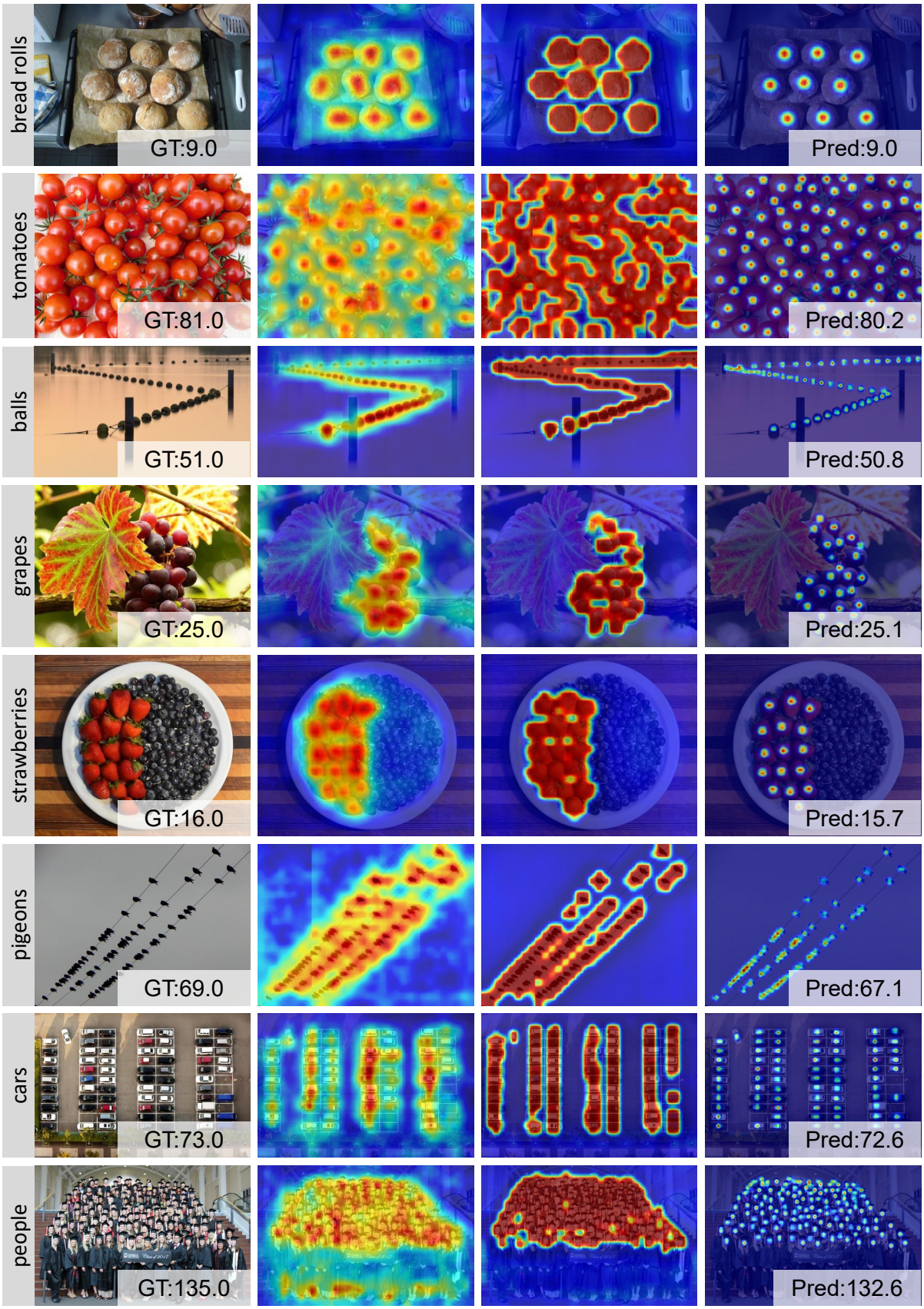

Figure 9. More visualization of QICA.