A Fixed-Point Approach to Unified Prompt-Based Counting

Wei Lin, Antoni B. Chan

Department of Computer Science, City University of Hong Kong
Tat Chee 83, Kowloon Tong, Hong Kong SAR, China
elonlin24@gmail.com, abchan@cityu.edu.hk

Overview

This supplementarial material is organized as follows:

1. Results on cross-prompt and multi-prompt adaptation is discussed to show the advantages of the unified prompt-based counting;
2. More details about the three components in the proposed prompt-based counting model are presented;
3. The pseudo-code is presented to show the whole computation process of fixed-point loss in contrastive training.
4. The detailed results in ablation studies.
5. Additional visualizations of the proposed method are presented to showcase its advantages and disadvantages.

Cross/Multi-Prompt Adaptation

Converting different types of prompts into the same representation, prompt masks, offers two additional advantages: (a) When new types of prompts are introduced, we can seamlessly transform them into prompts and directly apply them in our model without requiring training; (b) This approach enables us to combine multiple prompts for counting the object of interest.

To showcase the first advantage, we incorporate an additional prompt type, namely instance mask prompts, for prompt-based counting without requiring extra training. The instance mask is generated by converting the box annotations into instance masks using the SAM method (Kirillov et al. 2023). The results are presented in Table S1. The instance prompts yield comparable outcomes to box-guided training, as the prompts originate from box annotations. However, when using instance prompts, the estimation errors are slightly lower than the results obtained from box annotations, owing to the more precise representation provided by instance masks.

Regarding multiple prompts, we present the results in Table S2. Overall, the performance remains consistent with the results obtained using a single type of prompt. However, when the text prompt is combined with other prompts, such as box or point prompts, its performance improves. The original MAE and MSE of the text prompt are 16.81 and 105.83, respectively. These values decrease to 16.71 and 105.80 when the box prompt is added. Meanwhile, the MAE and MSE decrease to 16.78 and 105.79 when combined with point prompts.

prompts	train	validation		test	
		MAE	MSE	MAE	MSE
box		16.87	59.45	16.68	105.08
text	w/	16.92	58.92	16.81	105.83
point		17.16	59.38	15.86	103.27
instance	w/o	16.74	59.28	16.58	104.79

Table S1: Cross-prompt adaptation.

box	text	point	validation		test	
			MAE	MSE	MAE	MSE
✓	✓		16.89	58.91	16.71	105.80
✓		✓	16.76	59.36	16.50	104.79
	✓	✓	16.96	59.01	16.78	105.79
✓	✓	✓	16.73	58.88	16.64	105.75

Table S2: Counting with Mutiple types of prompts.

Model Structure

Our prompt-based counting model coonsists of three main components: an image backbone, the cross-attention module, and the density predictor, as illustrated in Figure 2 of the main paper.

Backbone & Image Feature. The backbone utilize the first four blocks of ResNet-101 (He et al. 2016) as the backbone, the image feature can be represented as:

$$\boldsymbol{F} = \mathcal{G}\left(\left[\boldsymbol{F}_{\frac{1}{8}}, \mathcal{U}_2\left(\boldsymbol{F}_{\frac{1}{16}}\right)\right], \Theta_{\mathcal{G}}\right), \tag{S1}$$

where $\mathcal{G}$ is a `conv_3×3` layer with parameters $\Theta_{\mathcal{G}}$, and $\mathcal{U}_2$ represents bilinear interpolation with a scale factor of 2. The inputs to $\mathcal{G}$, denoted as $\boldsymbol{F}_{\frac{1}{8}}$ and $\boldsymbol{F}_{\frac{1}{16}}$, are feature maps extracted from the last two blocks in the backbone. Their subscripts indicate the downsampling rate compared to the input. $\boldsymbol{F}$ has the same resolution with $\boldsymbol{F}_{\frac{1}{8}}$.

Cross Attention & Density Feature. The aim of cross attention is to let the prompt token $\boldsymbol{t} \in \mathbb{R}^C$ communicate with the flattened feature $\boldsymbol{F} \in \mathbb{R}^{N \times C}$, and then generate density feature accordingly. Specifically, we use $\mathcal{F}$ to generate *query*

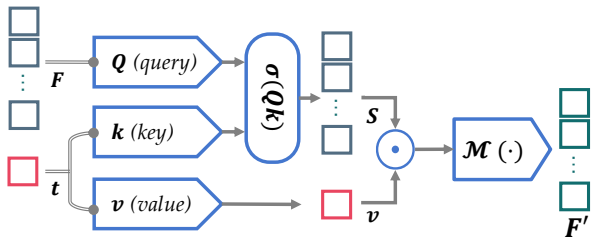


Figure S1: The pipline of cross-attention module.

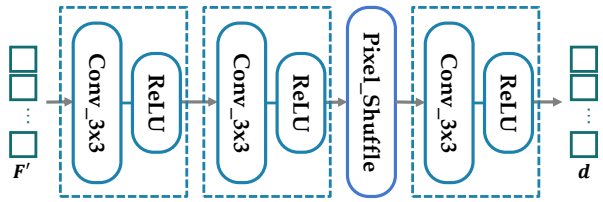


Figure S2: The pipline of density predictor.

embedding, and use $\boldsymbol{t}$ to generate *key* and *value* embeddings:

$$\boldsymbol{Q} = \boldsymbol{F}\boldsymbol{W}_q, \quad \boldsymbol{k} = \boldsymbol{W}_k\boldsymbol{t}, \quad \boldsymbol{v} = \boldsymbol{W}_v\boldsymbol{t} \tag{S2}$$

in which $\boldsymbol{W}_q \in \mathbb{R}^{C\times C'}$, $\boldsymbol{W}_k \in \mathbb{R}^{C'\times C}$, and $\boldsymbol{W}_v \in \mathbb{R}^{C'\times C}$ are learnable parapeters to perform cross attention. Prior attention models use `softmax` to compute the weights for each *value* embedding. However, there is only one token so that the result of `softmax` would wlways be one, contrast to the movetation of weight. Thus, we replace the `softmax` with Sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, resulting the following way to generate density features:

$$\boldsymbol{F}' = \mathcal{M}(\boldsymbol{S}^\top \boldsymbol{v}), \quad \boldsymbol{S} = \sigma(\boldsymbol{Q}\boldsymbol{k}) \in \mathbb{R}_+^N, \tag{S3}$$

where $\mathcal{M}$ is a linear projection to transform the attention results into density features, denoted as $\boldsymbol{F}'$. The visualized pipline is displayed in Figure S1.

Density Predictor. We utilize the identical density predictor released in SPDCN (Lin et al. 2022), benefiting from its straightforward yet effective performance. As illustrated in Figure S2, the density predictor is composed of three combined layers of `conv_3×3` and `ReLU`. Before reaching the final convolutional layer, a `pixel_shuffle` layer with scale factor of 8 is employed to upscale the density features, aligning them with the resolution of the input image.

Pseudo Code of Contrastive Training

Algorithm S1 summarizes the procedure for computing the fixed-point loss under the contrastive training scheme. At the beginning, $\boldsymbol{d}_i^{(0)}$ and $\boldsymbol{d}_j^{(0)}$ are initilized with $\boldsymbol{m}_i$ and $= \boldsymbol{0}_{\boldsymbol{m}_j}$ respectively. Subsequently, the fixed-point estimation is performed iteratively.

In the t-th iteration, we first compute the prompt token $\boldsymbol{t}^{(t+1)}$ based on both the positive and negative prompt masks, $\boldsymbol{d}_i^{(t)}$ and $\boldsymbol{d}_j^{(t)}$. Following this, the positive and negative density map are updated via the function $\mathcal{D}$ with parameters θ,

Algorithm S1: Contrastive Training

Input: Postive sample $(\boldsymbol{F}_i, \boldsymbol{m}_i, \boldsymbol{d}_i')$, negative sample $(\boldsymbol{F}_j, \boldsymbol{m}_j, \boldsymbol{d}_j')$, and the number of iteration T.

Output: Contrastive fixed-point loss $\hat{\mathcal{L}}_{ij}$.

Initialize $\boldsymbol{d}_i^{(0)} \leftarrow \boldsymbol{m}_i$ and $\boldsymbol{d}_j^{(0)} \leftarrow \boldsymbol{0}_{\boldsymbol{m}_j}$.
for $t \leftarrow 0$ to $T-1$ **do**
  $\{$*Detach* $\boldsymbol{d}_i^{(T-1)}$ *and* $\boldsymbol{d}_j^{(T-1)}$ *from the computation graph according to (17) in the main paper.*$\}$
  if $t = T-1$ **then**
    $\boldsymbol{d}_i^{(t)} \leftarrow \texttt{DETACH}\left(\boldsymbol{d}_i^{(t)}\right)$.
    $\boldsymbol{d}_j^{(t)} \leftarrow \texttt{DETACH}\left(\boldsymbol{d}_j^{(t)}\right)$.
  end if
  Comprehensive token $\boldsymbol{t}^{(t+1)} \leftarrow \frac{\boldsymbol{F}_i^\top \boldsymbol{d}_i^{(t)} + \boldsymbol{F}_j^\top \boldsymbol{d}_j^{(t)}}{\|\boldsymbol{d}_i^{(t)}\|_1 + \|\boldsymbol{d}_j^{(t)}\|_1}$.
  The positive density map $\boldsymbol{d}_i^{(t+1)} \leftarrow \mathcal{D}_\theta(\boldsymbol{F}_i,\ \boldsymbol{t}^{(t+1)})$.
  The negative density map $\boldsymbol{d}_j^{(t+1)} \leftarrow \mathcal{D}_\theta(\boldsymbol{F}_j,\ \boldsymbol{t}^{(t+1)})$.
end for
The token at infinity $\boldsymbol{t}^{(\infty)} \leftarrow \frac{\boldsymbol{F}_i^\top \boldsymbol{d}_i'}{\|\boldsymbol{d}_i'\|_1}$.
The positive density at infinity $\boldsymbol{d}_i^{(\infty)} \leftarrow \mathcal{D}_\theta(\boldsymbol{F}_i, \boldsymbol{t}^{(\infty)})$.
The negative density at infinity $\boldsymbol{d}_j^{(\infty)} \leftarrow \mathcal{D}_\theta(\boldsymbol{F}_j, \boldsymbol{t}^{(\infty)})$.
The positive fixed-point loss $\hat{\mathcal{L}}_i \leftarrow \left\|\boldsymbol{d}_i^{(\infty)} - \boldsymbol{d}_i'\right\| + \left\|\boldsymbol{d}_i^{(T)} - \texttt{DETACH}\left(\boldsymbol{d}_i^{(\infty)}\right)\right\|^2$.
The negative fixed-point loss $\hat{\mathcal{L}}_j \leftarrow \left\|\boldsymbol{d}_j^{(\infty)}\right\| + \left\|\boldsymbol{d}_j^{(T)} - \texttt{DETACH}\left(\boldsymbol{d}_j^{(\infty)}\right)\right\|^2$
return $\hat{\mathcal{L}}_{ij} \leftarrow \hat{\mathcal{L}}_i + \hat{\mathcal{L}}_j$.

as described in (9) of the main paper. Here $\mathcal{D}$ comprises the cross-attention module in Figure S1 and the density predictor in Figure S2. Note that both density maps are detached from the computation graph before the final iteration.

To compute the fixed-point loss, we also predict the corresponding density map with the token at infinity, $\boldsymbol{t}^{(\infty)}$. The final loss function is computed for both the positive and negative samples. The distinction lies in that the prediction for the positive component should align with the ground truth, while the prediction for the negative component should approach an all-zero density map. It is worth noting that $\boldsymbol{d}^{(\infty)}$ is detached in the finite loss calculation to prevent degenerate solutions where both $\boldsymbol{d}^{(T)}$ and $\boldsymbol{d}^{(\infty)}$ converge towards zero (Chen and He 2021).

Detailed Results in Ablation Studies

In the main paper, we only present the average MAE/MSE across three types of prompts to conserve space. Here, we provide the detailed corresponding data. Table S3 compares the methods for generating text prompt masks; Table S4 presents the results while comparing fixed-point loss with L2 loss; Table S5 discusses the enhancements achieved by different components in fixed-points; and Table S6 provides

Mask	box (val)		text (val)		point (val)	
	MAE	MSE	MAE	MSE	MAE	MSE
cosine	18.55	69.32	20.59	79.30	18.46	69.70
softmax	16.87	59.45	16.92	58.92	17.16	59.38
Mask	box (test)		text (test)		point (test)	
	MAE	MSE	MAE	MSE	MAE	MSE
cosine	16.97	108.13	17.84	108.77	16.12	105.82
softmax	16.68	105.08	16.81	105.83	15.86	103.27

Table S3: Comparison of text prompt mask generation.

loss function	prompt	validation		test	
		MAE	MSE	MAE	MSE
L2	box	18.84	66.96	17.26	107.62
	text	18.37	65.49	16.83	105.83
	point	20.07	69.52	17.55	106.03
fixed-point	box	16.87	59.45	16.68	105.08
	text	16.92	58.92	16.81	105.83
	point	17.16	59.38	15.86	103.27

Table S4: Comparison between MSE and fixed-point loss.

the N-MAE and N-MSE values for different prompts.

More Visualization

In this section, we also present more visualizations of the proposed method. In Figure S3, we provide some counting examples with text prompts. It demonstrates that our model can generate accurate density maps even when the given image contains multiple types of objects. For convenience, we use $\boldsymbol{d}^{(t)}$ to denote the prediction in the t-th iteration. The visualization shows that $\boldsymbol{d}^{(1)}$ consistently predicts a lower count than $\boldsymbol{d}^{(2)}$. When compared with $\boldsymbol{d}^{(1)}$, the count of $\boldsymbol{d}^{(2)}$ is closer to the ground truth.

In Figure S4, we illustrate some failure cases. In the first row of Figure S4, the minimum repetitive unit is a single *lens*. However, the object of interest, *sunglasses*, naturally has two lenses, doubling the repetitive unit. This discrepancy causes the model to predict a count that is twice the ground truth. In the second row of Figure S4, two types of pills with competely different appearances and shapes are presented. The model only counts one of them, while the other one is recognized as irrelevant object.

Figure S5 showcases two examples of counting with different prompts. These diverse prompts also exhibit varying levels of performance, but the estimation errors remain consistent across the entire dataset.

Prompt	infinity	finity	validation set		test set	
			MAE	MSE	MAE	MSE
box	$\mathcal{L}_\infty$	–	27.08	72.56	30.06	111.42
	–	$\mathcal{L}_{T'}$	20.67	68.34	17.29	107.67
	$\mathcal{L}_\infty$	$\mathcal{L}_{T'}$	16.87	61.86	16.46	106.99
	$\mathcal{L}_\infty$	$\mathcal{L}_T$	16.87	59.45	16.68	105.08
text	$\mathcal{L}_\infty$	–	27.93	74.35	30.82	110.88
	–	$\mathcal{L}_{T'}$	21.07	70.71	17.31	106.90
	$\mathcal{L}_\infty$	$\mathcal{L}_{T'}$	16.99	61.47	16.71	107.32
	$\mathcal{L}_\infty$	$\mathcal{L}_T$	16.92	58.92	16.81	105.83
point	$\mathcal{L}_\infty$	–	19.73	68.76	20.03	108.32
	–	$\mathcal{L}_{T'}$	20.64	67.13	17.53	107.67
	$\mathcal{L}_\infty$	$\mathcal{L}_{T'}$	17.54	63.10	15.69	103.42
	$\mathcal{L}_\infty$	$\mathcal{L}_T$	17.16	59.38	15.88	103.27

Table S5: Ablation study on different combinations of infinity and finity part in fixed-point loss.

contrastive training	prompt	validation set		test set	
		N-MAE	N-MSE	N-MAE	N-MSE
w/o	box	38.27	73.37	50.91	88.28
	text	37.89	70.08	49.85	88.34
	point	38.12	75.81	50.68	85.67
w/	box	1.90	17.57	3.85	14.84
	text	1.83	17.29	3.90	15.03
	point	1.97	16.62	3.92	15.70

Table S6: Ablation study on contrastive training.

References

Chen, X.; and He, K. 2021. Exploring Simple Siamese Representation Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 15750–15758.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; Dollár, P.; and Girshick, R. 2023. Segment Anything. In *International Conference on Computer Vision (ICCV)*.

Lin, W.; Yang, K.; Ma, X.; Gao, J.; Liu, L.; Liu, S.; Hou, J.; Yi, S.; and Chan, A. B. 2022. Scale-Prior Deformable Convolution for Exemplar-Guided Class-Agnostic Counting. In *British Machine Vision Conference (BMVC)*, 313.

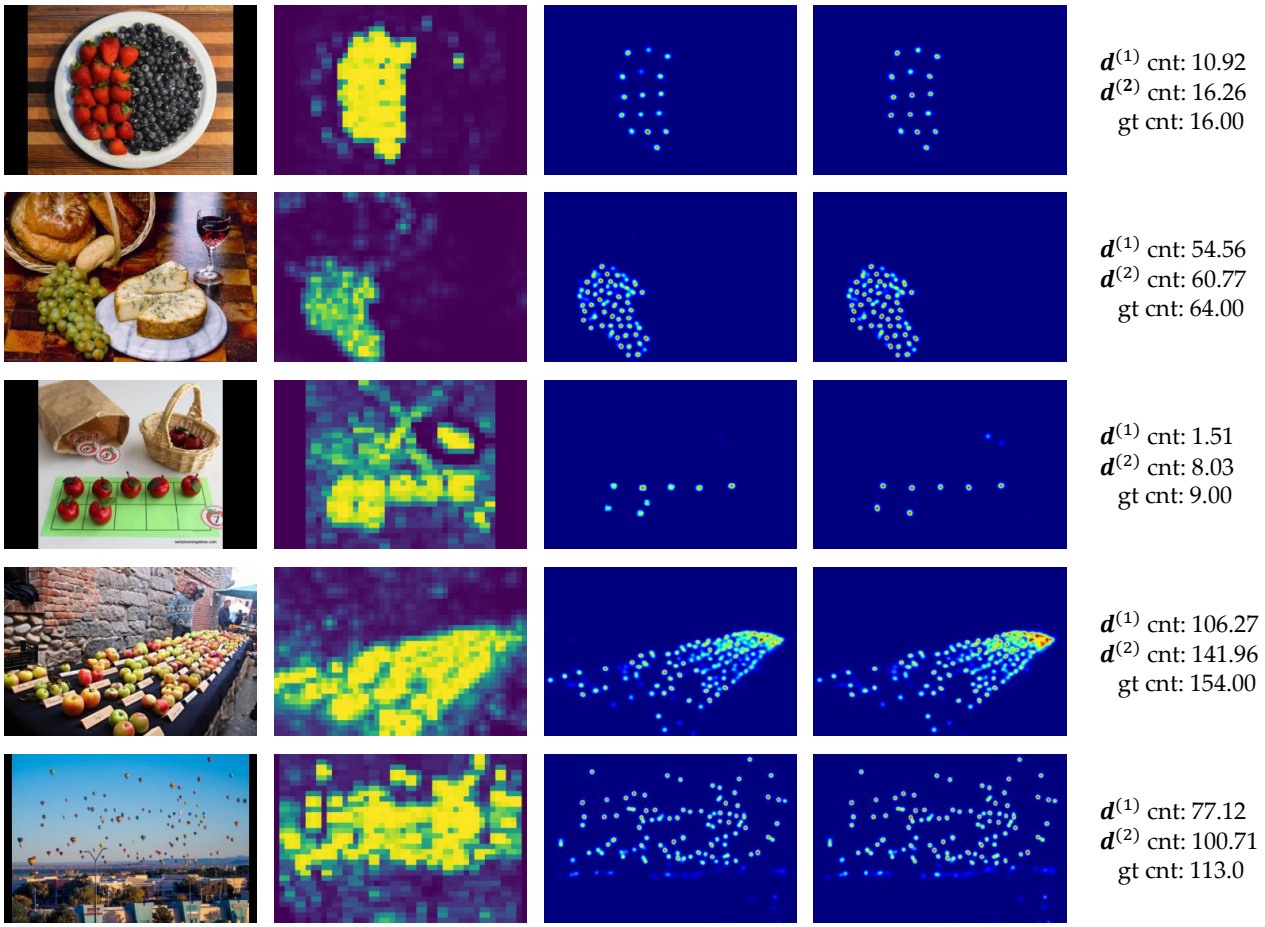


Figure S3: Examples of counting with text prompts.

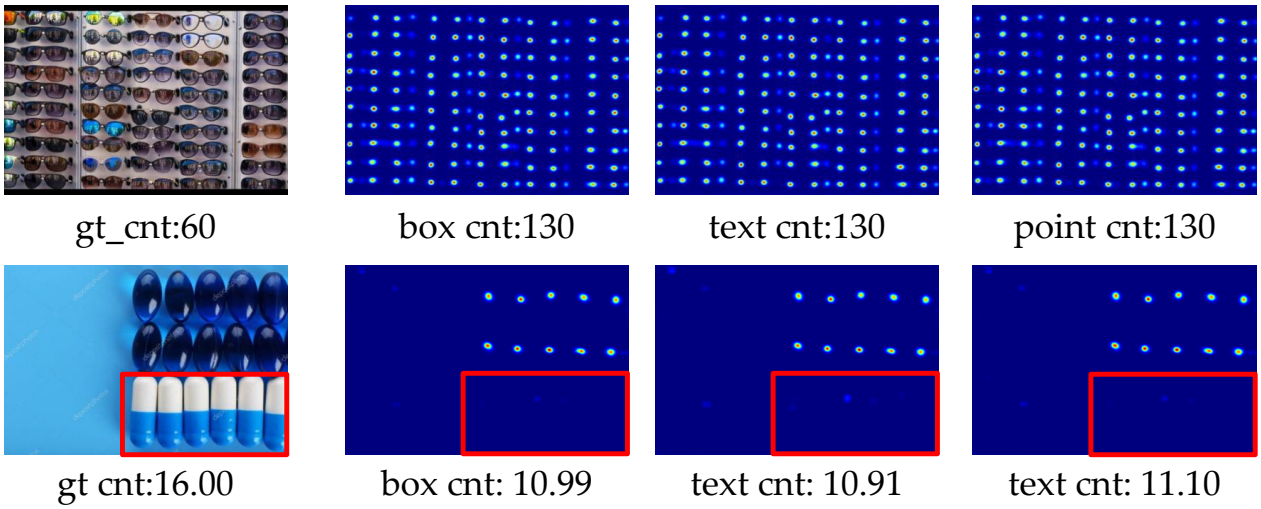
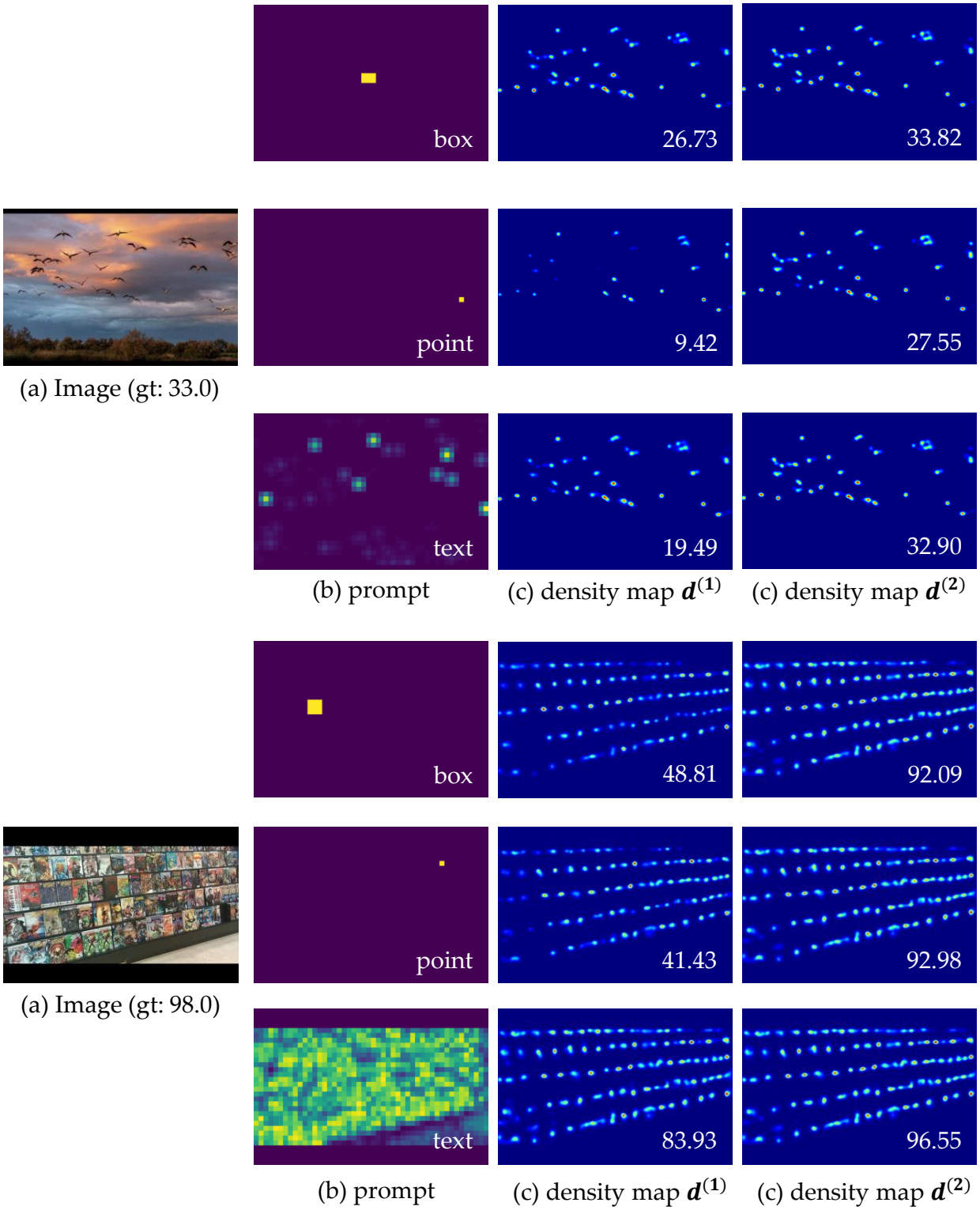


Figure S4: Typical failure cases.



(a) Image (gt: 33.0)

(b) prompt (c) density map $\boldsymbol{d}^{(1)}$ (c) density map $\boldsymbol{d}^{(2)}$

(a) Image (gt: 98.0)

(b) prompt (c) density map $\boldsymbol{d}^{(1)}$ (c) density map $\boldsymbol{d}^{(2)}$

Figure S5: Examples of counting with different types of prompts.