

AFreeCA: Annotation-Free Counting for All

Adriano D’Alessandro[✉], Ali Mahdavi-Amiri[✉], and Ghassan Hamarneh[✉]

Simon Fraser University, Burnaby, Canada
{acdaless, amahdavi, hamarneh}@sfu.ca

Abstract. Object counting methods typically rely on manually annotated datasets. The cost of creating such datasets has restricted the versatility of these networks to count objects from specific classes (such as humans or penguins), and counting objects from diverse categories remains a challenge. The availability of robust text-to-image latent diffusion models (LDMs) raises the question of whether these models can be utilized to generate counting datasets. However, LDMs struggle to create images with an exact number of objects based solely on text prompts but they can be used to offer a dependable *sorting* signal by adding and removing objects within an image. Leveraging this data, we initially introduce an unsupervised sorting methodology to learn object-related features that are subsequently refined and anchored for counting purposes using counting data generated by LDMs. Further, we present a density classifier-guided method for dividing an image into patches containing objects that can be reliably counted. Consequently, we can generate counting data for any type of object and count them in an unsupervised manner. Our approach outperforms unsupervised and few-shot alternatives and is not restricted to specific object classes for which counting data is available. Code available at: github.com/adrian-dalessandro/AFreeCA.

Keywords: Object Counting · Synthetic Data · Feature Learning

1 Introduction

Object counting is an important task in computer vision, with diverse applications such as crowd analysis [12, 27, 32], wildlife monitoring [1], and traffic analysis [10]. These tasks, historically reliant on fully-supervised methods, demand detailed annotations that are labor-intensive to collect. For instance, the annotation of 5,109 images for a crowd counting dataset necessitated 3,000 hours, averaging 35 minutes per image [30]. Although semi-supervised approaches have sought to reduce this burden by requiring annotations for only a subset of the data, they still demand hundreds of hours of labor. In response, research into unsupervised counting methods has aimed to eliminate the annotation burden entirely. Despite these efforts, such methods remain challenging to develop and the existing methods are limited to crowd counting [2, 18]. In this paper, using text-to-image latent diffusion models (LDMs), we propose an *unsupervised* counting method that can be applied to a *wide variety* of objects as in Fig. 1.

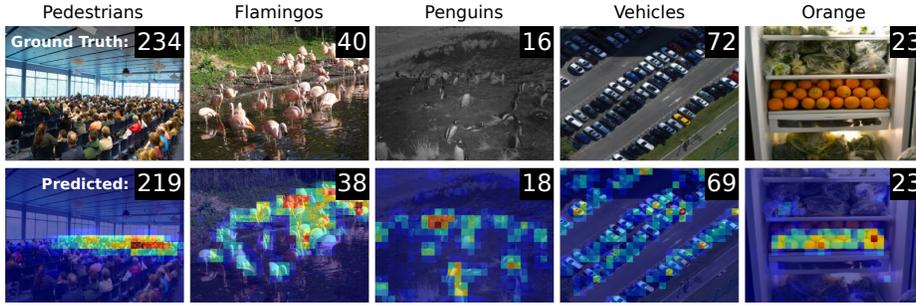


Fig. 1: We propose a method which exploits synthetic counting data generated by Stable Diffusion. With this, we establish an *annotator-free* method that produces accurate count maps without location-based supervision for a wide range of object categories.

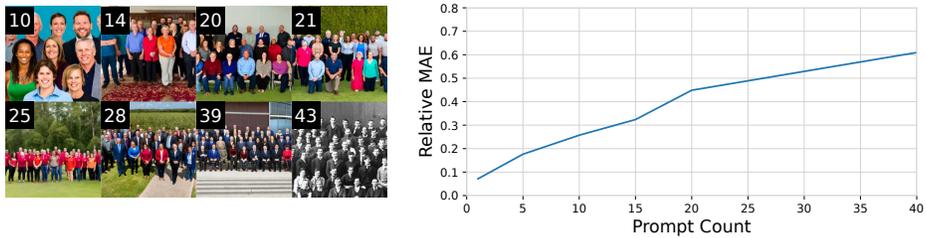


Fig. 2: *Left:* when given a prompt count of 20, Stable Diffusion outputs images with a similar but often incorrect object count. *Right:* as the prompt count increases, the relative error between the true underlying count and the prompt count increases.

Recently, advancements in text-to-image LDMs, such as Stable Diffusion [24], have presented novel opportunities to address various challenges. These methods excel at producing synthetic images with a high degree of realism that accurately reflect the contents of text-based prompts. This capability enables the generation of labeled synthetic images automatically by integrating label information into the prompts. Recent works have highlighted the significant potential of this strategy for zero-shot recognition, few-shot recognition, and model pre-training [9, 26]. However, these strategies have only focused on object recognition, raising the question of whether they can be applied to object counting.

A straightforward approach for extending the use of LDMs to object counting might involve prompting the network to produce images using explicit quantity labels, such as “An image with 20 people” with “20” used as the prompt count label as seen in Fig. 2 (Left). However, as previous work has also indicated, the text encoders used in LDMs have a limited understanding of quantity [20]. To demonstrate this, we annotated 40 synthetic samples per-category across different crowd count categories and analyzed the discrepancy between prompted and actual counts, termed label noise. Fig. 2 (Right) shows that the relative discrepancy increases with the magnitude of the prompt count label, suggesting that synthetic data is unreliable for images with high counts. To learn robust count-



Fig. 3: Workflow. Our framework uses simple prompts to create synthetic data for training a sorting model, a density classifier, and a count anchoring network. These elements are combined into a model which can accurately count diverse object categories even within dense images by subdividing them into smaller, more manageable areas.

related features given this label noise we train a sorting network on a dataset comprised of image triplets, ordered by their known count ranking. These triplets are generated by synthetically adding to or removing objects from real images using LDMs, enabling the network to capture high-quality counting features. While these features do not directly correspond to specific count numbers, we anchor them using synthetic counting data. To accomplish this, we fine-tune only a linear layer on-top of the sorting network, ensuring the stability of the learned features [14, 28]. Further, given the diminished reliability of the synthetic data for high object counts, we adopt an approach at inference-time to divide dense regions into smaller patches. This partitioning strategy not only aligns the patch with the more reliable counting range of our network but it also helps in preserving high-frequency details by sourcing higher-resolution patches directly from the original images. Our method’s workflow is detailed in Fig. 3. In summary, our contributions are listed in the following.

- To the best of our knowledge, we propose the *first unsupervised* counting method that can be used for a *variety of object categories*.
- We demonstrate that latent diffusion models are able to produce valuable counting and sorting data across diverse object categories. We introduce a

method to leverage this data, initially focusing on learning to sort followed by a fine tuning strategy to learn counting.

- To maximize the utilization of image resolution and improve performance on highly dense images, we introduce a method that dynamically partitions dense images by leveraging guidance from a density classification network.
- We surpass SOTA unsupervised and zero-shot crowd counting methods on several crowd counting benchmark datasets. We also provide various ablation studies to justify our design.

2 Related Work

Fully-Supervised Counting. Object counting has advanced through density map learning, using Gaussian kernels convolved with dot map annotations [16,17,32]. There are a wide range of diverse approaches to leveraging these annotations. GauNet [4] employs locally connected Gaussian kernels for density map generation, introducing low-rank approximation with translation invariance. GLoss [29] reframes density map estimation as an unbalanced optimal transport problem, proposing a specialized loss function. And, AMSNet [11] uses neural architecture search and a multi-scale loss to incorporate structural information.

Unsupervised Counting. In our work, we define unsupervised counting as any method that eliminates the need for new annotations, thus reducing the annotation burden for object counting problems. This definition includes methods that utilize foundation models. CSS-CCNN [2] introduces an unsupervised approach using self-supervised pre-training with image rotation and Sinkhorn matching, requiring prior knowledge of maximum crowd count and power law distribution parameters. CrowdCLIP [18] leverages features from CLIP, a pre-trained language-image model, through multi-modal ranking with text prompts. They use a filtering strategy to select crowd-containing patches and match features to crowd count categories represented in text prompts.

Generative Models. Stable Diffusion (SD) [24] is a generative model for image synthesis based on latent diffusion models. SD generates images through a multi-step process. Initially, a real image is encoded via a variational autoencoder, yielding a compressed representation in a lower-dimensional latent space. Gaussian noise is then iteratively applied to these features. To restore meaningful features, SD employs the reverse diffusion process, an iterative denoising mechanism. Finally, the denoised features are decoded to synthesize a new image. SD can be conditioned on text embeddings, typically from a text-encoder like CLIP [22], enabling image generation guided by user-defined text prompts.

3 Methodology

To count objects in images from an *arbitrary* category, we propose using latent diffusion models (LDMs) to synthesize data. However, LDMs often misinterpret

quantity in prompts, leading to discrepancies between the intended and actual object counts in generated images, known as *label noise*. This issue worsens with higher prompt counts, highlighting two key problems: the existence of a count threshold beyond which generated data becomes unreliable due to excessive label noise, and the diminished reliability of learned features for dense regions, increasing the risk of the model overfitting to noise. Our methodology addresses these problems by combining several strategies explained in the following.

First, we generate highly reliable synthetic sorting data by adding and removing objects from an image using Stable Diffusion (SD) [24] (Fig. 3). This process produces triplets of images ranked based on object count that can be used to learn robust counting features through training a *sorting network* [6, 18, 19]. Following this, we use SD to generate *noisy* synthetic counting data. This data is then used to fine-tune a counting network built upon our sorting network. By using the count data as anchors within the sorting features, we can preserve the reliability of the object quantity features obtained by the sorting network while establishing a correspondence between the features and the count.

As the data generated by SD tends to be more accurate for smaller counts, there could be a decline in performance when dealing with images containing a large number of objects. We extend the model’s reach to a wider range of object counts by recognizing that any image can be partitioned into sub-regions with fewer objects. However, over-partitioning sparse regions can introduce noise due to boundary artifacts, double counting, etc. Thus, we need a method to identify dense regions within an image. As a result, we generate synthetic crowd density classification data and then fine-tune a density guidance network on top of our pre-trained sorting model to identify dense image regions. At the inference stage, this density guidance is instrumental in identifying image sections that might exceed the counting model’s reliable counting range. These identified regions are then processed at a higher resolution (when available) to ensure accurate counting. By retrieving higher-resolution patches from the original image, we address issues related to feature loss caused by resizing images to a fixed size before feeding them into a network. Therefore, dividing an image into patches leads to more precise counting estimations.

3.1 Learning to Sort

Generating Synthetic Sorting Data We generate a dataset of images sorted based on their object counts by modifying reference images to either add or remove objects, using a LDM for image-to-image synthesis or out-painting. Unlike previous approaches that focus on intra-image ranking for counting [19], our strategy can add objects and also remove objects while preserving the scene’s perspective and original features. This broadens the diversity and range of object counts in the resulting image triplets. The process begins with a base dataset of real or generated reference images, represented as $\mathcal{D}^{\text{ref}} = \{x_i^{\text{ref}}\}^{N_{\text{ref}}}$, where x_i^{ref} denotes an image with an unknown number of objects c_i^{ref} , and N_{ref} is the total number of such images. To synthesize images with fewer objects, we apply

Stable Diffusion (SD):

$$x_i^{\text{syn-}} = SD(x_i^{\text{ref}}, t_p, t_n) \quad (1)$$

where $x_i^{\text{syn-}}$ is the resultant image with fewer objects, guided by a text prompt t_p and, to prevent the addition of targeted objects, a negative prompt t_n . As an example, we set t_p to “an empty place” and t_n to “pedestrians, humans, people, crowds” for crowd counting problems. Due to the stochastic nature of the image generation process, $x_i^{\text{syn-}}$ contains an unknown number of objects c_i^{syn} . In the supplementary material, we empirically verify that the relationship, $c_i^{\text{ref}} \geq c_i^{\text{syn}}$, holds in 99% of cases.

For adding objects, we engage in a similar process using outpainting:

$$x_i^{\text{syn+}} = SD(x_i^{\text{ref}}, \mathcal{M}, t_p, t_n) \quad (2)$$

where \mathcal{M} is a mask indicating where to outpaint and, thus, add new objects and is set to a thick band around the perimeter of the image x_i^{ref} , which is 1/3rd of the image size. This approach effectively increases the object count and scene density. We transform each image x_i^{ref} into four augmented versions with increased object counts $\{x_{ij}^{\text{syn+}}\}_{j=1}^4$ and four with decreased counts $\{x_{ik}^{\text{syn-}}\}_{k=1}^4$, from which the 16 ordered triplets $\{x_{ij}^{\text{syn-}} \geq x_i^{\text{ref}} \geq x_{ik}^{\text{syn+}}\}_{j,k=1}^4$ are generated.

Pre-Training For a given triplet of images $X = \{x^{\text{syn-}}, x^{\text{ref}}, x^{\text{syn+}}\}$ with rank labels $Y = \{0, 1, 2\}$, we generate a similarity matrix $[S^y]_{i,j} = -|y_i - y_j|$. We encode X using network f_θ to produce feature vectors $Z = \{z^{\text{syn-}}, z^{\text{ref}}, z^{\text{syn+}}\}$; $z_i \in \mathbb{R}^{2048}$. We further utilize a sorting head v_Θ to produce a continuous valued output \hat{y}_i from z_i as an estimate of the rank of x_i . We then calculate a second similarity matrix $[S^z]_{i,j}$ using the cosine similarity between z_i and z_j . S^z and S^y encode the relational structure between examples and are used to align the feature space and label space with the ground truth ordering by minimizing the following losses (this is akin to RankSim [7] with the key difference that we use count sorting labels instead of ground truth values like they do):

$$\ell_{\text{sort}}^y = \sum_i^3 (\mathbf{rk}(S_i^y) - \mathbf{rk}(S_i^{\hat{y}}))^2 \quad \ell_{\text{sort}}^z = \sum_i^3 (\mathbf{rk}(S_i^y) - \mathbf{rk}(S_i^z))^2, \quad (3)$$

where $S_i^{\hat{y}}$ is the predicted similarity matrix, S_i^* is i -th row of S^* , and \mathbf{rk} is a non-differentiable ranking function. The total sorting loss is then defined as:

$$\mathcal{L}_{\text{sort}} = \ell_{\text{sort}}^y + \lambda \ell_{\text{sort}}^z, \quad (4)$$

where λ is set to 5.0. To optimize \mathbf{rk} , we follow the same strategy as in RankSim [7] and apply a blackbox combinatorial solver [21].

3.2 Learning to Count from Synthetic Data

Our approach generates synthetic images with approximate object counts using SD for text-to-image synthesis:

$$\{x_i^s, c_i^p\} = SD(t_p, t_n), \quad (5)$$

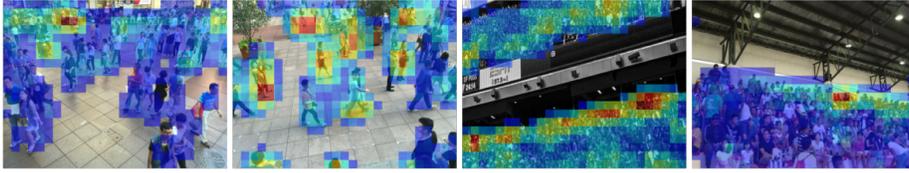


Fig. 4: Sorting Features. We calculate the channel-wise mean of the features produced by the sorting network to demonstrate where the network is active. The network appears to focus on the object of interest across a wide range of crowd densities.

where c_i^p is the object count from the prompt, and x_i^s are the generated images. This yields a dataset of synthetic examples $\mathcal{D}_{\text{cnt}}^s = \{x_i^s, c_i^p\}^{N_{\text{cnt}}}$. We use simple prompts to ensure accurate image generation, and to avoid complex prompt engineering. For crowd counting, prompts are straightforward, e.g., “20 people.” Our experiments involve generating images for a wide range of counts from 1 to 1000, which we elaborate on in the supplementary material. We generate 150 images for each prompt category, plus 800 images with zero objects generated by random sampling scene category names from the Places365 [33] dataset for prompts and explicitly excluding objects using negative prompts. For training our counting network with the synthetic dataset $\mathcal{D}_{\text{cnt}}^s$ and the pre-trained sorting network f_θ , we adopt a strategy to preserve the integrity of f_θ features. Previous works have indicated that fine-tuning an entire network distorts the pre-trained features of that network and degrades performance on out-of-distribution data [14, 28]. In Fig. 4, we observe that the feature maps produced by f_θ focus on the image regions where objects exist, which is a property that we would like to preserve. Given that we are training our counting network using synthetic data, and we are evaluating on real data, to avoid the potential for any feature distortion we only fine-tune a linear layer atop the pre-trained sorting network to anchor its features to actual count values.

To further ensure that our counting network is as accurate as possible, we automatically filter out outliers from the synthetic dataset using a feature space analysis approach, which is effective for mitigating the impacts of label noise [15, 31, 34]. We adopt a simplified version of CleanNet [15], which calculates class prototypes as well as the similarity between samples and prototypes to detect noisy samples. We compute features z_i for each image x_i^s using f_θ and then create a prototype vector z_μ^c for each prompt count category c :

$$z_\mu^c = \frac{1}{N} \sum_{i=1}^N z_i^c. \quad (6)$$

We then filter out synthetic images whose features align closer with a different category’s prototype than their own. Finally, we train a counting network, g_Φ

optimizing a mean squared error:

$$\mathcal{L}_{count} = \frac{1}{N_{cnt}} \sum_{(x_i^s, c_i^p) \in \mathcal{D}_{cnt}^s} (g_\phi \circ f_\theta(x_i^s) - c_i^p)^2. \quad (7)$$

3.3 Crowd Density Classification

Given that our counting networks is more reliable for smaller counts, we partition dense images into patches with lower counts to enhance estimation accuracy in each patch. This partitioning process is directed by a classifier that discerns between dense and sparse images. Given that dense image regions may be poorly represented by our counting model, a dataset is necessary for training such a classifier. We use stable diffusion to produce a classification dataset of synthetic images with various crowd densities $\mathcal{D}_{dense}^s = \{x_i^s, y_i^{den}\}^{N_{den}}$, where x_i^s is the synthetic image, y_i^{den} is the density category recorded in the prompt, and N_{den} is the size of the dataset. As highlighted in Fig. 3, we categorize densities into three groups: no crowd, sparse crowd, and dense crowd, embedding these categories directly into the generation prompts to create representative images for each. We re-purpose the zero-object examples from the earlier noisy count dataset for our "no crowd" category. We then use this synthetic dataset to train our classifier by fine-tuning a density classification layer, h_ϕ , on top of pre-trained sorting network. The loss function for this step is:

$$\mathcal{L}_{dense} = \frac{1}{N_{den}} \sum_{(x_i^s, y_i^{den}) \in \mathcal{D}_{dense}^s} \ell_{CE}(h_\phi \circ f_\theta(x_i^s), y_i^{den}), \quad (8)$$

where ℓ_{CE} is the cross-entropy loss.

3.4 Density Classifier Guided Partitioning (DCGP)

We previously argued that by partitioning images into sub-region patches, each patch will have fewer objects and be more likely to fall within the accurate range of our counting model. However, images often contain groups of objects at a variety of sizes and crowd densities due to the perspective of the scene. Given this, we propose a technique called density classifier guided partitioning (DCGP), which treats image patches differently based on their estimated density.

For each image, we produce a count map to approximate the number of objects within the respective region (see Fig. 5, bottom). In cases where a region exhibits sparsity according to our density classifier, like the top-right quadrant of the count map depicted in Fig. 5, the original count maps can be directly employed to estimate the count within that region. Conversely, if a count map corresponds to a dense region, the spatial patch associated with that region, potentially at a higher resolution, is extracted from the input image. This patch is then forwarded to our counting network to obtain a more precise estimate, considering the smaller number of objects within the patch, enabling the counting network to perform more effectively in estimation. This hybrid resolution counting estimation is eventually combined to find the final count.

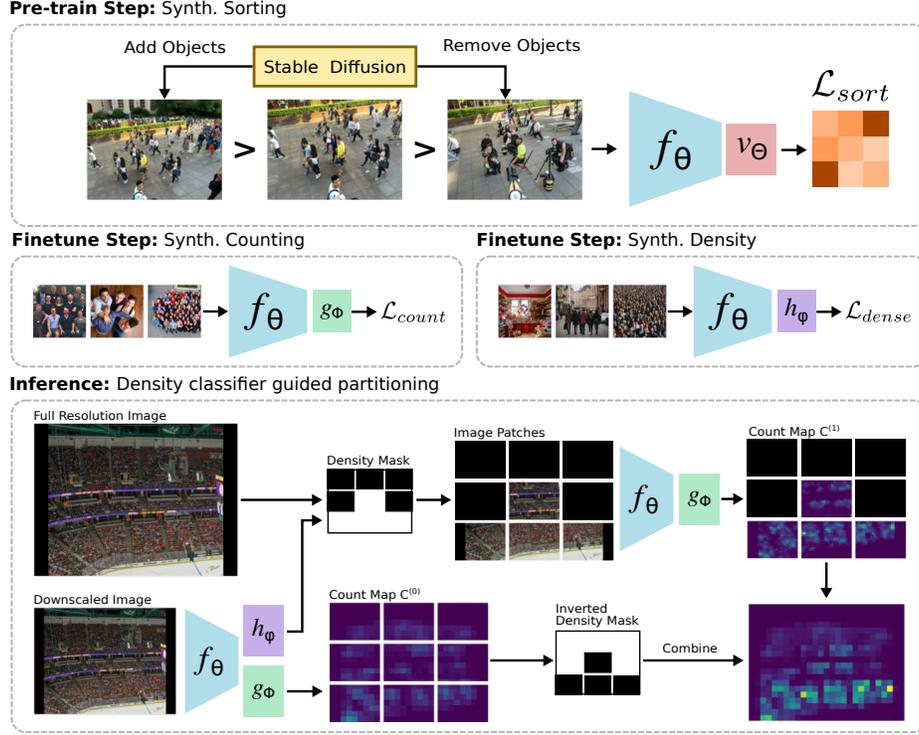


Fig. 5: Methodology. Our strategy involves three distinct steps supported by a synthetic training signal extracted from stable diffusion. The pre-training step sorts synthetic and real images to learn high quality object quantity features from the source distribution. The finetuning step utilizes the pre-trained features and synthetic data to train a counting head and a density classification head. Finally, the density classifier guides inference by partitioning dense images so that there are fewer objects per image.

Formally, to obtain a count map we adapt our pre-trained sorting network f_θ , omitting the global average pooling layer. This modified network produces feature maps $z \in \mathbb{R}^{H \times W \times 2048}$, where H is the height of the feature map and W is the width. We calculate the regional counts for each feature map element as follows:

$$c_{ij}^{(0)} = g_\phi(z_{ij}/(H \cdot W)), \quad (9)$$

where $c_{ij}^{(0)}$ denotes the count for the (i, j) -th element, providing a comprehensive count map for the image. We further classify each region's density using f_θ and a density classifier h_ϕ to generate a density map D , where D_{ij} represents the density class for the (i, j) -th region in z . This allows us to estimate which regions in the whole image are dense and require further processing.

Finally, we partition a high resolution version of a given image into a 3×3 grid of image patches. For sparse patches, we directly sum counts from $c^{(0)}$. For

Table 1: Crowd Counting Performance. We evaluate the performance of our method on the test set of crowd counting benchmark datasets. It is evident that our method outperforms other unsupervised and zero-shot techniques.

Method	Type	SHB		JHU		SHA		QNRf	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
ADSCNet [3]	Full supervised	6.4	11.3	-	-	55.4	97.7	71.3	132.5
GLoss [29]	Full supervised	7.3	11.7	59.9	259.5	61.3	95.4	84.3	147.5
GauNet [5]	Full supervised	6.2	9.9	58.2	245.1	54.8	89.1	81.6	153.7
CLIP-Count [13]	Zero-Shot	45.7	77.4	-	-	192.6	308.4	-	-
CSS-CCNN++ [2]	Unsupervised	-	-	197.9	611.9	195.6	293.2	414.0	652.1
CrowdCLIP [18]	Unsupervised	69.3	85.8	213.7	576.1	146.1	236.3	283.3	488.7
Ours	Unsupervised	35.0	50.7	173.8	519.4	152.7	219.0	283.1	453.2

Table 2: Object Counting Performance. Our method outperforms few-shot and zero-shot methods on the CARPK dataset.

Method	Type	CARPK	
		MAE	MSE
FamNet [23]	Few-shot	28.84	44.47
BMNet+ [25]	Few-shot	10.44	13.77
CLIP-Count [13]	Zero-shot	11.96	16.61
Ours	Unsupervised	9.35	12.29

dense patches, we process them through the counting network for a precise patch count estimate $c^{(1)}$, summing these to achieve the image’s final count.

4 Experiments & Results

4.1 Performance

Datasets. We benchmark on 4 crowd counting datasets: ShanghaiTechA (SHA) [32], ShanghaiTechB (SHB) [32], UCF-QNRf [12], and JHU-Crowd++ [27] with average crowd counts of 501, 123, 815, and 346 respectively. We further benchmark on a vehicle counting dataset [10] and a penguin counting dataset [1].

Main Results. Our results in Tab. 1 and Tab. 2 showcase our method’s performance on both crowd and vehicle counting tasks, evaluated using mean absolute error (MAE) and mean squared error (MSE). We compare our approach to unsupervised, few-shot, and zero-shot methods, where authors have made those results available. Few-shot and zero-shot methods aim to create a general counting network for any category using a large, manually annotated dataset with a diverse set of categories. Few-shot methods rely on an exemplar, sampled from the target image, to define the category of interest, whereas zero-shot methods use a text prompt. Unlike these methods, which depend on extensive annotated datasets but produce a general counting network, our unsupervised approach



Fig. 6: Qualitative Crowd Counting. We compare the count maps produced by our model to the ground truth count maps. The count maps are annotated with the ground truth counts and predicted counts. Our method accurately localizes crowds across a range of crowd densities, without any location-based supervision.

Table 3: Partitioning Strategy. We explore the impact of the image partitioning rate for naive partitioning and density classifier guidance.

Strategy	Rate	SHB		JHU		SHA		Q NRF	
		MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Fixed	1x1	42.1	68.1	203.6	649.6	196.6	321.4	392.5	707.1
Fixed	2x2	51.6	62.1	202.4	531.0	165.7	258.2	314.5	484.8
Fixed	3x3	47.1	57.1	219.5	501.8	186.9	278.8	425.6	559.9
Simple GP	1x1→2x2	42.1	68.1	181.7	541.9	163.1	255.9	299.9	487.7
Simple GP	1x1→3x3	42.1	68.1	173.9	495.3	155.7	225.4	309.5	480.2
DCGP	1x1→2x2	38.5	61.0	182.8	562.9	156.9	230.8	324.0	541.7
DCGP	1x1→3x3	35.0	50.7	173.8	519.4	152.7	219.0	283.1	453.2

Table 4: Pre-training Ablation Study. We explore the impact of different pre-training strategies on the final performance of the network

Pre-training	SHB		JHU		SHA		Q NRF	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
ImageNet	68.1	104.7	304.9	654.9	316.2	410.0	423.1	655.9
Intra-Image Rank	52.8	76.3	256.4	610.4	186.6	291.2	380.0	694.9
ℓ_{sort}^y only	38.2	58.0	183.9	551.1	188.4	293.5	359.3	579.1
$\ell_{sort}^y + \ell_{sort}^z$ (Ours)	35.0	50.7	173.8	519.4	152.7	219.0	283.1	453.2

learns to count objects for a single category without any manual annotations. Despite this difference, we include these methods in our comparison for two reasons: First, they share the common goal of removing or reducing the annotation burden. Second, there are no unsupervised counting methods which have provided evaluation on object categories beyond crowds.

With respect to crowd counting, we find that our method outperforms CLIP-Count [13] on all available datasets. It also outperforms both CrowdCLIP [18]

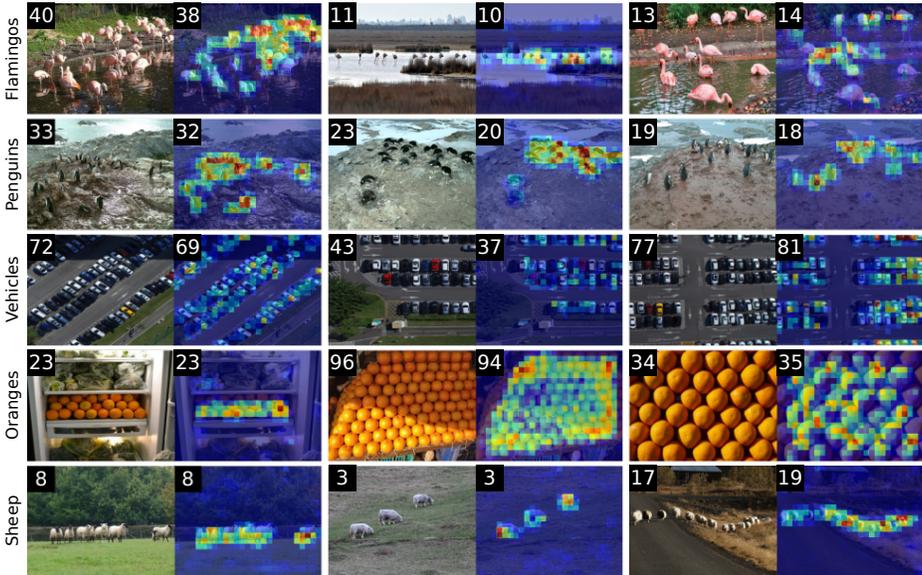


Fig. 7: Qualitative Object Counting. We evaluate our method across a diverse range of object categories and demonstrate its proficiency in accurately predicting object counts and also generating precise count maps for each evaluated category.

Table 5: Counting Network Ablation Study. We explore the performance of different strategies when training the counting network with synthetic counting data.

Strategy	SHB		JHU		SHA		QNRf	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
\mathcal{L}_{count} only train	122.9	154.6	318.5	785.7	408.8	531.8	711.2	1027.6
Full Network Finetune	43.9	74.6	292.4	749.0	340.8	462.5	664.8	965.2
Last Layer Finetune	35.0	50.7	173.8	519.4	157.7	219.0	283.1	453.2

and CSS-CCNN [2] on every dataset, with the sole exception of the SHA dataset in terms of MAE. Notably, our technique significantly improves on the JHU and SHB datasets, indicating that it particularly excels in scenarios with fewer objects. With respect to object counting, there are no unsupervised counting methods which have explored objects beyond pedestrians in crowds. However, we find that our unsupervised approach surpasses existing zero-shot and few-shot methods, as demonstrated on the CARPK dataset. Additionally, we assess our method on the penguins dataset [1], to establish a benchmark for future researchers, where we report a MAE of 5.1, and a MSE of 8.0.

In summary, our method not only outperforms existing unsupervised approaches in crowd counting but also outperforms zero-shot methods across both crowd and vehicle counting tasks. This suggests that our method provides superior performance for arbitrary categories.

Table 6: High Resolution Partitioning. We explore the performance implications of selecting partitions from a high-resolution image compared to a low resolution image.

Strategy	SHB		JHU		SHA		Q NRF	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Low Resolution	35.6	55.0	180.2	563.8	154.4	223.8	317.6	552.8
Full Resolution	35.0	50.7	173.8	519.4	152.7	219.0	283.1	453.2

Qualitative Results. We highlight qualitative results for crowd counting in Fig. 6 by comparing count maps estimated by our method to count maps calculated from ground truth dot maps. We demonstrate that our approach produces accurate count maps that localize crowds across a wide range of counts without *any* location-based supervision. We further extend our evaluation to arbitrary object categories, as seen in Fig. 7. For penguins and vehicles, we utilize sorting data generated from training set images of the penguins [1] and the CARPK [10] datasets, respectively, assessing performance on their test sets. For flamingos, oranges, and sheep, we create synthetic images with simple prompts (e.g., “*Many sheep. Photograph.*”), combined with random scene names from the Places365 dataset [33]. We evaluate quality on a mix of images from the FSC147 dataset [23] and manually annotated public domain images, confirming our method’s robustness in accurately localizing diverse object categories.

4.2 Ablation Study

Impact of Partition Strategy. In Tab. 3, we examine the effect of partitioning strategies on network performance. Initially, we explore a straightforward approach of dividing all images into an $M \times M$ grid at different rates of M , which we refer to as fixed partitioning. We then evaluate a simple guided partitioning strategy, where we use the density classifier to provide a single density estimate for the whole image. If the whole image is classified as dense, we use $M \times M$ partitioning, and 1×1 otherwise. Finally, we evaluate DCGP using different partitioning rates M , where we start with a 1×1 rate and partition with an $M \times M$ rate for image regions determined to be dense by h_ϕ . Notably, DCGP provides a significant performance boost compared to naively applying a 1×1 rate or $M \times M$ rate, indicating that our method is successfully addressing the challenges introduced by the synthetic label noise. Further, we determine that the optimal partitioning rate is 1×1 to 3×3 , which we apply to all datasets.

Impact of Pre-training. Our approach introduces a pre-training strategy aimed at developing high-quality features for object quantity estimation, serving as a foundation for the subsequent training of our counting network. In Tab. 4, we explore different pre-training strategies to determine the impact of the feature learning strategy on the final performance of the network. We compare our method with networks pre-trained on ImageNet [8] and using an intra-image ranking signal [19], as well as examining the sorting signal alone. Our findings

demonstrate that our pre-training technique significantly outperforms these alternatives, and that regularizing the feature space improves performance.

Counting Network Training. In Sec. 3.2 we argued that finetuning the sorting network with the noisy synthetic counting data can distort the learned features. In Tab. 5, we examine different strategies for producing the counting network. Initially, we train the network solely with the counting signal, which confirms that relying exclusively on noisy synthetic counting data yields inferior results. Further investigation into finetuning the entire pre-trained sorting network reveals that such an approach adversely affects performance, supporting our argument that we should favor linear probing to avoid feature distortion.

Impact of Image Resolution. At inference time, we resize all images to a fixed resolution. However, many images are of a higher resolution than this fixed size. When we partition an image, we sample the patches from the higher resolution image to preserve image details. In Tab. 6, we explore the impact of this strategy and find that it improves performance for all datasets. This effect is especially pronounced for datasets with very high resolution images like QNRF.

5 Conclusions and Future Work

In this paper, we provide an *unsupervised* counting method that can be applied to *multiple object categories*. We have tackled this challenging task by using synthetic counting data generated by latent diffusion models (LDMs), which provides a flexible strategy for effectively eliminating the annotation burden. However, LDMs face challenges in reliably understanding object quantity, which results in noisy annotations. To mitigate this issue, we employed LDMs to create two additional types of synthetic data: one by manipulating the number of objects within images, which yields an ordered image with a weak but very reliable object quantity label, and the other by generating synthetic images with varying object density categories, offering the ability to detect dense image regions. Given these components, we deploy a novel inference time strategy that automatically detects when images are dense and then partitions those images to reduce the total number of objects per sub-image. We demonstrate our method’s superiority over the SOTA in unsupervised crowd counting and zero-shot crowd counting across multiple benchmark datasets. Our work not only significantly alleviates the annotation burden but outlines a new direction for unsupervised counting. Our approach, like others, has limitations. For instance, its performance falls short compared to supervised methods. Injecting limited reliable counting data into the counting network may enhance performance. Additionally, our method operates effectively on images generated by LDMs, but may not perform well on non-natural image datasets such as medical data. Addressing these challenges can be a future research direction.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the Discovery Grant program. We also extend our deepest gratitude to the Digital Research Alliance of Canada for the use of the Cedar supercomputing cluster, which was essential to our research. Finally, we thank the Computing Science Department at Simon Fraser University for their continued support and resources.

References

1. Arteta, C., Lempitsky, V., Zisserman, A.: Counting in the wild. In: European Conference on Computer Vision (2016)
2. Babu Sam, D., Agarwalla, A., Joseph, J., Sindagi, V.A., Babu, R.V., Patel, V.M.: Completely self-supervised crowd counting via distribution matching. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) *Computer Vision – ECCV 2022*. pp. 186–204. Springer Nature Switzerland, Cham (2022)
3. Bai, S., He, Z., Qiao, Y., Hu, H., Wu, W., Yan, J.: Adaptive dilated network with self-correction supervision for counting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)*
4. Cheng, Z.Q., Dai, Q., Li, H., Song, J., Wu, X., Hauptmann, A.G.: Rethinking spatial invariance of convolutional networks for object counting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 19638–19648 (June 2022)
5. Cheng, Z.Q., Dai, Q., Li, H., Song, J., Wu, X., Hauptmann, A.G.: Rethinking spatial invariance of convolutional networks for object counting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 19638–19648 (2022)
6. D’Alessandro, A.C., Mahdavi-Amiri, A., Hamarneh, G.: Learning-to-count by learning-to-rank. In: *2023 20th Conference on Robots and Vision (CRV)*. pp. 105–112 (2023). <https://doi.org/10.1109/CRV60082.2023.00021>
7. Gong, Y., Mori, G., Tung, F.: RankSim: Ranking similarity regularization for deep imbalanced regression. In: *International Conference on Machine Learning (ICML) (2022)*
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778 (2016)
9. He, R., Sun, S., Yu, X., Xue, C., Zhang, W., Torr, P., Bai, S., QI, X.: Is synthetic data from generative models ready for image recognition? In: *The Eleventh International Conference on Learning Representations (ICLR) (2023)*, <https://openreview.net/forum?id=nUmCcZ5RKF>
10. Hsieh, M.R., Lin, Y.L., Hsu, W.H.: Drone-based object counting by spatially regularized regional proposal networks. In: *The IEEE International Conference on Computer Vision (ICCV)*. IEEE (2017)
11. Hu, Y., Jiang, X., Liu, X., Zhang, B., Han, J., Cao, X., Doermann, D.: Nas-count: Counting-by-density with neural architecture search. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*. pp. 747–766. Springer (2020)

12. Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., Shah, M.: Composition loss for counting, density map estimation and localization in dense crowds. In: Proceedings of the European conference on computer vision (ECCV). pp. 532–546 (2018)
13. Jiang, R., Liu, L., Chen, C.: Clip-count: Towards text-guided zero-shot object counting. In: Proceedings of the 31st ACM International Conference on Multimedia. p. 4535–4545. MM ’23, Association for Computing Machinery, New York, NY, USA (2023)
14. Kumar, A., Raghunathan, A., Jones, R.M., Ma, T., Liang, P.: Fine-tuning can distort pretrained features and underperform out-of-distribution. In: International Conference on Learning Representations (2022), <https://openreview.net/forum?id=UYneFzXSJWh>
15. Lee, K.H., He, X., Zhang, L., Yang, L.: Cleanet: Transfer learning for scalable image classifier training with label noise. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
16. Lempitsky, V., Zisserman, A.: Learning to count objects in images. *Advances in neural information processing systems* **23** (2010)
17. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1091–1100 (2018)
18. Liang, D., Xie, J., Zou, Z., Ye, X., Xu, W., Bai, X.: Crowdclip: Unsupervised crowd counting via vision-language model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2893–2903 (June 2023)
19. Liu, X., Van De Weijer, J., Bagdanov, A.D.: Leveraging unlabeled data for crowd counting by learning to rank. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7661–7669 (2018)
20. Paiss, R., Ephrat, A., Tov, O., Zada, S., Mosseri, I., Irani, M., Dekel, T.: Teaching clip to count to ten. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3170–3180 (October 2023)
21. Pogančič, M.V., Paulus, A., Musil, V., Martius, G., Rolinek, M.: Differentiation of blackbox combinatorial solvers. In: International Conference on Learning Representations (2020), <https://openreview.net/forum?id=BkevoJSYPB>
22. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
23. Ranjan, V., Sharma, U., Nguyen, T., Hoai, M.: Learning to count everything. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
24. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
25. Shi, M., Hao, L., Feng, C., Liu, C., Cao, Z.: Represent, compare, and learn: A similarity-aware framework for class-agnostic counting. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
26. Shipard, J., Wiliem, A., Thanh, K.N., Xiang, W., Fookes, C.: Diversity is definitely needed: Improving model-agnostic zero-shot classification via stable diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 769–778 (June 2023)

27. Sindagi, V.A., Yasarla, R., Patel, V.M.: Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**(5), 2594–2609 (2020)
28. Trivedi, P., Koutra, D., Thiagarajan, J.J.: A closer look at model adaptation using feature distortion and simplicity bias. In: *The Eleventh International Conference on Learning Representations* (2023), https://openreview.net/forum?id=wkg_b4-IwTZ
29. Wan, J., Liu, Z., Chan, A.B.: A generalized loss function for crowd counting and localization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 1974–1983 (2021)
30. Wang, Q., Gao, J., Lin, W., Li, X.: Nwpu-crowd: A large-scale benchmark for crowd counting and localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020). <https://doi.org/10.1109/TPAMI.2020.3013269>
31. Wu, P., Zheng, S., Goswami, M., Metaxas, D., Chen, C.: A topological filter for learning with label noise. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) *Advances in Neural Information Processing Systems*. vol. 33, pp. 21382–21393. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/f4e3ce3e7b581ff32e40968298ba013d-Paper.pdf
32. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 589–597 (2016)
33. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017)
34. Zhu, Z., Dong, Z., Liu, Y.: Detecting corrupted labels without training a model to predict. In: *Proceedings of the 39th International Conference on Machine Learning, Proceedings of Machine Learning Research*, vol. 162, pp. 27412–27427. PMLR (17–23 Jul 2022), <https://proceedings.mlr.press/v162/zhu22a.html>