

APPENDIX

This appendix is organized as follows.

- In section A, we illustrate the simulated image embedding retrieval process, including the intermediate processes.
- In section B, we provide results and explanations for additional ablation studies.
- In section C, examples of the retrieved simulated samples in the case of negative samples and congested scenes.
- In section D, we compare the inference performance of the proposed method with other annotator-free crowd counting methods.
- In section E, we provide details of the datasets and metrics we used.
- In section F, the computational efficiency of the image retrieval process is analyzed.
- In section G, the computational cost and inference performance against counting performance are discussed.
- In section H, we provide a theoretical explanation for the improvement from the knowledge augmentation.

A IMAGE RETRIEVAL PROCESS

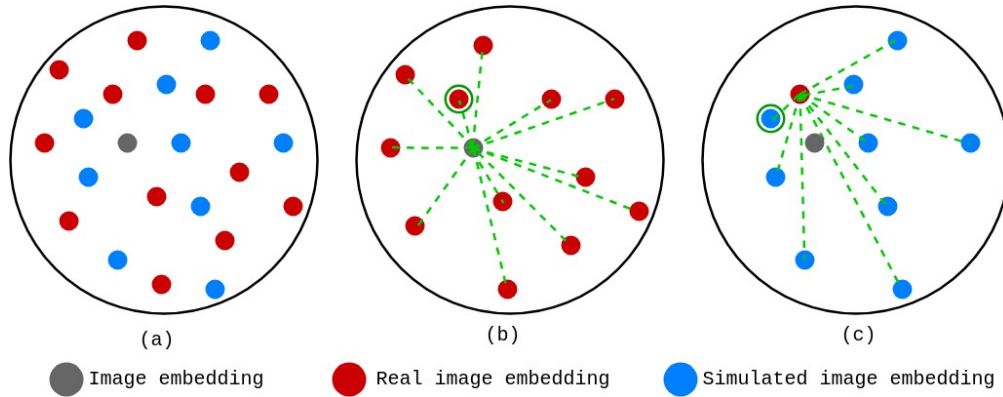


Figure 7: Knowledge retrieval process

In this section, we elaborate on the knowledge retrieval process described in section 3.2 using illustrations. In knowledge retrieval, we extract K image-text pairs for an input image I_i . First, we get the image embeddings \mathbf{e}_i of I_i using the pre-trained image encoder (Ψ). Since \mathbf{e}_i is produced from Ψ and Ψ is trained using real images and simulated images, we assume \mathbf{e}_i lies on the same embeddings space as the image embeddings of the real and simulated images. This is demonstrated in figure 7a where the gray color image embedding is in the same manifold as the red color real image embeddings and blue color simulated image embeddings. Next, we perform the maximum inner product search (MIPS) with the image embeddings of the real crowd image crops in the reference database.

In figure 7b, we demonstrate the retrieval of the closest embedding. In MIPS, first, we compute the distance between \mathbf{e}_i and real image embeddings under the vector inner product. Then, we find the real image embedding closest to or the most similar to \mathbf{e}_i . Then, we perform MIPS between the selected real image embedding and the simulated image embeddings. In figure 7c, we demonstrate the retrieval of the closest simulated embedding.

B ADDITIONAL ABLATION STUDIES

Detailed ablation on augmentations The crucial module of the pipeline is the Knowledge Augmentation Module (KAM). In the KAM, we use three different embedding augmentations, as depicted by the first three columns of table 4.

The performance gain by each augmentation type is provided for only \mathbf{v}_i^L in table 4. Therefore, to understand which augmentation types improve the performance, we provide the counting performance for each augmentation type and their combinations compared against the baseline performance in the table 7. The most performance gain has come from the components \mathbf{v}_i^L and \mathbf{v}_i^{VV} compared to the baseline method. The two augmentations deliver text information and visual information, respectively, but the cross-attention is taken between the image embeddings and retrieved patch embeddings. However, the performance gain from \mathbf{v}_i^{LV} is marginal compared to the other two augmentations where the cross-attention is taken between the image and retrieved text embeddings.

Table 7: Detailed ablation of different augmentations

Fusion type			K	MAE		
\mathbf{v}_i^L	\mathbf{v}_i^{LV}	\mathbf{v}_i^{VV}		JHU	SHA	QNRF
\times	\times	\times	-	170.2	143.1	223.6
\checkmark	\times	\times	32	148.6	123.6	215.4
\times	\checkmark	\times	32	152.8	128.3	219.8
\times	\times	\checkmark	32	145.3	125.8	216.5
\checkmark	\checkmark	\times	32	149.3	122.3	221.7
\checkmark	\times	\checkmark	32	142.8	118.8	215.7
\checkmark	\checkmark	\checkmark	32	142.3	118.4	214.9

Different retrieval processes. We consider the effect of not using real crowd images in the retrieval process and directly retrieving from the simulated dataset. However, the image encoder is pre-trained with real crowd images in the mix. When directly retrieving from the simulated dataset, we observed an MAE of 243.8 for JHU-Crowd++, which is poorer than the CrowdCLIP and CSS-CCNN performances. This is because, even though the image encoder is trained to embed real and simulated images in the same space, the training of the KAM and the decoder disregards the domain gap between real and simulated images.

Different amount of reference data. We evaluate the effect of the reference set size on the performance for five cardinalities by randomly sampling 10%, 25%, 50%, 75%, and 80% image-text pairs from GCC dataset. The ablation study recorded an average MAE of 224.2, 210.6, 174.0, 144.7, and 142.8, respectively, on JHU-Crowd++ for five trials. The performance was higher for larger reference set sizes. This is because in larger reference databases, for a given test image crop, a positive simulated crop is closer than in smaller databases, providing more accurate information retrieval.

C QUALITATIVE RESULTS

We provide qualitative results to demonstrate the performance of the retrieval process in the proposed method in figure 8. In the first row, we have a negative test image. Most of the retrieved test images for the negative sample had zero crowd counts and had similar backgrounds. Nonetheless, some retrieved patches had smaller counts (< 4) where the background was similar to the test image. In the second row, we have a congested test image. The retrieved patches for the congested scene are of similar crowd density patterns, even though most of the images do not fill up the entire image. This validates the idea that the retriever searches for simulated images that resemble the crowd density pattern of the test image, as first mentioned in section 5.2 with figure 4.

D INFERENCE SPEED

We present a comparison of inference speeds, as outlined in table 8. The runtime of our proposed annotator-free method is significantly higher than other annotator-free methods, such as CrowdCLIP and CSS-CCNN. CrowdCLIP gives an interval of FPS values as it uses progressive filtering for



Figure 8: Retrieved synthetic crops for a negative sample (top row) and congested sample (bottom row).

crowd patches with people, whereas the proposed work only has one forward pass through the image encoder. Then, CSS-CCNN utilizes a larger decoder to estimate the density map to predict the count, whereas we only use a linear layer to estimate the count directly. Additionally, the use of a vector database to retrieve samples improves inference time as the retrieval operation is simply the vector inner product. Notably, fully supervised methods necessitate maintaining high-resolution features to produce quality density maps. For instance, in CSRNet [Li et al. \(2018\)](#), features are 1/8 the size of the input, while in BL [Ma et al. \(2019\)](#), they are 1/16 the size, resulting in slower inference speeds.

Table 8: The comparisons of Frames Per Second (FPS) between our method and other methods. The results are conducted on an NVIDIA A6000 GPU

Method	Annotated data	Label	Resolution	FPS
CSRNet Li et al. (2018)	Real	density	1024×768	18.4
BL Ma et al. (2019)	Real	density	1024×768	21.3
CSS-CCNN Babu Sam et al. (2022)	\times	\times	1024×768	37.4
CrowdCLIP Liang et al. (2023)	Real	count text	1024×768	[24.0, 50.8]
Ours	Synthetic	count	1024×768	42.8

E DATASETS

JHU-Crowd++ [Sindagi et al. \(2020\)](#) contains 2,722 training images, 500 validation images, and 1,600 testing images, collected from diverse scenarios. The total number of people in each image ranges from 0 to 25,791.

ShanghaiTech [Zhang et al. \(2016\)](#) contains 1,198 crowd images with 330,165 annotations. The images of the dataset are divided into two parts: Part A and Part B. In particular, Part A contains 300 training images and 182 testing images, and Part B consists of 400 training images and 316 testing images.

UCF-QNRF [Idrees et al. \(2018\)](#) contains 1,535 images captured from unconstrained crowd scenes with about one million annotations. It has a count range of 49 to 12,865, with an average count of 815.4. Specifically, the training set consists of 1,201 images and the testing set consists of 334 images.

NWPU-Crowd [Wang et al. \(2020\)](#), a large-scale and challenging dataset, consists of 5,109 images, 2,133,375 instances annotated elaborately. To be specific, the images are randomly split into three parts, including training, validation, and testing sets, which contain 3,109, 500, and 1,500 images, respectively.

GCC Wang et al. (2019) dataset consists of 15, 212 images, with a resolution of 1080×1920 , containing 7, 625, 843 persons. Compared with the existing datasets, GCC is a larger-scale crowd counting dataset in terms of both the number of images and the number of persons.

Metrics we used for evaluate the counting performance were MAE and MSE as defined below:

$$\text{MAE} = \sum_{n=1}^N \frac{1}{N} |c_n - \hat{c}_n| \text{ and } \text{MSE} = \sqrt{\sum_{n=1}^N \frac{1}{N} |c_n - \hat{c}_n|^2}, \quad (8)$$

where c_n and \hat{c}_n are the groundtruth and predicted crowd count of the n^{th} image out the the N images tested.

F EFFICIENCY ANALYSIS

For the retrieval process, we use the naive maximum inner product search. This involves computing the similarity between image embeddings and crop embeddings in the reference database and sorting the similarity scores to find the closest neighbors.

Suppose the reference database is of size N , the embedding dimensionality is of size d , and we need to find the nearest k neighbors. Then, the computational efficacy of the whole process is $O(N \cdot d + N \cdot \log k)$. Accordingly, as the retrieval space scales, the time it takes for the retrieval process will increase. However, for larger reference databases, using approximation methods such as the k-d tree, the computational complexity can be reduced to $O(\log N)$ for smaller dimensional sizes, but still, the time consumed will increase with the size of the reference database.

G COMPUTATIONAL COST AND COMPLEXITY

We provide a comparison for the inference speed in table 8 in supplementary material. However, we will itemize the inference time and the computational complexity for the model with and without the KAM, along with the accuracy. For the proposed method, the inference time and computational complexity are influenced by three components: Image encoder and count decoder, knowledge retrieval process, and KAM. We tabulate the computational complexity in the following table.

Table 9: Computational efficiency of the architecture

	GFLOPS	Time (ms)	MAE
Baseline	70.564	8.55	170.2
MIPS	-	3.51	-
KAM	151.196	18.32	142.3

The MAE performance for the JHU public dataset is given in the above table. The baseline corresponds to the network without the KAM and the minimum model latency without the proposed improvements. The MIPS corresponds to the retrieval process with the inner product search to find the 16 nearest neighbors for a given image embedding.

H THEORETICAL ANALYSIS

To explain the contribution of knowledge augmentation to improving zero-shot crowd-counting, we use a probabilistic approach.

The goal is to predict the crowd-count c_i for the target embedding \mathbf{e}_i . Using a probabilistic framework, the prediction can be expressed as:

$$P_{source}(c_i | \mathbf{e}_i) = P_{source}(c_i | \mathbf{e}_i'),$$

where the augmented embedding is:

$$\mathbf{e}_i' = \mathbf{e}_i + \mathbf{v}_i^L + \mathbf{v}_i^{LV} + \mathbf{v}_i^{VV}.$$

Using Bayes' rule, we can rewrite the probability as follows:

$$P_{source}(c_i|\mathbf{e}'_i) \propto P(\mathbf{e}'_i|c_i)P_{source}(c_i),$$

where $P(\mathbf{e}'_i|c_i)$ and $P(c_i)$ denote the likelihood of the augmented embedding given the count and the prior probability of the count derived from the source distribution.

Then, the likelihood can be decomposed as

$$P(\mathbf{e}'_i|c_i) \propto P(\mathbf{e}_i|c_i) \prod_n P(\mathbf{v}_i^n|c_i)$$

where \mathbf{v}_i^n is each individual augmentation type from the KAM. However, each individual augmentation is computed from the KAM using the retrieved embeddings from the reference database. Therefore, the likelihood can be updated as:

$$P(\mathbf{e}'_i|c_i) \propto P(\mathbf{e}_i|c_i) \prod_n \prod_{k=1}^K P(r_{ik}^n|c_i)$$

where r_{ik}^n denotes the retrieved embedding augmented with multi-head attention (MHA), and k is the index of the retrieved embedding. Each \mathbf{v}_i^n thus encodes the aggregated likelihood information from its corresponding patches, ensuring that \mathbf{e}_{aug} effectively aligns with the count c_i as MHA behaves as a projection of the query embedding to the key embedding. Consequently, the retrieved embeddings \mathbf{v}_i^n encode domain-specific patterns, improving the likelihood estimation.

Without the retrieved embeddings, the likelihood distribution will only depend on \mathbf{e}_i , and as augmentations are introduced, the likelihood distribution is influenced by the source domain information. The influence of the source likelihood increases with the number of retrieved embeddings. In return, the posterior distribution $P_{source}(c_i|\mathbf{e}'_i)$ becomes a sharper posterior distribution. As the posterior distribution becomes sharper, the uncertainty involved with the prediction reduces, improving the prediction accuracy.