Low-Rank Similarity Mining for Multimodal Dataset Distillation

Yue Xu [1] **Zhilin Lin** [1] **Yusong Qiu** [1] **Cewu Lu** [1] **Yong-Lu Li** [1]

Abstract

Though dataset distillation has witnessed rapid development in recent years, the distillation of multimodal data, *e.g.*, image-text pairs, poses unique and under-explored challenges. Unlike unimodal data, image-text contrastive learning (ITC) data lack inherent categorization and should instead place greater emphasis on modality correspondence. In this work, we propose **Low-Rank Similarity Mining** (**LoRS**) for multimodal dataset distillation, that concurrently distills a ground truth similarity matrix with image-text pairs, and leverages low-rank factorization for efficiency and scalability. The proposed approach brings significant improvement to the existing algorithms, marking a significant contribution to the field of visual-language dataset distillation. We advocate adopting LoRS as a foundational synthetic data setup for image-text dataset distillation. **Our code is available at** `https://github.com/silicx/LoRS_Distill`.

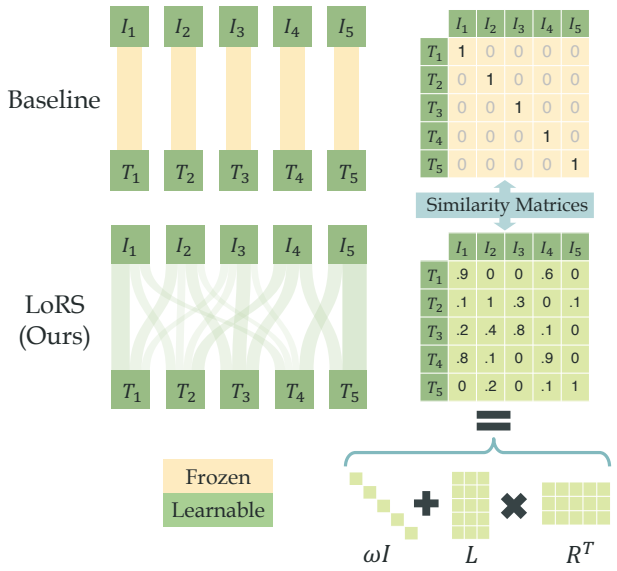


Figure 1. Vanilla dataset distillation could be adapted to image-text data but is limited by the fixed data pairing ("Baseline"). We propose similarity mining which simultaneously distills the ground truth similarity matrix, together with low-rank optimization for a fair data parameter size (**LoRS**). (I_i = i^{th} image, T_i = i^{th} text).

1. Introduction

Dataset distillation synthesizes a smaller and more compact dataset while retaining its essential information and model training performance. It becomes noteworthy in machine learning due to its high compression ratio, especially in the context of large-scale models and data. However, current algorithms are limited in the image domain and very few works involve other uni-modality such as text (Li & Li, 2021), video (Wang et al., 2023) or graph (Xu et al., 2023b) data. Since vision-language pre-training models (VLP) (Radford et al., 2021; Li et al., 2022) and multimodal large language models (MLLM) (Li et al., 2023; Liu et al., 2023a) become dominant, we direct our attention towards the paired image-text data. As the foundation of VLP, we focus on the image-text contrastive learning (ITC) data and aim for an effective image-text dataset distillation which could potentially enhance the efficiency and boost the study of multimodal models.

However, the distillation of image-text pairs is much more challenging than unimodal data: (1) The algorithm should not only compress each modality individually but should also correctly learn the *correspondence* between the modalities; (2) The unimodal data have categories and are distributed in clusters; but the image-text pair data is not categorized and distributed sparsely, which could lead to high sample variance for dataset distillation. As indicated by previous work (Lee et al., 2022), this fails existing algorithms like gradient matching (Zhao et al., 2020) and distribution matching (Zhao & Bilen, 2023). While the first work on image-text dataset distillation (Wu et al., 2023) achieves non-trivial performance with vanilla MTT (Cazenavette et al., 2022), it lacks specific adaption and exploitation of the image-text data. Therefore, we propose to emphasize learning the modality correspondence rather than summarizing the data patterns for each category.

[1]Shanghai Jiao Tong University. Correspondence to: Yong-Lu Li <yonglu_li@sjtu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

As shown in Fig. 1, vanilla dataset distillation algorithms exploit *fixed* image-text correspondence. To enhance the information density and bring more flexibility, we propose simultaneously learning the image-text similarity during dataset distillation as auxiliary information, *i.e.*, **Similarity Mining**. The distilled similarity matrix could be exploited in *any* image-text contrastive learning models, with only slight modifications to the contrastive loss functions. This method extends the vanilla distillation method to learn the full correspondence between synthetic image and text, which could be roughly considered as extending the N image-text pairs to N^2 paired data. Thus, we enrich the information of the synthetic data with little model overhead. We advocate adopting similarity mining as a fundamental algorithm setting in image-text dataset distillation.

To support the rationality and feasibility, we justify the similarity mining from the model learning perspectives: (1) **False negative mining**: the vanilla ITC models (*e.g.*, CLIP (Radford et al., 2021)) assume the samples in each batch are distinct so it uses the identity matrix as Ground Truth (GT) similarity matrix (only the sample itself is positive and other samples are all negative), but sometimes potential similarity between batch samples exists (Srinivasa et al., 2023), and similarity mining could find these samples and automatically fix the false loss penalty. (2) **Flexible contrastive learning anchors**: ITC can be regarded as the attraction and repulsion between the feature embedding and the anchor points. Similarity mining gives the flexibility to distinctly weigh the anchors so that some anchors can be equivalently merged without changing the learning dynamics, which will largely enhance the compression rate for dataset distillation. These will be detailed in Sec. 3.3.

The similarity mining works well when the data size $\tilde{M}$ is small. However, when the synthetic data size scales up, the parameter size of the similarity matrix could explode quadratically. *e.g.*, when $\tilde{M} = 100$, the similarity matrix is smaller than the one single image; but when $\tilde{M}$ grows to 10,000, the matrix is larger than the total image and text storage. Thus, we find that the similarity matrix is *sparse* in the non-diagonal area. For memory efficiency and to facilitate the learning of the similarity matrix, we exploit low-rank factorization for the similarity matrix. The similarity matrix is decomposed: $S = \omega I + \frac{\alpha}{r} LR^\top$, where the low-rank components L, R are $\tilde{M} \times r$ sized and reduce the space complexity to $O(\tilde{M})$. Overall, we propose **Lo**w **R**ank **S**imilarity Mining (**LoRS**) for image-text dataset distillation. Experiments show that our methods significantly enhance the distilled data performance and compression ratio.

The contribution of this work involves: **(1)** For image-text dataset distillation, we propose a new paradigm to learn the similarity matrix as a part of the synthetic data, which is well justified from the ITC training perspective. **(2)** We propose a novel and feasible implementation of similarity mining incorporating low-rank factorization. **(3)** Our method significantly outperforms the baseline methods with the same or smaller storage burden.

2. Related Work

2.1. Dataset Distillation

Dataset distillation (DD) aims to synthesize a small-scale dataset from a large-scale dataset, which can replace the original dataset in training while maintaining performance. Existing algorithms can be classified as: (1) **Meta-Model Matching**. Optimizing the empirical loss on the full dataset and keeping the transferability of distilled data. Following the first work of DD (Wang et al., 2018), many approaches have been proposed. KIP (Nguyen et al., 2020) integrates ridge regression to reduce computational complexity and further extend to an infinite wide network (Nguyen et al., 2021). RFAD (Loo et al., 2022) uses the Neural Network Gaussian Process kernel substitute in KIP. FRePo (Zhou et al., 2022) divides the network into a feature extractor and a classifier for optimization. RCIG (Loo et al., 2023) exploits implicit gradient to compute meta-gradient. (2) **Gradient-based**. DC (Zhao et al., 2020) aligns the training gradients of real and synthetic data. IDC (Kim et al., 2022b) improves DC by storing synthetic data in lower resolutions. MTT (Cazenavette et al., 2022) matches the parameters after multi-step training, which can be regarded as long-term gradient matching. TESLA (Cui et al., 2023) reduces the memory consumption of MTT. Shin *et al.*matches the loss sharpness of real and synthetic data, which is similar to the gradient. (3) **Feature-based**. DM (Zhao & Bilen, 2023) matches the distribution between real data and synthetic data, while CAFE (Wang et al., 2022) introduces layer-wise alignment of features. IDM (Zhao et al., 2023) further enhances DM with regularizer and model queue. (4) **Factorization Methods**. These methods decompose the data into bases and hallucinators, which can reduce the storage burden and increase the diversity of synthetic data. HaBa (Liu et al., 2022) employs convolutional network hallucinators, while LinBa (Deng & Russakovsky, 2022) uses linear ones. KFS (Lee et al., 2022) provides efficient sharing of information between generated examples and a better trade-off between compression ratio and quality. Frequency domain factorization has also been adopted (Shin et al., 2023).

Many other methods go beyond these categories and introduce innovations to DD. To optimize the existing methods, some works focus on data or model augmentation (Zhao & Bilen, 2021; Zhang et al., 2023) for generalization of DD, while some exploit sample selection for efficient DD (Liu et al., 2023c; Xu et al., 2023a; He et al., 2023) or to extend the application (Liu et al., 2023b; He et al., 2023). Generative models are used as synthetic image generators (Zhao

& Bilen, 2022; Cazenavette et al., 2023). SRe2L (Yin et al., 2023) proposes a 3-stage learning paradigm that is more efficient for large datasets. Bayesian inference also could be adopted for dataset distillation (Manousakas et al., 2020; Kim et al., 2022a; Tiwary et al., 2023). Wu *et al.*propose the first work on image-text dataset distillation (Wu et al., 2023) and achieve decent performance, by matching the trajectories of both image and text encoders. However, it does not make specific adaptations to the image-text data.

2.2. Image-text Contrastive Learning

Image-text contrastive learning is a crucial foundation of multimodal learning. CLIP (Radford et al., 2021) first adopts image-text contrastive learning which aligns the features obtained from encoders of different modalities. The model is trained on large-scale data to achieve *scale effect* and open-vocabulary transferability. ALIGN (Jia et al., 2021) and Flava (Singh et al., 2022) were among the first to present work on comparative learning. CHiLS (Novack et al., 2023) explored richer embedding with label hierarchy. FILIP (Yao et al., 2021) explored the toke-wise similarity between two modalities. ALBEF (Li et al., 2021) and CoCa (Yu et al., 2022) focused on cross-modal attention. BLIP (Li et al., 2022) and BLIP2 (Li et al., 2023) made a combination of methods for multimodal learning and performed well. There is also some recent work focusing on soft labels in CLIP-like models. SoftCLIP (Gao et al., 2023) achieved soft cross-modal alignment by generating intra-modal similarity. Andonian et al. (2022) used progressive self-distillation to learn robust models from noisy data.

3. Methodology

3.1. Preliminary

We first formulate the image-text contrastive learning (ITC). Based on an image-text pair dataset $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ with M paired images $\mathcal{X} = \{x_i\}_M$ and texts $\mathcal{Y} = \{y_i\}_M$, an ITC model (*e.g.*, CLIP) consists of an image encoder $u_i = f_V(x_i)$ and a text encoder $v_i = f_T(y_i)$. The model enables cross-modal retrieval with the cosine similarity metric $\hat{s}_{ij} = \cos\langle u_i, v_j \rangle$. The model is trained with contrastive learning such as InfoNCE (Oord et al., 2018) loss:

$$\begin{aligned} \mathcal{L}_{\text{NCE}}(\mathcal{B}) &= -\frac{1}{m}\sum_{i=1}^{m} \log(P_{ii}^V) + \log(P_{ii}^T), \\ P_{ij}^V &= \frac{\exp(\hat{s}_{ij}/\tau)}{\sum_k^m \exp(\hat{s}_{ik}/\tau)}, \; P_{ij}^T = \frac{\exp(\hat{s}_{ij}/\tau)}{\sum_k^m \exp(\hat{s}_{kj}/\tau)}, \end{aligned} \tag{1}$$

where $\mathcal{B} \subset \mathcal{D}$ is a data batch with size m. P_{ij}^V and P_{ij}^T are softmax probability of the estimated similarity $\hat{s}_{ij}$ and τ is a temperature factor. InfoNCE loss assumes that for each image x_i, only the paired text y_i is the positive anchor, and the other texts $y_k, k \neq i$ are negative. So it is aligning the estimated similarity matrix $\hat{S} = \{\hat{s}_{ij}\}$ to an identity GT similarity matrix I.

Table 1. Retrieval performance of different distillation algorithms. 200 pairs are synthesized on Flickr30k (Plummer et al., 2015). Details of the metrics please refer to Sec 4.1.

	DC (Zhao et al., 2020)	DM (Zhao & Bilen, 2023)	TESLA (Cui et al., 2023)
IR@1	0.1±0.0	0.6±0.1	**8.6±0.3**
IR@5	0.4±0.1	2.1±0.2	**25.3±0.2**
IR@10	1.0±0.1	3.7±0.4	**36.6±0.3**
TR@1	0.3±0.0	1.1±0.3	**14.5±0.5**
TR@5	0.6±0.0	4.5±0.1	**38.7±0.5**
TR@10	1.0±0.2	7.1±0.1	**53.4±0.5**

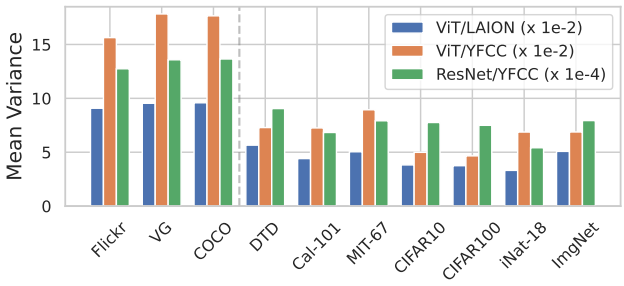


Figure 2. The image feature variance on different datasets. We adopt CLIP encoders with ResNet or ViT, pretrained on LAION or YFCC datasets. Three datasets at left: image-text datasets; seven at right: classification datasets.

The image-text data distillation is to **learn** a smaller synthetic dataset $\tilde{\mathcal{D}} = (\tilde{\mathcal{X}}, \tilde{\mathcal{Y}})$, *i.e.*, a data distillation algorithm would regard $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}$ as learnable parameters and optimize them based on the knowledge from real dataset $\mathcal{D}$. Most existing dataset distillation algorithms could be seamlessly applied to image-text data. However, Tab. 1 indicates that the short-term matching algorithms such as gradient matching (Zhao et al., 2020) and distribution matching (Zhao & Bilen, 2023) fail. This is due to the large data variance of image-text data since no category exists (Fig. 2, details please refer to Appendix Sec. A). As discussed by (Lee et al., 2022), these methods will suffer from the high batch variance, so they fail on the image-text data. Therefore, we exploit long-term matching algorithms like MTT (Cazenavette et al., 2022). It matches the model parameters after being trained on the real or synthetic data for multiple training steps so that the batch variance can be alleviated due to multi-step training.

In detail, assume the ITC model $\{f_V, f_T\}$ is parameterized by $\theta = (\theta_V, \theta_T)$. Starting from any initial parameter θ_0, MTT algorithms would train the ITC model on real data for T steps to θ_T, and train on synthetic data $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}$ for t steps to $\tilde{\theta}_t$. The synthetic data is optimized by minimizing the

long-term parameter matching loss:

$$\tilde{\mathcal{X}}, \tilde{\mathcal{Y}} = \arg\min_{\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}} \frac{\|\tilde{\theta}_t - \theta_T\|^2}{\|\theta_0 - \theta_T\|^2}. \tag{2}$$

$\tilde{\theta}_t$ is the function of $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}$ so the synthetic data will receive the gradients for optimization. In practice, we use TESLA (Cui et al., 2023) to reduce the memory consumption of the MTT algorithm.

3.2. Similarity Mining for Image-Text Distillation

Different from the uni-modal dataset distillation, for the image-text distillation, we should put more emphasis on learning the *correspondence* between the modalities other than the compression of each single modality. To enhance the information density of image-text data, we propose to simultaneously learn the GT similarity matrix, namely **similarity mining**. For M image-text pairs, the traditional ITC model assumes identity GT similarity, *i.e.*, only i^{th} image and i^{th} text are paired and others are all negative pairs. However, similarity mining assumes each combination of the data x_i, y_j is a pair but with different GT similarity s_{ij}, therefore we have $M \times M$ valid data pairs without increasing the data scale. The traditional ITC model is a special case when similarity matrix $S = I$. This intuitively enhances the correspondence information within the same number of image-text data.

Since InfoNCE focuses on the positive pairs, we shall extend the contrastive loss to fit a continuous GT similarity matrix to enable similarity mining. Given GT similarity $S = \{s_{ij}\}$, we propose the following implementations:

Extended InfoNCE (eNCE). We could extend InfoNCE to consider both positives and negatives:

$$\mathcal{L}_{\text{eNCE}}(\mathcal{B}, S) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{m} s_{ij} \left(\log(P_{ij}^V) + \log(P_{ij}^T) \right). \tag{3}$$

The eNCE resembles CWCL (Srinivasa et al., 2023) but without normalization. Note that the loss degrades to InfoNCE when GT $S = I$.

Binary Cross Entropy (BCE). The multi-label binary cross-entropy loss is suitable for the continuous similarity:

$$\mathcal{L}_{\text{BCE}}(\mathcal{B}, S) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{m} \ell\left(s_{ij}, \sigma(\hat{s}_{ij}/\tau)\right), \tag{4}$$

where σ is the sigmoid function and $\ell(y, p) = -y \log(p) - (1 - y) \log(1 - p)$.

Weighted Binary Cross Entropy (wBCE). Unlike InfoNCE, the BCE loss produces a large penalty for negative pairs. Since there are significantly more negative pairs than positives, we propose a weighted version of BCE by separately averaging the losses for positives and negatives:

$$\begin{aligned}\mathcal{L}_{\text{wBCE}}(\mathcal{B}, S) = &\frac{1}{|\{s_{ij} > \beta\}|} \sum_{i,j:s_{ij}>\beta} \ell\left(s_{ij}, \sigma(\hat{s}_{ij}/\tau)\right) \\ &+ \frac{1}{|\{s_{ij} \leq \beta\}|} \sum_{i,j:s_{ij}\leq\beta} \ell\left(s_{ij}, \sigma(\hat{s}_{ij}/\tau)\right),\end{aligned} \tag{5}$$

where β is the positive/negative threshold and set to 0.5.

More specifically, based on these continuous contrastive losses, the image-text distillation with similarity mining is to learn an augmented synthetic dataset $\tilde{\mathcal{D}} = (\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}, \tilde{S})$, where $\tilde{S} = \{\tilde{s}_{ij}\}$ and $\tilde{s}_{ij}$ is the GT similarity between image x_i and text y_j. The synthetic data is learned with MTT loss:

$$\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}, \tilde{S} = \arg\min_{\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}, \tilde{S}} \frac{\|\tilde{\theta}_t - \theta_T\|^2}{\|\theta_0 - \theta_T\|^2}. \tag{6}$$

To use the distilled data, all the ITC models could be trained with the continuous contrastive losses above on the augmented synthetic dataset (Alg. 2). Similarity mining could be seen as a data reorganization method like LinBa (Deng & Russakovsky, 2022), HaBa (Liu et al., 2022), so it is **plug-and-play** for any base distillation algorithm. We will demonstrate its effectiveness both theoretically and empirically, foreseeing much potential for visual language pretraining and multimodal large models. We hope it will become the standard paradigm for image-text dataset distillation.

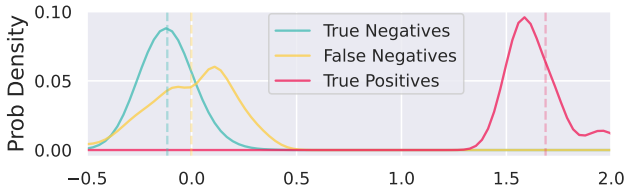


Figure 3. The histogram of the similarity value learned by similarity mining. False negatives are deliberately constructed and can be found by the algorithm.

3.3. Justification of Similarity Mining

The similarity mining technique could be justified from two perspectives:

False negative mining. Standard ITC models like CLIP assume that the image and text from different samples are negative pairs, which could be violated due to potential same or similar data samples in the noisy internet data. These potentially associated pairs could be ignorable for large-scale datasets like YFCC100M (Thomee et al., 2016) or LAION (Schuhmann et al., 2022) since there are enough true positives and negatives to draw the representations to the correct position. However, the small scale of synthetic

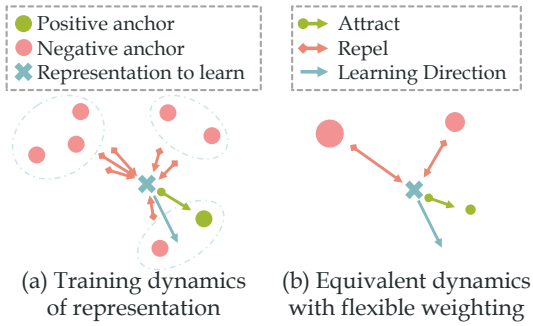


Figure 4. (a) Training dynamical system of a representation: it is attracted or repelled by the anchors. (b) If the anchor is flexibly weighted, dynamics could be equivalent to a system that has fewer components, and similarity mining could offer this flexibility.

data leads to low robustness to the false negatives and it requires a more accurate GT similarity.

So the similarity mining paradigm could alleviate the impact of false negatives since it can impose non-zero similarity for *potential* negative pairs. We conduct a toy experiment on Flickr30k. We initialize 100 synthetic pairs with 50 real data pairs and their replicates so that the ith and $(i+1)^{\text{th}}$ samples will be similar during distillation but regarded as negative pairs. Finally, there are 100 true positive pairs, 100 false negatives, and 9,800 true negatives. We show the normalized histogram of the distilled similarity in Fig. 3 and the similarity mining technique does find the false negatives by learning a relatively larger similarity value.

Flexible contrastive learning anchors. We would dig deeper into the image-text contrastive learning by first analyzing the contrastive loss gradients. For conciseness, the following discussion assumes the image and text representations u_i, v_j are normalized, and without loss of generality, we only discuss the gradient on image representation.

Proposition 3.1. *The gradient of InfoNCE loss wrt. the image representation u_n is:*

$$\frac{\partial \mathcal{L}}{\partial u_n} = \sum_{j=1}^{m} W_j v_j,\ W_j = \begin{cases} \frac{P^V_{nj}+P^T_{nj}-2}{m\tau}, & \text{if } j = n \\ \frac{P^V_{nj}+P^T_{nj}}{m\tau}, & \text{if } j \neq n \end{cases}. \quad (7)$$

The softmax probabilities P^V_{nj} and P^T_{nj} can be interpreted as the relative similarity between u_n and v_j. Given learning rate γ, the training dynamical system:

$$\dot{u}_n = -\gamma \frac{\partial \mathcal{L}}{\partial u_n} = \sum_{j=1}^{m} (-\gamma W_j) v_j. \quad (8)$$

So in a physical analogy, the training dynamics resemble the particle u_n is "attracted" or "repelled" by other particles v_j. The positive or negative *anchor points* v_j span a gradient field which displaces the representation u_i through time. Eq. 8 also reveals *approximately* linearity of the dynamics, *i.e.*, the anchors v_j independently affect the representation u_n and the overall dynamics depend on their linear superposition. It is not strictly linear since the P^V and P^T are *relative* similarity that involve other v_k components.

Since the system is linear, as shown in Fig. 4, it can be intuitively considered that if the anchor points have distinct weights, then multiple anchor points could be merged as one and the system will be equivalently transformed into a system with fewer components, which reduces the necessary data points and enables a larger compression ratio for dataset distillation. Here, the similarity mining offers flexibility and induces learnable weights W_j:

Proposition 3.2. *The gradient of extended InfoNCE loss wrt. the image representation u_n is:*

$$\frac{\partial \mathcal{L}}{\partial u_n} = \sum_{j=1}^{m} W_j v_j,\ W_j = \frac{s_{n\bullet} P^V_{nj} + s_{\bullet j} P^T_{nj} - 2 s_{nj}}{m\tau}, \quad (9)$$

$s_{n\bullet} = \sum_k^m s_{nk}$ and $s_{\bullet j} = \sum_k^m s_{kj}$ are marginal similarity.

Proposition 3.3. *The gradient of BCE loss wrt. the image representation u_n is:*

$$\frac{\partial \mathcal{L}}{\partial u_n} = \sum_{j=1}^{m} W_j v_j,\ W_j = \frac{\sigma(\hat{s}_{nj}/\tau) - s_{nj}}{m\tau}. \quad (10)$$

The gradient of $\mathcal{L}_{\text{wBCE}}$ is the weighted version of BCE. For all three loss functions, the weights of anchors involve learnable similarity element s_{ij}, bringing the flexibility of anchor significance and enabling a more compact and data-efficient learning system. The derivation of the equations and propositions above are given in the Appendix Sec. B.

3.4. Low Rank Similarity Mining

Though similarity mining could help the dataset distillation task, the size of extra storage of the similarity matrix increases quadratically and could even exceed the image and text storage when the data size is large. The large similarity matrix will also be tricky to optimize and more training iterations are required to train the matrix fully. Therefore, we exploit the low-rank nature of the similarity matrix to reduce the storage burden.

Ideally, from the perspective of false negative mining, the similarity matrix is instinctively low-rank: if two samples are similar, the two rows or columns in the similarity matrix will be similar according to triangle inequality, which induces a low-rank similarity matrix (Appendix Sec. C.1). However, the learned similarity matrix is not and we hope our method could model similarity matrices of all different ranks, including the simplest but full rank identity similarity matrix. So following (Hu et al., 2021), we propose to

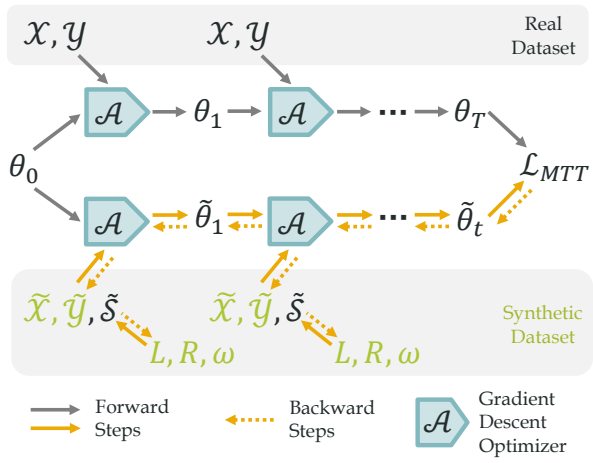


Figure 5. Computation graph of the proposed method **LoRS**. The green nodes are part of the learnable synthetic dataset.

apply low-rank approximation to the *residual similarity matrix*, *i.e.*, we factorize the similarity matrix $\tilde{S}$ into learnable diagonal and low-rank residual matrix:

$$\tilde{S} = \omega I + \frac{\alpha}{r} L R^\top, \tag{11}$$

where the diagonal weight $\omega \in \mathbb{R}^N$ and the low rank components $L, R \in \mathbb{R}^{N \times r}$ are learnable parameters. r is the rank of the residual similarity and α is a weighting factor, which is tuned as hyper-parameters. Therefore the parameter size of the similarity matrix is $N(2r+1)$, which is linear to the data size. In practice, we carefully select a small r and reduce the number of synthetic pairs to maintain the number of learnable parameters of the synthetic data, for a fair comparison to the baseline methods. The ω is initialized to ones, the L is initialized randomly and R is initialized to zero, to initialize $S = I$ and avoid the gradient vanishing. We also empirically analyze the learned similarity matrix to justify the low-rank technique in Appendix Sec. C.2 and Fig. 9 due to the page limit.

Overall, we propose the **L**ow-**R**ank **S**imilarity Mining (**LoRS**) technique for the image-text dataset distillation. This approach introduces a new component to the paired multimodal data but can be seamlessly embedded into all multimodal contrastive learning algorithms. The overview of the computation graph is also shown in Fig. 5. The learnable parameters of the synthetic data are $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}, L, R, \omega$, where L, R, ω are first composed to the synthetic similarity matrix $\tilde{\mathcal{S}}$, and then used to update the network parameters θ for the synthetic trajectory. The synthetic and real trajectories are aligned with MTT loss and update the 5 parameters by backpropagation. The algorithms are also summarized as Alg. 1 taking $\mathcal{L}_{wBCE}$ as an example, and the usage of synthetic data from LoRS is given in Alg. 2.

Algorithm 1 **L**ow-**R**ank **S**imilarity Mining (**LoRS**)

Input: Real data $\mathcal{X}, \mathcal{Y}$
Output: Synthetic data $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}$, synthetic similarity matrix $\tilde{S} = \omega I + \frac{\alpha}{r} L R^\top$

1: Initialize $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}$ by real data, $\omega = \mathbf{1}, L \sim \mathcal{N}, R = 0$,
2: **repeat**
3: Sample an initial network parameter θ_0
4: Train the network for T steps on $\mathcal{X}, \mathcal{Y}$ to θ_T
5: Train the network for t steps on $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}, \tilde{S}$ for t steps to $\tilde{\theta}_t$ with $\mathcal{L}_{\text{wBCE}}$, and keep the computation graph
6: Compute MTT loss $\mathcal{L}_{MTT} = \|\tilde{\theta}_t - \theta_T\|^2 / \|\theta_0 - \theta_T\|^2$
7: $X \leftarrow X - \gamma_X \nabla_X \mathcal{L}$; $Y \leftarrow Y - \gamma_Y \nabla_Y \mathcal{L}$
$L \leftarrow L - \gamma_s \nabla_L \mathcal{L}$, $R \leftarrow R - \gamma_s \nabla_R \mathcal{L}$
$\omega \leftarrow \omega - \gamma_s \nabla_\omega \mathcal{L}$
8: **until** convergence

Algorithm 2 Train a network with LoRS synthetic data

Input: Synthetic data $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}$, synthetic similarity matrix $\tilde{S}$
Output: A trained network θ

1: Random initialize network θ
2: **repeat**
3: Sample a batch $\{\tilde{x}, \tilde{y}\}_m$ and their similarity $\{\tilde{s}\}_{m \times m}$
4: Compute $\mathcal{L}_{\text{wBCE}}$ loss $\mathcal{L}_{\text{wBCE}}(\{\tilde{x}, \tilde{y}\}_m, \{\tilde{s}_{ij}\})$
5: $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}_{\text{wBCE}}$
6: **until** convergence

4. Experiments

4.1. Dataset and Metrics

We evaluate our method on Flickr30k (Plummer et al., 2015) and COCO (Lin et al., 2014) following the strong baseline (Wu et al., 2023) which exploits the MTT (Cazenavette et al., 2022) algorithm. Flickr30k and COCO are image captioning datasets with 31K and 123K images respectively, and each image is paired with five captions. The model performance is commonly measured by the recall of top-K retrieval (**R@K**): given a query from one modality, we retrieve the closest k matches from the other modality and measure the correctness. We denote the text-to-image retrieval as **IR@K** and image-to-text retrieval as **TR@K**.

4.2. Baselines and Proposed Method

We compare various baselines, involving:

(1) **Coreset Selection**: Random (random select a data subset), Herd (Welling, 2009), K-center (Farahani & Hekmatfar, 2009) and Forgetting (Toneva et al., 2018).

(2) **Dataset distillation**: MTT-VL (Wu et al., 2023) adapts MTT (Cazenavette et al., 2022) to image-text pairs (or namely MTT_{NCE}). TESLA (Cui et al., 2023) is an efficient implementation of MTT, so we also adapt TESLA to

Table 2. Results on Flickr30k (Plummer et al., 2015). The model trained on full dataset performs: IR@1=27.3, IR@5=57.1, IR@10=69.7; TR@1=33.9, TR@5=65.1, TR@10=75.2.

Pairs	Ratio	Metric	Coreset Selection				Dataset Distillation		
			Rand	Herd	K-Cent	Forget	MTT-VL	TESLA$_{\text{wBCE}}$	LoRS$_{\text{wBCE}}$
100 (99→LoRS)	0.3%	IR@1	1.0	0.7	0.7	0.7	4.7±0.2	0.5±0.2	**8.3±0.2**
		IR@5	4.0	2.8	3.1	2.4	15.7±0.5	2.3±0.2	**24.1±0.2**
		IR@10	6.5	5.3	6.1	5.6	24.6±1.0	4.7±0.4	**35.1±0.3**
		TR@1	1.3	1.1	0.6	1.2	9.9±0.3	5.5±0.5	**11.8±0.2**
		TR@5	5.9	4.7	5.0	4.2	28.3±0.5	19.5±0.9	**35.8±0.6**
		TR@10	10.1	7.9	7.6	9.7	39.1±0.7	28.9±1.0	**49.2±0.5**
200 (199 → LoRS)	0.7%	IR@1	1.1	1.5	1.5	1.2	4.6±0.9	0.2±0.1	**8.6±0.3**
		IR@5	4.8	5.5	5.4	3.1	16.0±1.6	1.3±0.2	**25.3±0.2**
		IR@10	9.2	9.3	9.9	8.4	25.5±2.6	2.5±0.2	**36.6±0.3**
		TR@1	2.1	2.3	2.2	1.5	10.2±0.8	2.8±0.5	**14.5±0.5**
		TR@5	8.7	8.4	8.2	8.4	28.7±1.0	10.4±1.5	**38.7±0.5**
		TR@10	13.2	14.4	13.5	10.2	41.9±1.9	17.4±1.6	**53.4±0.5**
500 (499 → LoRS)	1.7%	IR@1	2.4	3.0	3.5	1.8	6.6±0.3	1.1±0.2	**10.0±0.2**
		IR@5	10.5	10.0	10.4	9.0	20.2±1.2	7.3±0.4	**28.9±0.7**
		IR@10	17.4	17.0	17.3	15.9	30.0±2.1	12.6±0.5	**41.6±0.6**
		TR@1	5.2	5.1	4.9	3.6	13.3±0.6	5.1±0.2	**15.5±0.7**
		TR@5	18.3	16.4	16.4	12.3	32.8±1.8	15.3±0.5	**39.8±0.4**
		TR@10	25.7	24.3	23.3	19.3	46.8±0.8	23.8±0.3	**53.7±0.3**

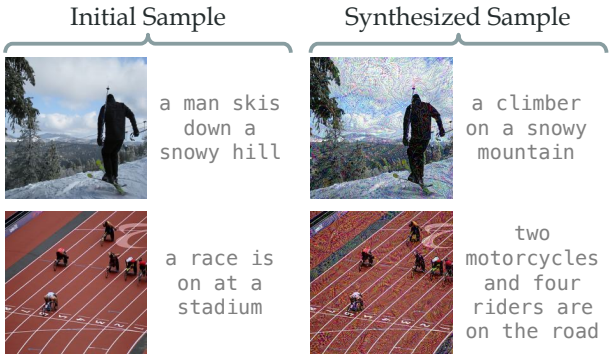


Figure 6. Examples of initialization and synthetic image-text pairs.

multimodal data with wBCE loss (TESLA$_{\text{wBCE}}$).

In comparison, we apply our LoRS technique to the TESLA (Cui et al., 2023) algorithm with weighted BCE loss (LoRS$_{\text{wBCE}}$).

4.3. Implementation Details

Following the setting of MTT strong baseline (Wu et al., 2023), we adopt ImageNet (Deng et al., 2009) pretrained NormalizerFree ResNet (NFNet) (Brock et al., 2021) as image encoder and pretrained BERT-base (Devlin et al., 2018) as text encoder. A linear layer is attached after the text encoder. At both the distillation and training stages, the pretrained weights are loaded and the text network is frozen for efficiency. We directly synthesize the text embedding rather than captions. We use TESLA (Cui et al., 2023) as the base distillation algorithm without label learning. We train the network on the full real dataset for 10 epochs 20 times as the expert trajectories. The experiments are conducted on one RTX 4090 GPU, revealing the efficiency of the method.

In the distillation stage, the images are resized to 224×224 and the text embeddings are 768-d. the synthetic data is learned with SGD and momentum 0.5. The image and text are initialized with random real samples. The rest hyper-parameters including learning rate and LoRS parameters vary among different datasets and synthetic data sizes, which are listed in Appendix Sec. F due to page limit. Particularly, for a fair comparison, we reduce the number of synthetic pairs for LoRS to maintain the synthetic parameters, *e.g.*for experiments with pairs=500, we reduce the number of pairs to 499 for LoRS to save $3\times224^2+768 = 151K$ parameters, which supports a maximum rank of $r = 150$. In practice, we use a smaller rank for efficiency, usually < 50.

4.4. Results

The results on Flickr30k and COCO are shown in Tab. 2 and 3. Compared to baselines, LoRS enhances the image-text distillation algorithms and could bring up to $\sim 50\%$ relative improvement. It is interesting that on Flickr30k, LoRS$_{\text{wBCE}}$ with 100 pairs significantly outperforms the MTT baseline with 500 pairs, showing the larger compression ratio of the similarity mining technique. It is worth noting that though LoRS changes the data structure, it only brings negligibly 0.3% memory and 0.8% training time overhead. For more analysis on the efficiency please refer to Appendix Sec. D.3. And the algorithm performance is more significant on Flickr30k since COCO is 3× larger than Flickr30k and has more complex data relationships.

Table 3. Results on COCO (Lin et al., 2014). The model trained on full dataset performs: IR@1=16.9, IR@5=41.9, IR@10=55.9; TR@1=19.6, TR@5=45.6, TR@10=59.5.

PAIRS	RATIO	METRIC	CORESET SELECTION				DATASET DISTILLATION		
			RAND	HERD	K-CENT	FORGET	MTT-VL	$\text{TESLA}_{\text{wBCE}}$	$\text{LoRS}_{\text{wBCE}}$
100 (99→LoRS)	0.8‰	IR@1	0.3	0.5	0.4	0.3	1.3±0.1	0.3±0.2	**1.8±0.1**
		IR@5	1.3	1.4	1.4	1.5	5.4±0.3	1.0±0.4	**7.1±0.2**
		IR@10	2.7	3.5	2.5	2.5	9.5±0.5	1.8±0.5	**12.2±0.2**
		TR@1	0.8	0.8	1.4	0.7	2.5±0.3	2.0±0.2	**3.3±0.2**
		TR@5	3.0	2.1	3.7	2.6	10.0±0.5	7.7±0.5	**12.2±0.3**
		TR@10	5.0	4.9	5.5	4.8	15.7±0.4	13.5±0.3	**19.6±0.3**
200 (199 → LoRS)	1.7‰	IR@1	0.6	0.9	0.7	0.6	1.7±0.1	0.1±0.1	**2.4±0.1**
		IR@5	2.3	2.4	2.1	2.8	6.5±0.4	0.2±0.1	**9.3±0.2**
		IR@10	4.4	4.1	5.8	4.9	12.3±0.8	0.5±0.1	**15.5±0.2**
		TR@1	1.0	1.0	1.2	1.1	3.3±0.2	0.7±0.2	**4.3±0.1**
		TR@5	4.0	3.6	3.8	3.5	11.9±0.6	3.1±0.5	**14.2±0.3**
		TR@10	7.2	7.7	7.5	7.0	19.4±1.2	5.3±0.8	**22.6±0.2**
500 (499 → LoRS)	4.4‰	IR@1	1.1	1.7	1.1	0.8	2.5±0.5	0.8±0.2	**2.8±0.2**
		IR@5	5.0	5.3	6.3	5.8	8.9±0.7	3.6±0.6	**9.9±0.5**
		IR@10	8.7	9.9	10.5	8.2	15.8±1.5	6.7±0.9	**16.5±0.7**
		TR@1	1.9	1.9	2.5	2.1	5.0±0.4	1.7±0.4	**5.3±0.5**
		TR@5	7.5	7.8	8.7	8.2	17.2±1.3	5.9±0.8	**18.3±1.5**
		TR@10	12.5	13.7	14.3	13.0	26.0±1.9	10.2±1.0	**27.9±1.4**

Table 4. Cross architecture generalization. The data is synthesized with NFNET+BERT and evaluated on various architectures.

PAIRS	METHOD	EVALUATE MODEL	FLICKR					
			IR@1	IR@5	IR@10	TR@1	TR@5	TR@10
499	$\text{TESLA}_{\text{NCE}}$	NFNET+BERT	8.2±0.1	23.9±0.4	34.7±0.4	13.0±0.5	34.5±0.5	49.4±0.5
		RESNET+BERT	3.0±0.2	10.8±0.5	17.0±0.8	6.0±0.9	18.8±0.7	27.7±1.2
		REGNET+BERT	3.2±0.8	11.1±1.8	17.5±1.3	5.8±0.1	18.6±0.6	28.1±1.0
499	$\text{LoRS}_{\text{wBCE}}$	NFNET+BERT	10.0±0.2	28.9±0.7	41.6±0.6	15.5±0.7	39.8±0.4	53.7±0.3
		RESNET+BERT	3.3±0.2	12.7±0.3	20.4±0.2	6.8±0.2	19.6±1.3	31.1±0.3
		REGNET+BERT	3.5±0.1	12.6±0.3	21.1±0.4	6.8±0.3	20.8±0.3	30.2±0.3

4.5. Cross Architecture Generalization

Following MTT (Cazenavette et al., 2022), we conduct a cross-architecture evaluation to study the generalization of the synthetic data. We distill the data with NFNet+BERT and evaluate it with other networks including RegNet (Radosavovic et al., 2020) and ResNet50 (He et al., 2016). It is not necessary to validate the generalization of the text network as we freeze the text encoder. Results in Tab. 4 show that our distilled data could generalize across networks (significantly surpasses the coreset selection methods in Tab. 2), and also outperform the baseline model. Note that the performance drop is also partly due to the performance of the architectures themselves (*e.g.*ResNet or RegNet trained on full data achieves about IR@1=28% and TR@1=22%, while NFNet achieves about IR@1=33% and TR@1=27%).

4.6. Ablation Study

Tab. 5 shows the results of the ablation study.

Learn full similarity matrix (No. 1-3). We implemented similarity mining with full learnable similarity matrix ($\tilde{N} \times \tilde{N}$ parameters, without the low-rank technique). The full similarity mining shows comparable performance with LoRS, indicating the feasibility of the low-rank approximation of the similarity matrix.

Losses (No. 4-6). Among the losses, $\mathcal{L}_{\text{wBCE}}$ slightly surpasses $\mathcal{L}_{\text{eNCE}}$, while significantly outperforms vanilla $\mathcal{L}_{\text{BCE}}$, mainly due to their balancedness. Along with the comparison of $\mathcal{L}_{\text{NCE}}$ in Tab. 2 and 3, we suggest choosing between $\mathcal{L}_{\text{wBCE}}$ and $\mathcal{L}_{\text{eNCE}}$ for LoRS.

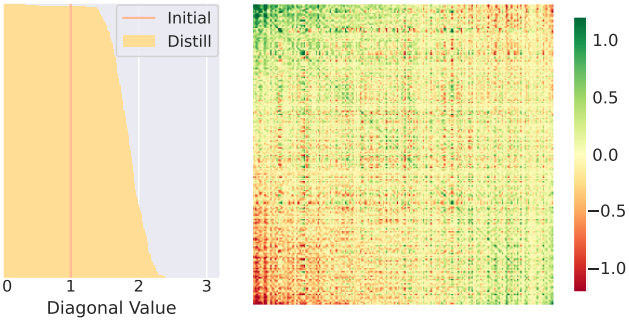
Rank r (No. 7-11). As long as r is not too small, it slightly affects the performance and $r = 20$ is sufficient here.

Components in low-rank factorization (No. 12-13). Removing the low-rank components L, R reduces the performance but still surpasses the one with an identity matrix (No. 13).

Fix image or text (No. 14-16). Freezing the image or text during distillation could greatly reduce the data performance and experiments show that learning text is more critical for the distillation. It is surprising that on Flickr30k, the experiment that only learns the similarity matrix (No. 16) could

Table 5. Various ablation studies with 200 pairs on Flickr30k (Plummer et al., 2015) and COCO (Lin et al., 2014).

No.	Model	Flickr						COCO					
		IR@1	IR@5	IR@10	TR@1	TR@5	TR@10	IR@1	IR@5	IR@10	TR@1	TR@5	TR@10
(1)	Similarity Mining+BCE	0.6	3.8	6.6	1.2	4.4	7.1	0.0	0.1	0.2	0.0	0.1	0.3
(2)	Similarity Mining+eNCE	4.1	13.4	20.8	5.5	17.3	26.3	1.3	5.1	9.0	1.9	7.1	11.7
(3)	Similarity Mining+wBCE	7.4	23.9	35.3	**15.1**	37.7	51.4	2.2	8.1	14.0	4.1	13.7	21.7
(4)	LoRS$_{\text{BCE}}$	0.1	0.6	1.0	0.2	0.7	1.2	0.0	0.1	0.2	0.1	0.7	1.5
(5)	LoRS$_{\text{eNCE}}$	8.2	**25.7**	**37.5**	13.0	36.3	51.0	2.0	7.7	13.0	3.9	13.9	22.2
(6)	LoRS$_{\text{wBCE}}$	**8.6**	25.3	36.6	14.5	**38.7**	**53.4**	**2.4**	**9.3**	**15.5**	**4.3**	**14.2**	**22.6**
(7)	LoRS$_{\text{wBCE}}$, $r = 1$	7.2	22.6	33.8	13.1	37.6	51.7	1.8	6.8	11.8	3.5	11.9	19.5
(8)	LoRS$_{\text{wBCE}}$, $r = 5$	**8.6**	25.3	36.6	14.5	**38.7**	**53.4**	1.6	6.8	12.1	2.9	10.6	17.7
(9)	LoRS$_{\text{wBCE}}$, $r = 10$	7.8	24.1	36.2	14.4	37.9	51.9	2.3	8.4	14.3	4.1	13.5	21.5
(10)	LoRS$_{\text{wBCE}}$, $r = 20$	7.9	24.0	35.7	13.7	37.8	50.2	**2.4**	**9.3**	**15.5**	**4.3**	**14.2**	**22.6**
(11)	LoRS$_{\text{wBCE}}$, $r = 40$	7.5	23.6	35.2	14.6	38.2	52.0	2.3	8.5	14.5	3.9	13.9	22.1
(12)	LoRS$_{\text{wBCE}}$ w/o L, R	7.4	23.4	36.0	13.9	38.1	52.2	2.3	8.3	14.0	3.8	13.6	21.9
(13)	LoRS$_{\text{wBCE}}$ w/o L, R, ω	0.2	1.3	2.5	2.8	10.4	17.4	0.1	0.2	0.5	0.7	3.1	5.3
(14)	LoRS$_{\text{wBCE}}$, fix image	4.9	16.3	24.9	9.8	29.4	41.1	1.4	5.3	9.3	1.9	6.6	11.7
(15)	LoRS$_{\text{wBCE}}$, fix text	4.3	15.5	25.2	9.2	24.7	36.3	0.6	2.9	5.3	1.5	5.8	9.9
(16)	LoRS$_{\text{wBCE}}$, fix image+text	1.9	8.2	13.9	4.3	12.8	19.9	0.6	2.5	4.7	1.2	5.2	8.8
(17)	CLIP Similarity	7.4	22.8	33.8	10.9	32.4	44.0	1.8	7.4	12.7	2.8	10.6	17.6



(a) Diagonal ω (b) Residual Similarity $LR^\top$

Figure 7. Synthesized similarity matrix $\tilde{S}$.

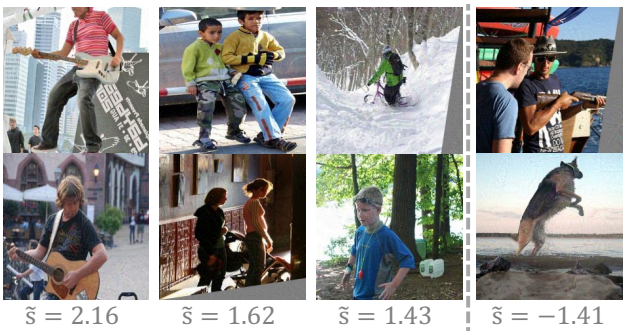


Figure 8. Sample pairs with different synthetic similarity $\tilde{s}$.

perform above random model.

Similarity from pretrained CLIP (No. 17). Instead of *learning* a similarity matrix, we directly compute the similarity matrix with a pretrained CLIP. However, the computed similarity matrix does not fit the distilled image and text, resulting in poor retrieval performance. This phenomenon is in line with the common conclusion in dataset distillation: the data that is suitable for network training may not be natural to humans.

4.7. Visualization

We visualize the image, text, and similarity matrix of 200 synthetic pairs for Flickr30k to present the distilled data.

Synthetic image and text. Fig. 6 shows the image and text before (initial) and after distillation. The images get DeepDream-style (Zeiler & Fergus, 2014), common in dataset distillation. The texts are retrieved by the closest caption in the train set to the distilled embeddings following (Wu et al., 2023). Appendix Sec. E gives more examples.

Learned similarity matrix For clarity, we separately show the diagonal and residual matrix in Fig. 7. Our method tends to learn large diagonal values since they are positive pairs. LoRS could also find the false negatives by learning certain similarity scores. We visualize some sample pairs with different synthetic similarities in Fig. 8. The samples that LoRS assigns large similarity values are also similar from a human perspective (left three pairs in the figure, with a similar person, background, etc.), which a regular CLIP model will erroneously regard as negative pairs.

5. Conclusions

In this work, we introduce Low-Rank Similarity Mining (LoRS) as an efficient solution for image-text dataset distillation. LoRS concurrently distills a ground truth similarity matrix with image-text pairs, leveraging low-rank factorization for efficiency and scalability. Our approach demonstrates a substantial improvement over existing algorithms. We advocate for the adoption of LoRS as a foundational synthetic data setup for image-text dataset distillation.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants No.62306175, No.62302296, National Key Research and Development Project of China (No.2022ZD0160102, No.2021ZD0110704).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning, especially multimodal learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Andonian, A., Chen, S., and Hamid, R. Robust cross-modal representation learning with progressive self-distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16430–16441, 2022.

Brock, A., De, S., Smith, S. L., and Simonyan, K. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning*, pp. 1059–1071. PMLR, 2021.

Cazenavette, G., Wang, T., Torralba, A., Efros, A. A., and Zhu, J.-Y. Dataset distillation by matching training trajectories. In *CVPR*, 2022.

Cazenavette, G., Wang, T., Torralba, A., Efros, A. A., and Zhu, J.-Y. Generalizing dataset distillation via deep generative prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3739–3748, 2023.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Cui, J., Wang, R., Si, S., and Hsieh, C.-J. Scaling up dataset distillation to imagenet-1k with constant memory. In *ICML*, pp. 6565–6590, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Deng, Z. and Russakovsky, O. Remember the past: Distilling datasets into addressable memories for neural networks. In *NeurIPS*, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Farahani, R. Z. and Hekmatfar, M. *Facility location: concepts, models, algorithms and case studies*. Springer Science & Business Media, 2009.

Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pp. 178–178. IEEE, 2004.

Gadre, S. Y., Ilharco, G., Fang, A., Hayase, J., Smyrnis, G., Nguyen, T., Marten, R., Wortsman, M., Ghosh, D., Zhang, J., et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024.

Gao, Y., Liu, J., Xu, Z., Wu, T., Liu, W., Yang, J., Li, K., and Sun, X. Softclip: Softer cross-modal alignment makes clip stronger. *arXiv preprint arXiv:2303.17561*, 2023.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, Y., Xiao, L., and Zhou, J. T. You only condense once: Two rules for pruning condensed datasets. *arXiv preprint arXiv:2310.14019*, 2023.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Kim, B., Choi, J., Lee, S., Lee, Y., Ha, J.-W., and Lee, J. On divergence measures for bayesian pseudocoresets. *arXiv preprint arXiv:2210.06205*, 2022a.

Kim, J.-H., Kim, J., Oh, S. J., Yun, S., Song, H., Jeong, J., Ha, J.-W., and Song, H. O. Dataset condensation via efficient synthetic-data parameterization. In *ICML*, 2022b.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lee, H. B., Lee, D. B., and Hwang, S. J. Dataset condensation with latent space knowledge factorization and sharing. *arXiv preprint arXiv:2208.10494*, 2022.

Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. C. H. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34: 9694–9705, 2021.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pp. 12888–12900. PMLR, 2022.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

Li, Y. and Li, W. Data distillation for text classification. *arXiv preprint arXiv:2104.08448*, 2021.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023a.

Liu, S., Wang, K., Yang, X., Ye, J., and Wang, X. Dataset distillation via factorization. In *NeurIPS*, 2022.

Liu, S., Ye, J., Yu, R., and Wang, X. Slimmable dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3759–3768, 2023b.

Liu, Y., Gu, J., Wang, K., Zhu, Z., Jiang, W., and You, Y. Dream: Efficient dataset distillation by representative matching. *arXiv preprint arXiv:2302.14416*, 2023c.

Loo, N., Hasani, R., Amini, A., and Rus, D. Efficient dataset distillation using random feature approximation. In *NeurIPS*, 2022.

Loo, N., Hasani, R., Lechner, M., and Rus, D. Dataset distillation with convexified implicit gradients. *arXiv preprint arXiv:2302.06755*, 2023.

Manousakas, D., Xu, Z., Mascolo, C., and Campbell, T. Bayesian pseudocoresets. In *NeurIPS*, 2020.

Nguyen, T., Chen, Z., and Lee, J. Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*, 2020.

Nguyen, T., Novak, R., Xiao, L., and Lee, J. Dataset distillation with infinitely wide convolutional networks. In *NeurIPS*, 2021.

Novack, Z., McAuley, J., Lipton, Z. C., and Garg, S. Chils: Zero-shot image classification with hierarchical label sets. In *International Conference on Machine Learning*, pp. 26342–26362. PMLR, 2023.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

Quattoni, A. and Torralba, A. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 413–420. IEEE, 2009.

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 *ICML*, 2021.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294, 2022.

Shin, D., Shin, S., and Moon, I.-C. Frequency domain-based dataset distillation. *arXiv preprint arXiv:2311.08819*, 2023.

Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., and Kiela, D. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15638–15650, 2022.

Srinivasa, R. S., Cho, J., Yang, C., Saidutta, Y. M., Lee, C.-H., Shen, Y., and Jin, H. Cwcl: Cross-modal transfer with continuously weighted contrastive loss. In *Thirty-seventh*

Conference on Neural Information Processing Systems, 2023.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

Tiwary, P., Shubham, K., Kashyap, V., et al. Constructing bayesian pseudo-coresets using contrastive divergence. *arXiv preprint arXiv:2303.11278*, 2023.

Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Wang, K., Zhao, B., Peng, X., Zhu, Z., Yang, S., Wang, S., Huang, G., Bilen, H., Wang, X., and You, Y. Cafe: Learning to condense dataset by aligning features. In *CVPR*, 2022.

Wang, T., Zhu, J.-Y., Torralba, A., and Efros, A. A. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Wang, Z., Xu, Y., Lu, C., and Li, Y.-L. Dancing with images: Video distillation via static-dynamic disentanglement. *arXiv preprint arXiv:2312.00362*, 2023.

Welling, M. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1121–1128, 2009.

Wu, X., Deng, Z., and Russakovsky, O. Multimodal dataset distillation for image-text retrieval. *arXiv preprint arXiv:2308.07545*, 2023.

Xu, Y., Li, Y.-L., Cui, K., Wang, Z., Lu, C., Tai, Y.-W., and Tang, C.-K. Distill gold from massive ores: Efficient dataset distillation via critical samples selection. *arXiv preprint arXiv:2305.18381*, 2023a.

Xu, Z., Chen, Y., Pan, M., Chen, H., Das, M., Yang, H., and Tong, H. Kernel ridge regression-based graph dataset distillation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2850–2861, 2023b.

Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., Liang, X., Li, Z., Jiang, X., and Xu, C. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021.

Yin, Z., Xing, E., and Shen, Z. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *arXiv preprint arXiv:2306.13092*, 2023.

Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.

Zhang, L., Zhang, J., Lei, B., Mukherjee, S., Pan, X., Zhao, B., Ding, C., Li, Y., and Xu, D. Accelerating dataset distillation via model augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11950–11959, 2023.

Zhao, B. and Bilen, H. Dataset condensation with differentiable siamese augmentation. In *ICML*, 2021.

Zhao, B. and Bilen, H. Synthesizing informative training samples with gan. *arXiv preprint arXiv:2204.07513*, 2022.

Zhao, B. and Bilen, H. Dataset condensation with distribution matching. In *WACV*, 2023.

Zhao, B., Mopuri, K. R., and Bilen, H. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020.

Zhao, G., Li, G., Qin, Y., and Yu, Y. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7856–7865, 2023.

Zhou, Y., Nezhadarya, E., and Ba, J. Dataset distillation using neural feature regression. *arXiv preprint arXiv:2206.00719*, 2022.

A. Dataset Variance

We compare the variance of image-text (multimodal) datasets and classification datasets to study why some dataset distillation algorithms fail on image-text datasets. We choose three multimodal datasets (Flickr30k, Visual Genome, and COCO-Caption) and seven image classification datasets (DTD, Caltech-101, MIT-indoor-67, CIFAR-10/100, iNaturalist-18 and ImageNet-1K). The images are forwarded to CLIP encoders with ResNet or ViT architecture, pretrained on YFCC100M (Thomee et al., 2016) or LAION-5B (Schuhmann et al., 2022). Note that we compute the intra-class variance for classification datasets. The full results are shown in Tab. 6 and plotted in Fig. 2. The variance of multimodal datasets is significantly larger than classification datasets.

Table 6. Comparison of dataset variance. CLS=classification dataset; MM=multimodal dataset.

Dataset	Type	Data Scale	YFCC100M ViT-B/32 ($\times 10^{-2}$)	YFCC100M ResNet50 ($\times 10^{-4}$)	LAION-5B ViT-B/32 ($\times 10^{-2}$)
Flickr30k (Plummer et al., 2015)	MM	29.0 K	9.082	15.635	12.744
Visual Genome (Krishna et al., 2017)	MM	108.2 K	9.543	17.832	13.578
COCO-Caption (Lin et al., 2014)	MM	113.3 K	9.588	17.656	13.652
DTD (Cimpoi et al., 2014)	CLS	5.6 K	5.657	7.289	9.052
Caltech101 (Fei-Fei et al., 2004)	CLS	9.1 K	4.411	7.256	6.833
MIT-indoor (Quattoni & Torralba, 2009)	CLS	15.6 K	5.043	8.940	7.918
CIFAR-10 (Krizhevsky et al., 2009)	CLS	50.0 K	3.823	4.991	7.759
CIFAR-100 (Krizhevsky et al., 2009)	CLS	50.0 K	3.750	4.663	7.498
iNaturalist-18 (Van Horn et al., 2018)	CLS	461.9 K	3.316	6.873	5.415
ImageNet-1K (Deng et al., 2009)	CLS	1281.2 K	5.079	6.880	7.937

B. Derivation of Loss Gradients

B.1. Proposition 3.1 and 3.2

Since we assume representations u_i and v_j are normalized, the cosine similarity $\tilde{s}_{ij} = u_i^\top v_j$. The $\mathcal{L}_{\text{eNCE}}$:

$$
\begin{aligned}
\mathcal{L}_{\text{eNCE}}(\mathcal{B}, S) &= -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m} s_{ij}\left[\log P_{ij}^{V} + \log P_{ij}^{T}\right] \\
&= -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m} s_{ij}\left[\log \frac{\exp(u_i^\top v_j/\tau)}{\sum_{k=1}^{m}\exp(u_i^\top v_k/\tau)} + \log \frac{\exp(u_i^\top v_j/\tau)}{\sum_{k=1}^{m}\exp(u_k^\top v_j/\tau)}\right] \\
&= \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m} s_{ij}\left[\log \sum_{k=1}^{m}\exp(u_i^\top v_k/\tau) + \log \sum_{k=1}^{m}\exp(u_k^\top v_j/\tau) - \frac{2}{\tau}u_i^\top v_j\right] \\
&\triangleq \frac{1}{m}(L_A + L_B - L_C).
\end{aligned}
\tag{12}
$$

Then we derive the gradient of each of the three components L_A, L_B, L_C:

$$
\begin{aligned}
\frac{\partial L_A}{\partial u_n} &= \frac{\partial}{\partial u_n}\sum_{i=1}^{m}\sum_{j=1}^{m} s_{ij}\log\sum_{k=1}^{m}\exp(u_i^\top v_k/\tau) = \frac{\partial}{\partial u_n}\sum_{j=1}^{m} s_{nj}\log\sum_{k=1}^{m}\exp(u_n^\top v_k/\tau) \\
&= \left(\frac{\partial}{\partial u_n}\log\sum_{k=1}^{m}\exp(u_n^\top v_k/\tau)\right)\left(\sum_{j=1}^{m} s_{nj}\right) \\
&= \left(\frac{\frac{\partial}{\partial u_n}\sum_{k=1}^{m}\exp(u_n^\top v_k/\tau)}{\sum_{k'=1}^{m}\exp(u_n^\top v_{k'}/\tau)}\right) s_{n\bullet} = \left(\frac{\sum_{k=1}^{m}\exp(u_n^\top v_k/\tau)v_k/\tau}{\sum_{k'=1}^{m}\exp(u_n^\top v_{k'}/\tau)}\right) s_{n\bullet} \\
&= \left(\sum_{k=1}^{m} P_{nk}^{V}\frac{v_k}{\tau}\right) s_{n\bullet},
\end{aligned}
\tag{13}
$$

$$\begin{aligned}\frac{\partial L_B}{\partial u_n} =& \frac{\partial}{\partial u_n}\sum_{i=1}^{m}\sum_{j=1}^{m} s_{ij}\log\sum_{k=1}^{m}\exp(u_k^\top v_j/\tau) = \sum_{i=1}^{m}\sum_{j=1}^{m} s_{ij}\left(\frac{\partial}{\partial u_n}\log\sum_{k=1}^{m}\exp(u_k^\top v_j/\tau)\right)\\ =& \sum_{i=1}^{m}\sum_{j=1}^{m} s_{ij}\left(\frac{\frac{\partial}{\partial u_n}\sum_{k=1}^{m}\exp(u_k^\top v_j/\tau)}{\sum_{k=1}^{m}\exp(u_k^\top v_j/\tau)}\right) = \sum_{i=1}^{m}\sum_{j=1}^{m} s_{ij}\left(\frac{\exp(u_n^\top v_j/\tau)v_j/\tau}{\sum_{k=1}^{m}\exp(u_k^\top v_j/\tau)}\right)\\ =& \sum_{i=1}^{m}\sum_{j=1}^{m} s_{ij}\left(P_{nj}^T\frac{v_j}{\tau}\right) = \left(\sum_{i=1}^{m} s_{ij}\right)\sum_{j=1}^{m}\left(P_{nj}^T\frac{v_j}{\tau}\right) = s_{\bullet j}\sum_{j=1}^{m}\left(P_{nj}^T\frac{v_j}{\tau}\right).\end{aligned} \tag{14}$$

$$\frac{\partial L_C}{\partial u_n} = \frac{\partial}{\partial u_n}\sum_{i=1}^{m}\sum_{j=1}^{m}\frac{2s_{ij}}{\tau}u_i^\top v_j = \frac{\partial}{\partial u_n}\sum_{j=1}^{m}\frac{2s_{nj}}{\tau}u_n^\top v_j = \sum_{j=1}^{m}\frac{2s_{nj}}{\tau}v_j. \tag{15}$$

The overall gradient of $\mathcal{L}_{\text{eNCE}}$ is:

$$\begin{aligned}\frac{\partial}{\partial u_n}\mathcal{L}_{\text{eNCE}}(\mathcal{B},S) =& \frac{1}{m}(L_A + L_B - L_C) = \frac{1}{m}\left(\sum_{k=1}^{m}P_{nk}^V\frac{v_k}{\tau}\right)s_{n\bullet} + \frac{1}{m}s_{\bullet j}\sum_{j=1}^{m}\left(P_{nj}^T\frac{v_j}{\tau}\right) - \frac{1}{m}\sum_{j=1}^{m}\frac{2s_{nj}}{\tau}v_j\\ =& \frac{1}{m\tau}\sum_{j=1}^{m}\left(P_{nj}^V s_{n\bullet} + P_{nj}^T s_{\bullet j} - 2s_{nj}\right)v_j.\end{aligned} \tag{16}$$

When the similarity matrix is identity, *i.e.* $S = I$, the gradient degrades to:

$$\frac{\partial}{\partial u_n}\mathcal{L}_{\text{NCE}}(\mathcal{B},S) = \frac{1}{m\tau}\sum_{j=1}^{m}\left(P_{nj}^V + P_{nj}^T - 2s_{nj}\right)v_j = \begin{cases}\frac{1}{m\tau}\sum_{j=1}^{m}\left(P_{nj}^V + P_{nj}^T - 2\right)v_j, & \text{if } n = j\\ \frac{1}{m\tau}\sum_{j=1}^{m}\left(P_{nj}^V + P_{nj}^T\right)v_j, & \text{if } n \neq j\end{cases} \tag{17}$$

B.2. Proposition 3.3

Similar to InfoNCE loss, the BCE loss

$$\mathcal{L}_{\text{BCE}}(\mathcal{B},S) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m}\ell\left(s_{ij},\sigma(\hat{s}_{ij}/\tau)\right) = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m}\ell\left(s_{ij},\sigma(u_i^\top v_j/\tau)\right), \tag{18}$$

where $\sigma(x)$ is the sigmoid function and $\ell(y,p) = -y\log(p) - (1-y)\log(1-p)$. We first have:

$$\begin{aligned}\ell\left(y,\sigma(x)\right) =& -y\log\frac{1}{1+e^{-x}} - (1-y)\log\frac{e^{-x}}{1+e^{-x}}\\ =& y\log(1+e^{-x}) + (1-y)x + (1-y)\log(1+e^{-x}) = \log(1+e^{-x}) + (1-y)x.\end{aligned} \tag{19}$$

Thus:

$$\frac{\partial\ell\left(y,\sigma(x)\right)}{\partial x} = \frac{-e^{-x}}{1+e^{-x}} + (1-y) = \sigma(x) - y. \tag{20}$$

Therefore the overall gradient of BCE is:

$$\frac{\partial}{\partial u_n}\mathcal{L}_{\text{BCE}}(\mathcal{B},S) = \frac{1}{m}\sum_{j=1}^{m}\frac{\partial}{\partial u_n}\ell\left(s_{nj},\sigma(u_n^\top v_j/\tau)\right) = \frac{1}{m\tau}\sum_{j=1}^{m}(\sigma(\tilde{s}_{nj}/\tau) - s_{nj})v_j. \tag{21}$$

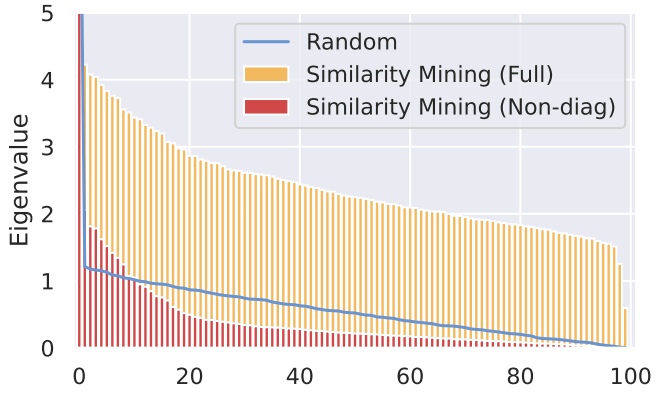


Figure 9. Eigenvalues of the distilled similarity matrix. The yellow bars are eigenvalues of the full similarity matrix; the red bars are that of the matrix without diagonal; the blue curve is the eigenvalues of a Gaussian noise matrix with the same mean and variance as the matrix of the red bars.

C. Justification of Low-Rank Technique

C.1. Low-Rank Nature of the Ideal Similarity Matrix

Given a image-text dataset on embedding space $\{u_i\}, \{v_i\}$ and a distance metric $d(\cdot,\cdot)$, the distance matrix is $D = \{d_{ij}\} = \{d(u_i, v_j)\}$. If two images embeddings u_i, u_j are similar, *i.e.*$d(x_i, x_j) < \epsilon$ where $\epsilon > 0$ is a small value, according to the triangle inequality, $\forall k$, $|d_{ik} - d_{jk}| = |d(u_i, v_k) - d(u_j, v_k)| \leq d(u_j, u_i) = \epsilon$. Hence the ith row and j^{th} row of the distance matrix are similar. As the similarity metric is always a function of the distance metric, we conclude that once similar samples exist, there are similar rows or columns in the similarity matrix, which leads to a low-rank similarity matrix.

C.2. Eigenvalue Analysis of Similarity Mining

We compute the eigenvalues of the fully learned similarity matrix and the matrix that zero-masked the diagonal. As shown in Fig. 9, the similarity without diagonal has much smaller eigenvalues. Compared to the random Gaussian matrix with the same mean and variance (blue), the residual similarity matrix has a more long-tailed eigenvalue distribution, indicating its low-rank nature.

D. Extended Ablation Study

D.1. Data Filtering

Data filtering (*e.g.* DataComp (Gadre et al., 2024)) has become a popular and efficient method to reduce the data scale for multimodal and image-text contrastive learning datasets, while it lacks comparison for dataset distillation studies. To enrich the discussion on image-text dataset distillation, we give an analysis and comparison of coreset selection, data filtering, and dataset distillation, as first summarized in Tab. 7.

Table 7. Comparison to of various data size reduction methods.

Method	From massive noisy data to high quality data	From high quality data to smaller data
Data filtering	**fast and good**	fast but worse
Coreset Selection	slow	good
Dataset Distillation	slower	**best**

Data filtering & Coreset selection. They share similar technical essences but are applied to different scenarios for different objectives of machine learning tasks. Data filtering is finding useful data from noisy large-scale internet/in-the-wild data, usually aiming to improve the data quality. Whilst, coreset selection focuses on smaller but more accurate datasets, only to enhance the training efficiency with a tolerable performance drop. Most coreset selection algorithms could be directly

Table 8. Comparison to data filtering methods. **Bold**: best method. Underlined: second best method.

DATASET	PAIRS	RESNET-50 YFCC100M	VIT-B/32 YFCC100M	VIT-B/32 LAION-5B	CORESET SELECTION	LORS
FLICKR30K	100	4.42	4.40	3.41	4.80	**27.38**
	200	6.95	6.26	6.41	6.52	**29.52**
	500	12.95	13.40	12.66	13.25	**31.58**
COCO	100	1.11	1.41	1.11	2.48	**9.37**
	200	2.20	2.63	2.00	3.52	**11.38**
	500	5.11	6.46	5.05	7.23	**13.45**

Table 9. Comparison of various architectures. LoRS consistently outperforms baselines.

IMAGE ENCODER	TEXT ENCODER	100 PAIRS				200 PAIRS			
		RAND	FILTER	MTT-VL	LORS	RAND	FILTER	MTT-VL	LORS
NFRESNET	BERT	4.1	6.3	16.0	**20.7**	5.1	7.1	17.9	**24.2**
VIT	BERT	2.9	4.3	11.5	**16.4**	2.6	4.1	12.6	**16.7**
NFREGNET	BERT	5.1	6.9	15.5	**22.0**	4.4	6.2	18.9	**26.2**
NFNET	DISTILBERT	7.4	10.7	24.0	**26.0**	6.2	10.6	24.4	**27.8**
NFNET	CLIP-TEXT	16.0	24.5	47.6	**54.9**	15.5	27.8	49.7	**60.1**

applied to data filtering, but are inefficient for the large-scale dataset, and most data filtering could be used for coreset selection but may have worse performance.

Coreset selection & Dataset distillation. The dataset distillation is an "upgraded" or learnable version of coreset selection, whose motivations and applications overlap. A fascinating point of dataset distillation is that it is possible to significantly reduce the data size but keep the performance even for a high-quality dataset, which is not possible for data filtering or selection.

Experimental analysis. To extend our comparison, we conduct experiments of data filtering. We adopt the CLIP/LAION score criterion in DataComp with 3 pretrained CLIP models (ResNet50 or ViT-B/32 image encoder; BERT text encoder; YFCC100M (Thomee et al., 2016) or LAION-5B (Schuhmann et al., 2022) pretraining), and the results are presented in Tab. 8. On the high-quality datasets, the data filtering method is comparable to coreset selection, while the dataset distillation method (LoRS) significantly outperforms both.

D.2. Network Architectures

Besides the NFNet+BERT model, we extend our experiments to more image and text encoder structures on Flickr30k (Plummer et al., 2015) in Tab. 9. LoRS surpasses baseline methods by a large margin, which shows generalization ability across various architectures.

Table 10. Time comparison. The results are in *second/iteration*.

DATASET	METHOD	100 PAIRS	200 PAIRS	500 PAIRS
FLICKR30K	BASELINE	6.47±0.27	6.59±0.48	6.43±0.32
	LORS	6.35±0.13	6.52±0.52	6.42±0.12
COCO	BASELINE	6.27±0.39	6.26±0.27	6.15±0.20
	LORS	6.00±0.14	5.89±0.07	5.92±0.13

D.3. Efficiency Analysis

Efficiency counts in dataset distillation. Memory and time cost are also one of the main motivations to leverage the low-rank method, as the storage of the similarity matrix should be considered as part of the synthetic dataset. Overall, the LoRS introduces negligible memory and time overhead compared to the large performance gain it brings. We analyze the efficiency of LoRS as follows.

- **Time**: we conduct experiments of distillation time comparison in Tab. 10. Due to the small parameter size of LoRS, it takes **comparable** time to the baseline method (the difference is smaller than the variance). Note that the synthetic data is optimized in batch so the data size (number of pairs) does not influence the iteration time.

Figure 10. More examples of synthetic images and retrieval text.

Table 11. Hyperparameters for different experiments.

DATASET	FLICKR			COCO		
PAIRS	100	200	500	100	200	500
LR: IMAGE	100	1000	1000	1000	1000	5000
LR: TEXT	100	1000	1000	1000	1000	5000
LR: LR	0.01	0.01	0.01	0.01	0.01	0.01
LR: SIMILARITY	10	10	100	5	50	500
INITIAL LR	0.1	0.1	0.1	0.1	0.1	0.1
BATCH SIZE	20	20	20	20	20	20
α	3	1	0.01	1	1	1
RANK r	10	5	20	10	20	20
MAX START EPOCH	2	2	3	2	2	2
SYNTH STEPS	8	8	8	8	8	8
EXPERT EPOCHS	1	1	1	1	1	1

- **Memory**: For a fair comparison, we have reduced the number of synthetic pairs only for LoRS in Tab. 2 3. So in these experiments, our method uses less memory storage but achieves significantly higher performance. And with the low-rank method, the memory overhead of LoRS is linear to the data size which is acceptable or even negligible. For example, with $r = 50$, the overhead of LoRS is 0.07% of the total data storage no matter the data scale.

E. More Visualizations of Synthetic Dataset

In Fig. 10, we provide more examples of image-text pairs of 200 synthetic pairs for Flickr30k to present the distilled data.

F. Hyper-parameters

We tune the hyper-parameters and list the values in Tab. 11. Many parameters of MTT are directly adopted from previous work (Wu et al., 2023).

G. Limitation

1. **Distillation of text**. Currently, we are only focusing on learning the synthetic text feature since direct distillation of text tokens is still under investigation and technically not feasible yet. We hope this issue can be addressed in future research.

2. **Storage of similarity matrix**. The similarity matrix takes additional storage. Though we have exploited the low-rank method to reduce the memory overhead of LoRS to negligibly linear complexity, there may be a trade-off between storage (the selection of rank r) and the performance which complicates the hyperparameter tuning.

3. **Loss design**. To use LoRS, the contrastive loss functions should be redesigned and chosen with empirical comparison.