GITD: A Generative Method for Multimodal Dataset Distillation

Shuo Wu[1], Songlin Jiang[1], and Guanjie Wang[1]

[1]Shanghai Jiao Tong University

ABSTRACT

The rapid advancement of machine learning has led to the emergence of large-scale models and massive datasets, increasing the demand for efficient data distillation techniques. This paper proposes GITD (Generative Image-Text Distillation), an innovative generative architecture for multimodal dataset distillation. GITD achieves superior text-image alignment, enhanced performance, and reduced overhead compared to traditional methods. Experimental results demonstrate significant performance improvements, with approximately 15% and 30% gains in image and text retrieval tasks, respectively. Additionally, GITD exhibits strong cross-architecture generalization, making it highly adaptable across different models.

Keywords: Dataset Distillation, Multimodal, Generate-filter

1. INTRODUCTION

The rapid development of machine learning has led to the emergence of many large-scale models and massive data sets. The accompanying increase in computing and storage requirements has brought great challenges to researchers. In order to better scale the data to properly reduce the training cost and overhead, people start to use data distillation to obtain more concise but still effective data. So far, single-modal data distillation of text,[1] images,[2] and videos[3] has made enough progress. Methods that are sufficiently general and efficient have been widely used. However, we should also see that with the emergence and rapid development of large language models, the importance of multimodal models and multimodal data has been gradually brought to the fore, and large-scale multimodal data sets have been continuously applied to experiments. There is a growing need for a general data distillation method that works well for multiple modalities.

However, existing data distillation techniques have largely been restricted to cater to image datasets, primarily due to the amenable data-optimization in the continuous pixel-domain of images,[4] which is not easy to apply to multi-modal data. Compared with single-modal data distillation, the difficulty of multi-modal data distillation is mainly reflected in the correspondence between data. We take words and pictures as an example, the data information in pictures is often many and complex, and contains some information that is difficult to describe, while words can only provide a limited and one-sided description of pictures. Because of this, one image data can often correspond to multiple text data, and it is difficult to preserve this relationship during the data distillation process.

Even for some studies that have made great progress in the field of data distillation, there is no universal multi-modal data distillation method. For LoRS,[5] it only uses the traditional method, which needs different adaptation for different networks, lacks sufficient universality, and at the same time, the efficiency is low, which has a non-negligible additional overhead. And D4M,[6] although a more novel diffusion model is used, which is novel and has certain generalization, and although it only uses the labels of the images and does not distil multi-modal data, it still has certain reference value and can be used as a possible multi-modal distillation method.

In response to this challenge, this paper proposes GITD (Generative Image-Text Distillation), aiming to achieve lightweight and highly generalized generative distillation for multimodal dataset. In order to improve the generality and efficiency of the method, and realize the use of cross-model and cross-architecture, GITD innovatively adopts the generate-filter method for multi-modal data, so as to effectively distill the mixed multi-modal data of image and text. Different from the common gradient matching method, GITD uses the K-center method to cluster the data, which improves the efficiency of calculation. In addition, different from general text

data distillation to obtain text features, GITD distills the text and obtains the caption of the text, so that the distilled data is more flexible and realistic, and easier to train and use later. As a generative method, we can achieve general data distillation across architectures, and a general and broad method can be used to obtain the distilled results for different architecture model data requirements.

2. RELATED WORKS

2.1 Data distillation

Data distillation aims to get a small data set from a large data set. The existing algorithms can be divided into three main categories: (1) Matching by Meta-model matching. It uses inner and outer loop to optimize the data summary, and the optimized learning algorithm can be transferred to the original data set, which has a very high transferability. This algorithm was originally proposed in 2018 by Wang[7] , but the original version has a relatively high computational cost and loss. A separate line of work focuses on using Neural Tangent Kernel (NTK)[8] based algorithms to solve the inner-loop in closed form and KIP[9] uses the NTK of a fully-connected neural network for efficient data distillation. Recently, FREPO[10] divided the network into feature extractor and classifier for optimization, which achieved stronger scalability and generalization ability. (2) By gradient matching. Originally proposed by DC,[11] gradient matching method can avoid inner loop unrolling compared to method (1), thus making the overall optimization more efficient. DSA[12] improves DC by performing the same image enhancement and optimizing the formulation. This optimization is generic and applies to all frameworks. IDC[13] stores data at a lower resolution to eliminate spatial redundancy and upsample to the original scale when used. (3) By trajectory matching. MTT was proposed in 2022.[14] Subsequently TESLA[15] re-parameterizes the parameter-matching loss of MTT by using linear algebraic manipulations to make the bilevel optimization's memory complexity independent of N.

2.2 Generate-filter

Generate-filter refers to the generative experiment in which, after generating a certain number of candidate samples, the most useful samples are obtained as the final training data or input through secondary selection. This process can find relatively more representative and effective samples when the data is insufficient or the generated data is not completely reliable, so as to avoid the interference of low-quality data on the experimental results. Therefore, the key to Generate-filter is to find the most appropriate selection method and judgment method. D4M[6] uses K-Means method to perform Prototype Learning on the extracted features after extracting the features, so as to obtain the most representative prototypes, ignore those that are not representative, and finally generate new distilled data according to them. RDED[16] first divides the image into multiple small patches, finds the key patch with the highest score through the pre-trained observer model, and then selects a certain number of key patches to spell out new images, and generates region-level soft labels for these images.

3. METHOD

In this section, we propose GITD (generative image-text distillation) to achieve low cost, lightweight, and highly generalized generative image-text dataset distillation. We first introduce the necessary preliminaries in Sec. 3.1, and then provide a detailed explanation of GITD in Sec. 3.2.

3.1 Preliminaries

3.1.1 Latent Diffusion Models (LDM)

Diffusion models are a class of generative models that learn to generate data by gradually denoising a normally distributed variable. The process consists of two main phases: the forward diffusion process and the reverse denoising process. Latent Diffusion Models (LDM)[17] enhance the efficiency of traditional diffusion models by operating in a lower-dimensional latent space rather than the high-dimensional data space. This is achieved by first encoding the data into a latent representation using an encoder E, and then applying the diffusion process in this latent space.

Latent Space Encoding. Given a data point x_0, it is first encoded into a latent representation z_0 using an encoder E:

$$z_0 = E(x_0)$$

The diffusion process is then applied to z_0 instead of x_0, significantly reducing computational complexity.

Forward Diffusion Process. Given a data point x_0 sampled from the data distribution $q(x_0)$ and a prompt y, the forward process gradually adds Gaussian noise to x_0 over T timesteps, producing a sequence of noisy samples $x_1, x_2, \ldots, x_T$. This process is defined as

$$q(x_t|x_{t-1}, y) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$$

where β_t is the noise schedule at timestep t, and $\mathcal{N}$ denotes the Gaussian distribution. The prompt y is used as a conditioning signal throughout the process.

Reverse Denoising Process. The reverse process learns to iteratively denoise x_T back to x_0 by estimating the noise added at each timestep, conditioned on the prompt y. This is typically parameterized by a neural network ϵ_θ:

$$p_\theta(x_{t-1}|x_t, y) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t, y), \Sigma_\theta(x_t, t, y))$$

where μ_θ and Σ_θ are the mean and covariance predicted by the model, now explicitly conditioned on the prompt y.

Training Objective. The model is trained to minimize the variational lower bound (VLB) on the negative log-likelihood, which simplifies to predicting the noise added at each timestep, conditioned on the prompt y:

$$\mathcal{L} = \mathbb{E}_{x_0,\epsilon,t,y}\left[\|\epsilon - \epsilon_\theta(x_t, t, y)\|^2\right]$$

where ϵ is the actual noise added during the forward process, and y is the prompt used for conditioning.

Decoding to Data Space. After the reverse process, the denoised latent representation z_0 is decoded back to the data space using a decoder D

$$x_0 = D(z_0)$$

3.1.2 CLARA Clustering

CLARA (Clustering Large Applications)[18] is an efficient clustering algorithm designed for large datasets, extending the PAM (Partitioning Around Medoids)[18] method. It selects actual data points, called medoids, as cluster centers, which are more robust to outliers compared to centroids. CLARA improves scalability by employing a sampling strategy: it randomly selects multiple subsets S from the dataset X, applies PAM to each subset to find medoids, and assigns the entire dataset to the best set of medoids. The objective is to minimize the total clustering cost, defined as:

$$\text{Cost}(M) = \sum_{i=1}^{n} \min_{m_j \in M} d(x_i, m_j)$$

where M is the set of medoids, $d(x_i, m_j)$ is the dissimilarity between data point x_i and medoid m_j, and n is the number of data points. By focusing on subsets, CLARA significantly reduces computational complexity while maintaining clustering quality.

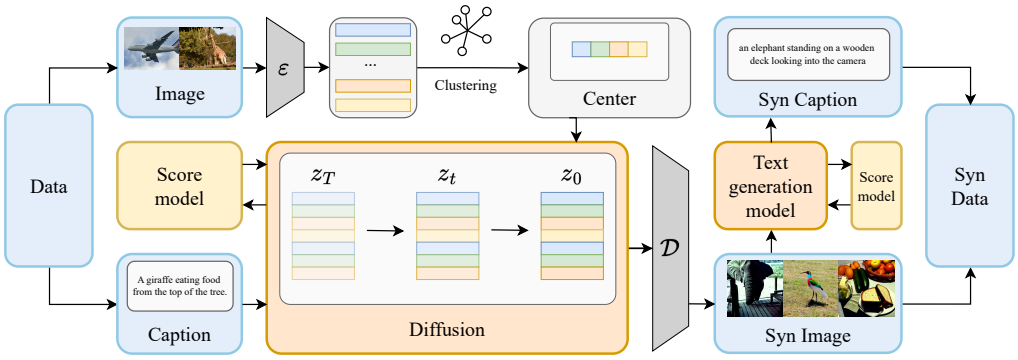

Figure 1. Pipeline of GITD (generative image-text distillation).

3.2 GITD (Generative Image-Text Distillation)

Based on the previous analysis, existing works have the disadvantages of high operating costs, weak generalization, and lack of support for multimodal data distillation. To solve these deficiencies, we propose GITD (generative image-text distillation), which is a low-cost, lightweight, and highly generalized generative architecture for multimodal dataset distillation.

Fig. 1 shows the pipeline of GITD. Given a multimodal dataset, GITD firstly extracts latent embeddings from original images. Then, GITD applies CLARA Clustering algorithm to find the most representative medoids of the latent embeddings. Subsequently, using the original corresponding text as the prompt, GITD performs the diffusion process with pre-trained diffusion model on the latent embeddings of the cluster centers and decodes the generated embeddings to generate synthetic images. Finally, GITD leverages pre-trained text generation model to generate the corresponding synthetic text for synthetic images. In the following paragraphs, we will elaborate on the various parts of the pipeline.

Latent Embedding Extraction. GITD leverages the encoder ε inherent in the pre-trained LDM to map the original image data into the latent embedding space. The latent embeddings capture the essential features of the images, enabling efficient clustering and generation.

CLARA Clustering for Medoid Selection. To select the most representative samples in the latent space, GITD applies k-medoids clustering algorithm. To improve the clustering efficiency, GITD employs the CLARA algorithm, enabling the pipeline to perform clustering efficiently on larger datasets. The selected medoids, serve as the foundation for generating synthetic data, ensuring diversity and coverage of the dataset.

Image Generation with LDM. From the latent embeddings of the cluster medoids, GITD conducts a diffusion process to generate high-quality synthetic embeddings, which is guided by the pre-trained LDM (e.g. `stable-diffusion-v1-5`[19]). The original text associated with each medoid serves as the prompt to condition the diffusion process, ensuring alignment between the generated embeddings and the semantic content of the original data. Then GITD utilizes the decoder $\mathcal{D}$ that corresponds to the encoder in the LDM to generate synthetic images from the synthetic embeddings.

Text Generation. To complete the generation of the synthetic mutimodal dataset, GITD utilizes the pre-trained text generation model (e.g. `blip2-opt-6.7b-coco`[20]) to produce corresponding captions of the synthetic images. The synthetic text and synthetic images together form a synthetic multimodal dataset.

Score Model for Image&Text Generation. To further increase the quantity of generated images and text, GITD introduces a multimodal pre-trained model (e.g. `clip-vit-base-patch32`[21]) as the score model. As illustrated in the Fig. 1, during the diffusion-based image generation process, multiple images are generated and then evaluated by the score model. The score model takes the generated images and their corresponding original texts as input, encodes them into embeddings, and computes the similarity between these embeddings. A higher similarity results in a higher score. Finally, the generated image with the highest score is selected as the final output. Similarly, during the text generation phase, GITD generates multiple text candidates and evaluate

them using the score model based on their corresponding synthetic images. The text with the highest score is selected as the final synthetic text. By leveraging the score model, GITD enhances the quality and alignment of the synthetic dataset.

4. EXPERIMENTS

In this section, we first describe the setup in Sec. 4.1. We use it to evaluate our generative image-text dataset distillation performance. We then compare the performance of our methods to baseline and verify the superiority of our method in Sec. 4.2. In Sec. 4.3, we demonstrate the capabilities of our method on cross-architecture. We further conduct the ablation study in Sec. 4.4, and visualize the synthetic image-text pair examples in Sec. 4.5.

4.1 Setup

Datasets and Tasks. We apply our method on image-text retrieval tasks. We evaluate our method on multimodal datasets: Flickr30K[22] and COCO,[23] which are widely used for image-text retrieval tasks.

Evaluation models. We train the evaluation models on the synthetic multimodal dataset. In comparison experiments, we use pre-trained NFNet[24] as image model and Bert[25] as text model. In cross-architecture Experiments, we further explore ResNet50,[26] RegNet,[27] NF_ResNet50,[28] NF_RegNet[29] as other image models and CLIP[21] as other text model.

Evaluation metrics. We use Recall R@K (for K $\in \{1, 5, 10\}$), which represents the fraction of relevant items that are returned within the top K results of image-text retrieval task, as the evaluation metrics in experiments. More specifically, we use IR@K (Image Recall) and TR@K (Text Recall) to evaluate. The metrics are measured on the evaluation model.

Baselines. We employ the following coreset selection methods as baselines: Random (random select a data subset), Herd,[30] K-center[31] and Forgetting.[32]

Testbed. All the experiments are conducted on a single NVIDIA RTX 4070 GPU, which is quite lightweight.

4.2 Comparison Results

The results on Flickr30k and COCO are shown in Table 1. Compared to baselines, GITD demonstrates superior performance across almost all data scales and datasets. In particular, GITD outperforms the baselines with an relative improvement of around $\sim 15\%$ in TR and $\sim 30\%$ in IR, while the maximum relative improvement can reach up to 73.3%. This demonstrates the effectiveness of GITD.

Table 1. Results on Flickr30K and COCO. The data is evaluated on NFNet+Bert. R, H, K, and F denote the baseline methods: Random (random subset selection), Herding, K-center, and Forgetting, respectively.

Dataset	#pairs	Metrics	TR					IR				
			R	H	K	F	GITD	R	H	K	F	GITD
Flickr30K	100	R@1	1.3	1.1	0.6	1.2	**1.7**	1.0	0.7	0.7	0.7	**1.1**
		R@5	**5.9**	4.7	5.0	4.2	**5.9**	4.0	2.8	3.1	2.4	**4.2**
		R@10	10.1	7.9	7.6	9.7	**10.3**	6.5	5.3	6.1	5.6	**8.0**
	200	R@1	2.1	2.3	2.2	1.5	**2.7**	1.1	1.5	1.5	1.2	**1.6**
		R@5	8.7	8.4	8.2	8.4	**9.3**	4.8	5.5	5.4	3.1	**6.5**
		R@10	13.2	14.4	13.5	10.2	**15.7**	9.2	9.3	9.9	8.4	**10.9**
COCO	100	R@1	0.8	0.8	**1.4**	0.7	1.1	0.3	0.5	0.4	0.3	**0.6**
		R@5	3.0	2.1	3.7	2.6	**3.8**	1.3	1.4	1.4	1.5	**2.6**
		R@10	5.0	4.9	5.5	4.8	**6.6**	2.7	3.5	2.5	2.5	**4.8**
	200	R@1	1.0	1.0	1.2	1.1	**1.4**	0.6	0.9	0.7	0.6	**1.0**
		R@5	4.0	3.6	3.8	3.5	**5.3**	2.3	2.4	2.1	2.8	**3.6**
		R@10	7.2	7.7	7.5	7.0	**8.9**	4.4	4.1	5.8	4.9	**6.5**

4.3 Cross-Architecture Experiments

In the cross-architecture experiments, We choose MTT-VL[33] and LoRS,[5] two works that have demonstrated excellent performance on multimodal datasets, as baselines. The results of cross-architecture experiments on different methods are shown in Table 2. The baselines demonstrate a significant performance drop in cross-architecture evaluation, with performance generally declining to around 50% of its original performance. In contrast, GITD maintains the performance without noticeable degradation and even achieves performance improvement on TR@5 with NF-RegNet, which the baselines fail to accomplish. This demonstrates that GITD transfers exceptionally well across different models compared to the baseline.

Table 2. The results of cross-architecture experiments for different methods. For MTT-VL and LoRS, the datasets are synthesized using the NFNet+Bert model. The evaluation is conducted on the COCO dataset, with Bert as the text model. For each method, the performance evaluated on NFNet+Bert serves as the 100% baseline for each R@K metric.

Method	Evaluate	TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10
MTT-VL	NF-ResNet50	52.5%	51.9%	54.2%	95.7%	87.9%	86.2%
	NF-RegNet	36.4%	34.3%	39.6%	53.2%	54.8%	86.2%
LoRS	ResNet50	33.0%	43.9%	49.0%	43.9%	49.2%	57.9%
	RegNet	35.0%	43.6%	50.7%	43.9%	52.3%	56.2%
GITD	NF-ResNet50	88.2%	83.1%	76.7%	72.7%	81.0%	77.5%
	NF-RegNet	112%	114%	91.3%	63.6%	90.1%	81.3%

To further explore the cross-architecture generalization capability of GITD, We evaluate its performance on different text and image evaluation models. The results are presented in Table 3. Across different image models, GITD demonstrates consistent performance, while for text models, GITD significantly outperforms on CLIP compared to Bert. This is attributed to the fact that the pre-trained CLIP model is substantially more powerful than Bert, while the capabilities of NFNet, NF_ResNet50 and NF_RegNet are relatively similar. These results indicate that GITD exhibits strong cross-architecture generalization capabilities for both text and image modalities.

Table 3. The results of cross-architecture experiments for GITD. The evaluation is conducted on the COCO dataset with 100 pairs.

Text Model	Vision Model	TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10
BERT	NFNet-l0	1.7	5.9	10.3	1.1	4.2	8.0
	NF_ResNet50	1.5	4.9	7.9	0.8	3.4	6.2
	NF_RegNet	1.9	6.7	9.4	0.7	3.8	6.5
CLIP	NFNet-l0	6.3	18.7	27.9	5.0	15.9	23.8
	NF_ResNet50	3.5	10.9	15.3	2.8	8.6	14.2
	NF_RegNet	4.8	14.3	23.0	4.1	12.9	20.6

4.4 Ablation Study

To measure the effectiveness of each proposed method of GITD, we create several variants of GITD as follows to conduct ablation study. 1) For GITD/C, the CLARA clustering is replaced by randomly selected embeddings; 2) GITD/D skips the diffusion process and directly select the original images corresponding to the cluster centers as the synthetic images; 3) GITD/T skips the text generation process and directly select the original text corresponding to the cluster centers as the synthetic texts; 4) GITD/S removes the score model, and the first generated image-text pair is directly used as the synthetic image-text pair.

Table 4. The performance of different variants of GITD. The evaluation is conducted on the Flickr30K dataset with 100 pairs. The evaluation model is NFNet+Bert.

GITD Variants	TR			IR		
	R@1	R@5	R@10	R@1	R@5	R@10
GITD	1.7	5.9	10.3	1.1	4.2	8.0
GITD/C	1.0	5.0	8.8	0.6	3.7	6.5
GITD/D	1.4	5.4	9.3	0.8	3.9	6.8
GITD/T	1.7	5.8	10.2	0.9	3.4	6.8
GITD/S	1.7	5.7	9.7	1.1	4.2	7.3

Table 4 presents the results of the ablation study. All variants exhibit a decline in performance, indicating the importance of each component in GITD.

4.5 Visualization

To illustrate the quality of the synthetic data intuitively, We provide visualizations of the synthetic image-text pairs for Flickr30K. As shown in Fig. 2, the generated images are highly realistic, and the captions are precisely aligned with the visual content, demonstrating the superior quality of the synthetic dataset.

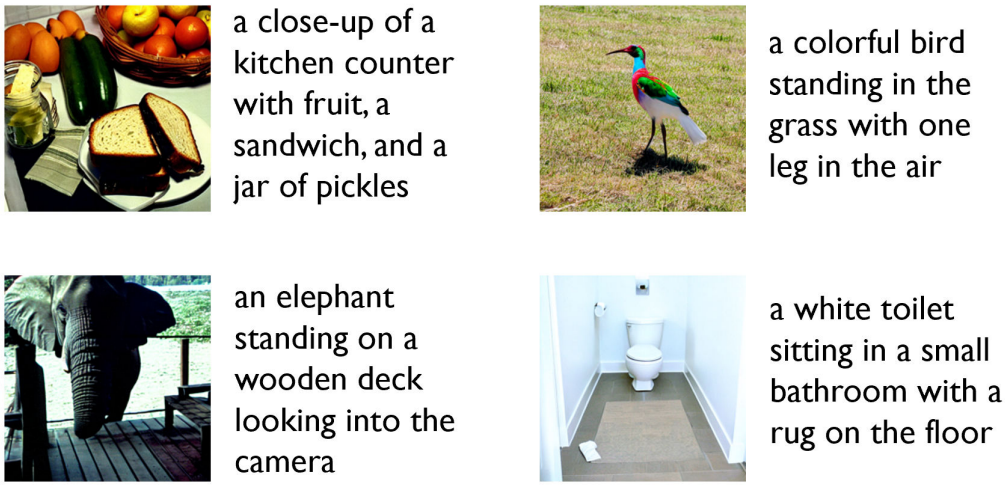

Figure 2. Examples of synthetic image-text pairs.

5. DISCUSSIONS AND LIMITATIONS

Due to the generative method employed by GITD, it demonstrates strong generalization capabilities without relying on specific architectures, making it directly applicable to a wide range of models. Additionally, GITD does not require any specific training, resulting in low time cost and overhead. Furthermore, by using the diffusion process, text generation model and score model, the generated synthetic multimodal dataset achieves enhanced realism and improved alignment.

However, the performance of GITD is relatively insufficient compared to the most common traditional methods (such as gradient matching, trajectory matching). One reason for this deficiency is the lack of training on specific dataset. If the diffusion models and text generation models can be fine-tuned on the dataset to be distilled, the performance can be much better. The other reason is that GITD only leverages the information of selected centre during data generation with generative methods, while the traditional methods use the information of whole original dataset.

Meanwhile, compared to the improvement of IR, the improvement of TR is relatively poor. On the one hand, the task of text distillation itself is quite challenging; On the other hand, the generation of text in GITD does not directly utilize the original caption in the dataset, which may lead to lower performance.

6. CONCLUSION

In the end, this work innovatively proposes GITD, a novel generative architecture for multimodal dataset distillation. GITD demonstrates better text-image alignment, better performance and lower overhead. Experiments show that GITD achieves significant improvements (approximately 15% on IR and 30% on TR) compared to traditional coreset selection methods. Meanwhile, GITD transfers exceptionally well across different models compared to the traditional dataset distillation methods, which shows strong cross-architecture generalization capabilities of GITD. We hope that this novel generative multimodal dataset distillation method can be more widely applied to the training of cross-modal large datasets.

REFERENCES

[1] Li, Y. and Li, W., "Data distillation for text classification," *arXiv preprint arXiv:2104.08448* (2021).

[2] 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*], 2850–2861 (2023).

[3] Deng, J., Pan, Y., Yao, T., Zhou, W., Li, H., and Mei, T., "Relation distillation networks for video object detection," in [*Proceedings of the IEEE/CVF international conference on computer vision*], 7023–7032 (2019).

[4] Sachdeva, N. and McAuley, J., "Data distillation: A survey," *arXiv preprint arXiv:2301.04272* (2023).

[5] Xu, Y., Lin, Z., Qiu, Y., Lu, C., and Li, Y.-L., "Low-rank similarity mining for multimodal dataset distillation," *arXiv preprint arXiv:2406.03793* (2024).

[6] Su, D., Hou, J., Gao, W., Tian, Y., and Tang, B., "Dˆ 4: Dataset distillation via disentangled diffusion model," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 5809–5818 (2024).

[7] Wang, T., Zhu, J.-Y., Torralba, A., and Efros, A. A., "Dataset distillation," *arXiv preprint arXiv:1811.10959* (2018).

[8] Jacot, A., Gabriel, F., and Hongler, C., "Neural tangent kernel: Convergence and generalization in neural networks," *Advances in neural information processing systems* **31** (2018).

[9] Nguyen, T., Chen, Z., and Lee, J., "Dataset meta-learning from kernel ridge-regression," *arXiv preprint arXiv:2011.00050* (2020).

[10] Zhou, Y., Nezhadarya, E., and Ba, J., "Dataset distillation using neural feature regression," *Advances in Neural Information Processing Systems* **35**, 9813–9827 (2022).

[11] Zhao, B., Mopuri, K. R., and Bilen, H., "Dataset condensation with gradient matching," *arXiv preprint arXiv:2006.05929* (2020).

[12] Zhao, B. and Bilen, H., "Dataset condensation with differentiable siamese augmentation," in [*International Conference on Machine Learning*], 12674–12685, PMLR (2021).

[13] 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 [*International Conference on Machine Learning*], 11102–11118, PMLR (2022).

[14] Cazenavette, G., Wang, T., Torralba, A., Efros, A. A., and Zhu, J.-Y., "Dataset distillation by matching training trajectories," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 4750–4759 (2022).

[15] Cui, J., Wang, R., Si, S., and Hsieh, C.-J., "Scaling up dataset distillation to imagenet-1k with constant memory," in [*International Conference on Machine Learning*], 6565–6590, PMLR (2023).

[16] Sun, P., Shi, B., Yu, D., and Lin, T., "On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 9390–9399 (2024).

[17] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B., "High-resolution image synthesis with latent diffusion models," in [*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*], 10684–10695 (2022).

[18] Kaufman, L. and Rousseeuw, P. J., [*Finding groups in data: an introduction to cluster analysis*], John Wiley & Sons (2009).

[19] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B., "High-resolution image synthesis with latent diffusion models," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*], 10684–10695 (June 2022).

[20] Li, J., Li, D., Savarese, S., and Hoi, S., "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in [*International conference on machine learning*], 19730–19742, PMLR (2023).

[21] 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*], 8748–8763, PMLR (2021).

[22] 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*], 2641–2649 (2015).

[23] 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*], 740–755, Springer (2014).

[24] Brock, A., De, S., Smith, S. L., and Simonyan, K., "High-performance large-scale image recognition without normalization," in [*International conference on machine learning*], 1059–1071, PMLR (2021).

[25] Devlin, J., "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805* (2018).

[26] 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*], 770–778 (2016).

[27] 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*], 10428–10436 (2020).

[28] Brock, A., De, S., and Smith, S. L., "Characterizing signal propagation to close the performance gap in unnormalized resnets," *arXiv preprint arXiv:2101.08692* (2021).

[29] Xu, J., Pan, Y., Pan, X., Hoi, S., Yi, Z., and Xu, Z., "Regnet: Self-regulated network for image classification," *IEEE Transactions on Neural Networks and Learning Systems* **34**(11), 9562–9567 (2022).

[30] Welling, M., "Herding dynamical weights to learn," in [*Proceedings of the 26th annual international conference on machine learning*], 1121–1128 (2009).

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

[32] 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).

[33] Wu, X., Deng, Z., and Russakovsky, O., "Multimodal dataset distillation for image-text retrieval," *arXiv preprint arXiv:2308.07545* (2023).