OPT2023: 15th Annual Workshop on Optimization for Machine Learning

Multi-head CLIP: Improving CLIP with Diverse Representations and Flat Minima

Mo Zhou[*] MOZHOU@CS.DUKE.EDU
Duke University

Xiong Zhou XIONGZHO@AMAZON.COM
Li Erran Li LILIMAM@AMAZON.COM
AWS AI, Amazon

Stefano Ermon ERMON@CS.STANFORD.EDU
Stanford University; AWS AI, Amazon

Rong Ge RONGGE@CS.DUKE.EDU
Duke University

Abstract

Contrastive Language-Image Pre-training (CLIP) has shown remarkable success in the field of multimodal learning by enabling joint understanding of text and images. In this paper, we introduce a novel method called Multi-head CLIP, inspired by Stein Variational Gradient Descent (SVGD) and Sharpness-aware Minimization (SAM). Our approach aims to enhance CLIP's learning capability by encouraging the model to acquire diverse features while also promoting convergence towards a flat loss region, resulting in improved generalization performance. We conduct extensive experiments on two benchmark datasets, YFCC15M and CC3M, to evaluate the effectiveness of our proposed method. The experimental results consistently demonstrate that multi-head CLIP outperforms both the original CLIP architecture and CLIP with the SAM optimizer.

1. Introduction

In recent years, the domain of multimodal representation learning, particularly in the context of natural images and text, has gained significant attention from researchers. A notable breakthrough in this field is the Contrastive Language-Image Pre-training (CLIP) framework, which employs contrastive learning techniques from self-supervised learning to pretrain vision-language models [27]. By leveraging large amounts of unlabeled data, CLIP enables the model to learn powerful joint representations of images and text. As a result, CLIP has exhibited impressive performance across various downstream tasks, even in the zero-shot setting.

Given the simplicity and the immense potential of CLIP, researchers have invested significant efforts to enhance and refine the framework of CLIP from various perspectives. For example, [17] and [23] have introduced the self-supervision of data to improve the utilization of training data. [15] proposed a new encoder-decoder architecture and a dataset bootstrapping method to enhance data quality. Others have proposed fine-grained representations alignment [29] and new loss function [6] to improve the performance.

[*] Work done during internship at AWS AI, Amazon.

In this work, we present a new perspective aimed at extending the existing CLIP framework. Rather than restricting the model to learn a single representation for each piece of data, our approach introduces the concept of learning multiple representations. We propose that learning diverse representations within a flat loss region may capture more comprehensive information, potentially leading to improved performance in downstream tasks. Our method, called multi-head CLIP, is designed to learn diverse representations simultaneously, drawing inspirations from two popular algorithms: Stein Variational Gradient Descent (SVGD) [19] and Sharpness-aware minimization (SAM) [5]. Similar to SVGD, we incorporate a diverse regularization term to encourage distinct representations for the same data. At the same time, building on the ideas of SAM, we encourage the model to converge to a flat minima by constraining the parameters of different representation to remain close to each other. Note that flatness of loss in the parameter space does not contradict with diversity of representations, as different representations may have very similar loss – our approach aims to find regions in parameter space where there are diverse representations all achieving similarly low loss. To evaluate the effectiveness of our proposed multi-head CLIP, we conduct experiments comparing its performance to the original CLIP framework and CLIP trained with SAM. The results demonstrate that Multi-head CLIP achieves superior zero-shot performance, suggesting that the multiple learned representations are more beneficial for downstream tasks than a single representation.

2. Method

In this section, we first briefly review the original CLIP method and then present our proposed new method multi-head CLIP.

2.1. Background: Contrastive Language-Image Pre-training (CLIP)

CLIP is a contrastive learning method that tries to learn representations for image and text data [27]. It first maps the image and text data into different embeddings in the space embedding space using the corresponding image and text encoders. Then, it uses the InforNCE loss on top of these embeddings to encourage the image and text embeddings from the same pair of data to stay close (positive pair) and keep others to stay away (negative pair).

CLIP shows remarkable ability of learning representations that could enable great zero-shot learning performance. For example, when pretraining with large number of image-text pair data, it performs well on image classification task without training on the specific dataset (e.g., ImageNet).

2.2. Our Method: Multi-head CLIP

CLIP provides a single representation for each data through image/text encoder. Intuitively, having multiple representations per data can capture more information than a single representation by capturing different perspective of the data. Such multiple representations also give the flexibility to use for downstream tasks and could improve the performances. In this paper, we propose a new method called multi-head CLIP that is able to provide multiple diverse representations for each data and allow better performance on several benchmarks.

Given a dataset $\mathcal{D} = \{(x_i^I, x_i^T)\}_{i=1}^n$, our goal is to learn useful representations for each image-text pair (x_i^I, x_i^T) using neural networks $f^I(x_i^I; \theta^I)$ and $f^T(x_i^T; \theta^T)$. In particular, for a given input x, image encoder $f^I(x; \theta^I)$ outputs m representations $\{f^I(x; \theta_i^I)\}_{i=1}^m$ (similar for text encoder

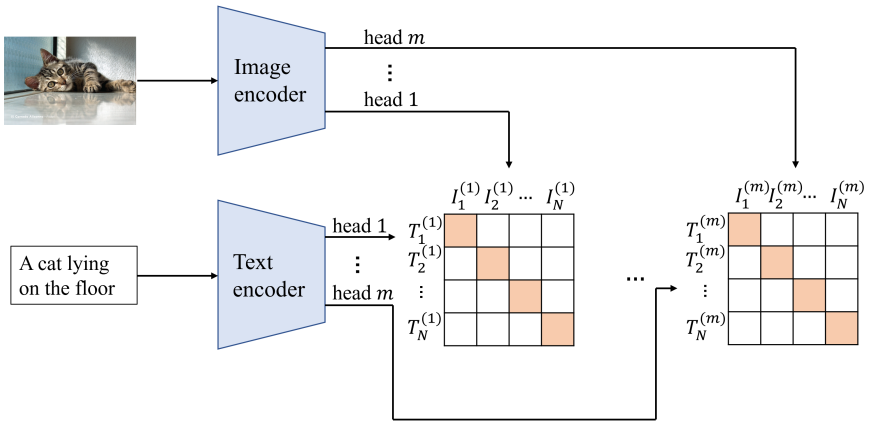

Figure 1: Our proposed multi-head CLIP method. Both image encoder and text encoder output m representations for each image-text pair ($m = 1$ becomes the original CLIP model). The objective consists 3 parts as shown in (1): loss function $L(\theta)$, diversity regularization R_{div} and closeness regularization R_{close}.

$f^T(x;\theta)$), where $\theta^I = (\theta_1^I, \ldots, \theta_m^I)$ and $\theta^T = (\theta_1^T, \ldots, \theta_m^T)$. Here we use the same architecture but allow different parameters for these m heads of the output.

Our method aims to minimize the following objective with such multi-head network f^I and f^T:

$$\min_{\{\theta_i^I\},\{\theta_i^T\}} \sum_i L(\theta_i^I, \theta_i^T) + \lambda_{div} \sum_{M\in\{I,T\}} R_{div}(\theta^M) + \lambda_{close} \sum_{M\in\{I,T\}} R_{close}(\theta^M), \tag{1}$$

where $L(\theta)$ is the loss function, R_{div} is the diversity regularization, R_{close} is the closeness regularization and λ_{div} and λ_{close} are regularization coefficient to control the strength of regularization effects. Note that when $m = 1$, our method goes back to the original CLIP method. Also, to improve the parameter efficiency, the different head of f^I and f^T will share the parameters in lower layers and only keep the top layers' parameters different. See Figure 1 for an illustration.

For loss function, we use the common contrastive objective in self-supervised learning [26], following the same setup in CLIP. Since we minimize the sum of loss over all representations, this ensures the loss is small across all representations so that the quality of all representations are good.

The diversity term R_{div} encourages the representation learned by each head to be different so that multiple representations can capture more diverse information than a single representation. Specifically, we use the norm to measure the difference between representations:

$$R_{div}(\theta_1, \ldots, \theta_m) = -\sum_{i<j} \left\| f_{\theta_i} - f_{\theta_j} \right\|_2^2, \tag{2}$$

where $\left\| f_{\theta_i} - f_{\theta_j} \right\|_2^2 = (1/n) \sum_x \|f(x;\theta_i) - f(x;\theta_j)\|_2^2$ is used to approximate the ideal norm $\mathbb{E}_x[\|f(x;\theta_i) - f(x;\theta_j)\|_2^2]$. Note that in CLIP, the representation is normalized so that $\|f(x;\theta_i)\|_2 = 1$. Therefore, we will use the equivalent form $R_{div}(\theta_1, \ldots, \theta_m) = (1/n) \sum_{i<j} \sum_x f(x;\theta_i)^\top f(x;\theta_j)$ in the experiments. We will discuss more details about the diversity term in Section 3.1.

The closeness term R_{close} encourages $\theta_1, \ldots, \theta_m$ to be close in the parameter space. Note that this is different from the diversity term that encourages the diversity in the representation space.

Specifically, we use

$$R_{close}(\theta_1, \dots, \theta_m) = \sum_{i<j} \left\|\theta_i - \bar{\theta}\right\|_2^2, \tag{3}$$

where $\bar{\theta} = (1/m)\sum_i \theta_i$ is the average of θ_i's. As we will discuss in Section 3.2, this term is motivated by the recent success of SAM [5] that tries to find a flat solution in the sense that the loss in the whole local region are small. Similarly, we use the closeness term to ensure these different head are close in the parameter space while keeping loss low, and therefore find a flat region that has better generalization performance.

3. Intuitions and Connections with Existing Methods

In this section, we discuss the connections between two regularization terms with existing methods.

3.1. Diversity Term: Connection to Stein Variational Gradient Descent

SVGD SVGD is a well-known particle-based sampling algorithm to obtain samples from a target distribution [19]. Specifically, SVGD uses a set of m particles $\{\theta_i\}_{i=1}^m$ and moves them according to the update rule below to approximate the target distribution p_*: for each particle θ_i, we have

$$\theta_i^{t+1} \leftarrow \theta_i^t + \eta_t \phi(\theta_i^t), \quad \phi(\theta) = \frac{1}{m}\sum_{j=1}^m [k(\theta_j^t, \theta)\nabla \log p_*(\theta_j^t) + \nabla_{\theta_j^t} k(\theta_j^t, \theta)], \tag{4}$$

where η_t is the stepsize and $k(x, y)$ is a kernel. The second term $\frac{1}{m}\sum_{j=1}^m \nabla_{\theta_j^t} k(\theta_j^t, \theta)$ above can be viewed as a diversity term that encourages different particles to stay away from each other [19]. Liu [18] showed that the empirical measures of the SVGD samples weakly converge to the target distribution when the number of particles goes to infinity.

Given that SVGD is a sampling algorithm that samples from target distribution p_*, one common way to use SVGD as a deep learning algorithm is to adapt the Bayesian perspective and try to sample the optimal parameter θ from $p(\theta|\mathcal{D})$, i.e., posterior of parameter θ given training data $\mathcal{D}$. With some derivations (see Appendix B for details), we have the SVGD update in (4) becomes

$$\phi(\theta) = -\frac{N}{m}\sum_{j=1}^m [k(\theta_j^t, \theta)\nabla \tilde{L}(\theta_j^t)] + \frac{1}{m}\sum_{j=1}^m [\nabla_{\theta_j^t} k(\theta_j^t, \theta)], \tag{5}$$

where $\tilde{L}(\theta) := L(\theta) + \lambda \|\theta\|_2^2$ is the train loss with ℓ_2 regularization with some λ. In particular, the first term above can be viewed as the weighted sum of the gradient of m particles, and the second term is encouraging the diversity of these particles.

Connection to diversity term We now show that the diversity term shares similar idea as SVGD at high level. In fact, the kernel $k(x, y)$ plays an important role in SVGD to encourage the diversity among different particles. We choose kernel $k(\theta, \theta') = \exp(-\|f_\theta - f_{\theta'}\|_2^2/h)$ that measures the similarity in the representation space. Suppose we have approximate $\left\|f_{\theta_i} - f_{\theta_j}\right\|_2^2 \approx \mathbb{E}_x[\|f(x;\theta_i) - f(x;\theta_j)\|_2^2]$ with sufficient many samples. The update (5) now becomes

$$\phi(\theta_i^t) \approx \frac{N}{m}\sum_{j=1}^m e^{-\frac{1}{h}\left\|f_{\theta_i} - f_{\theta_j}\right\|_2^2} \cdot \nabla \tilde{L}(\theta_j^t) + \frac{1}{m}\sum_{j=1}^m e^{-\frac{1}{h}\left\|f_{\theta_i} - f_{\theta_j}\right\|_2^2} \cdot \underbrace{\frac{2}{h}\mathbb{E}_x[\nabla f(x;\theta_j^t)(f(x;\theta_i^t) - f(x;\theta_j^t))}_{g(\theta_i^t, \theta_j^t)}$$

We claim that the second term would encourage the similar representations to move away from each other, and it is approximately the same as the gradient on diversity term R_{div} in (2). To see this, recall that we have the closeness term that encourages θ_i's to be close to each other. Therefore, the Jacobian matrix $\nabla f(x_k;\theta_j)$ is approximately the same as the Jacobian matrix $\nabla f(x_k;\theta_i)$. This leads to $g(\theta_i,\theta_j)$ roughly points to the same direction as $\nabla_{\theta_i}\mathbb{E}_x[\|f(x;\theta_i)-f(x;\theta_j)\|_2^2]$, which is the direction of the gradient on diversity term. Since such direction increases the distance between f_{θ_i} and f_{θ_j}, we can see that moving according to such direction encourages the representations of each particle to become diverse.

3.2. Closeness Term: Connection to Sharpness-Aware Minimization

SAM Sharpness-Aware Minimization (SAM) has recently shown to achieve good performance compared with SGD on various tasks [5]. The main idea of SAM is to not only minimize the loss at current parameter $L(\theta)$, but also minimize the loss around the local region of current parameter by minimizing the loss under worst-case perturbation, that is

$$\min_{\theta}\max_{\|\varepsilon\|_2\le\rho} L(\theta+\varepsilon),$$

where ρ is the hyperparameter that controls the size of neighborhood. In general, solving the inner maximization problem exactly is hard so that one could use the first order approximation

$$\varepsilon(\theta)=\arg\max_{\|\varepsilon\|\le\rho}\varepsilon^\top\nabla L(\theta)=\rho\nabla L(\theta)/\left\|\nabla L(\theta)\right\|_2.$$

Intuitively, SAM seeks flat minima (in the sense of largest eigenvalue of Hessian) by minimizing such worst-case perturbed loss, and flat minima is believed and observed to have close connection with the good generalization performance [9, 11, 12].

Connection to closeness term The closeness term in (3) is motivated by SAM that minimizes the loss in the local region and tries to generalizes SAM to go beyond the worst-case perturbation. In fact, we can view SAM as using 2-head network that outputs $f(x;\theta)$ and $f(x;\theta+\varepsilon(\theta))$ in our framework. Our method tries to generalize this 2-head network to arbitrary m-head network, by allowing m outputs $f(x;\theta_1),\dots,f(x;\theta_m)$ while still maintaining $\theta_1,\dots,\theta_m$ stay close. In this way, the closeness term could still keep the advantages of SAM and tries to find the flat minima that could generalize well.

4. Experiments

Setup To show the efficacy of the proposed optimization approach, we compare it against to the optimizer used by the original CLIP as well as the SAM optimizer. We use two mid-scale datasets, Conceptual Captions 3M (CC3M) and a 15M subset of the YFCC100M (YFCC15M), to pretrain all the models. Following prior work [7, 27], we use ResNet-50 as the image encoder and the transformer as the text encoder.

In order to examine the isolated impact of the optimizer, we maintain consistency among all models by keeping the hyperparameters unrelated to optimization unchanged. We tune the learning rate, weight decay, and other optimization-related parameters and report the highest achieved accuracy. All the models are trained on 8 Nvidia V100 GPUs. See Appendix C for details of hyperparameters.

Results We conducted extensive experiments to evaluate the performance of our proposed multi-head CLIP approach compared to the original CLIP with AdamW optimizer and CLIP with SAM optimizer. The evaluation was performed on two benchmark datasets, CC3M and YFCC15M, with zero-shot performance reported using the ImageNet dataset [3].

To evaluate the performance of multi-head CLIP, we utilized the multiple representations learned from each sample. Two approaches were explored: concatenating the features and using the average as the feature for the data. Through experimentation, we found that using the average feature yielded better performance. Table 1 summarizes the top-1 accuracies achieved by each method. Notably, multi-head CLIP outperformed both CLIP with adamW optimizer and CLIP with SAM optimizer on both datasets. On the CC3M dataset, multi-head CLIP achieved an accuracy of 23.5%, surpassing the accuracy of CLIP with AdamW optimizer (19.4%) and CLIP with SAM optimizer (22.6%). This result indicates that the incorporation of multiple representations in our proposed method enables the model to capture more diverse and discriminative features, enhancing its ability to recognize and classify images effectively. Similarly, on the YFCC15M dataset, Multi-head CLIP achieved an accuracy of 33.9%. In comparison, CLIP with AdamW optimizer obtained an accuracy of 31.3%, while CLIP with SAM optimizer achieved 29.1%. These results demonstrate the consistent superiority of multi-head CLIP in leveraging multiple representations to enhance the model's understanding of both image and text inputs.

The improved performance of Multi-head CLIP can be attributed to its ability to learn diverse representations while encouraging convergence to a flat loss region. By promoting diversity among representations, the model gains a broader perspective and captures a richer set of features, leading to enhanced generalization capabilities.

Algorithm	Optimizer	Pre-training dataset	Top-1 accuracy on ImageNet
CLIP	AdamW	CC3M	19.4%
CLIP	SAM	CC3M	22.6%
Multi-head CLIP	AdamW	CC3M	**23.5**%
CLIP	AdamW	YFCC15M	31.3%
CLIP	SAM	YFCC15M	29.1%
Multi-head CLIP	AdamW	YFCC15M	**33.9**%

Table 1: Comparison of Top-1 accuracies on ImageNet

5. Conclusion

In this work, we introduced a new method called multi-head CLIP that extends the original CLIP framework. Our approach allows the model to learn multiple representations for each data point, resulting in improved performance compared to the original CLIP. Our experiments demonstrated that multi-head CLIP outperforms the original CLIP, indicating that better representations are learned. This finding suggests that the idea of learning multiple representations for each data point has broader applications and benefits. By enabling the model to learn diverse representations, our method shows promising results and opens up new possibilities for improving representation learning in various tasks.

References

[1] Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. *arXiv preprint arXiv:2110.08529*, 2021.

[2] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LtKcMgGOeLt.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent YF Tan. Efficient sharpness-aware minimization for improved training of neural networks. *arXiv preprint arXiv:2110.03141*, 2021.

[5] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

[6] Andreas Fürst, Elisabeth Rumetshofer, Johannes Lehner, Viet Thuong Tran, Fei Tang, Hubert Ramsauer, D P Kreil, Michael K Kopp, Günter Klambauer, Angela Bitto-Nemling, and Sepp Hochreiter. CLOOB: Modern hopfield networks with infoLOOB outperform CLIP. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=q-LMlivZrV.

[7] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. *Advances in Neural Information Processing Systems*, 35:6704–6719, 2022.

[8] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR, 2017.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

[10] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[11] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

[12] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=H1oyRlYgg.

[13] Minyoung Kim, Da Li, Shell X Hu, and Timothy Hospedales. Fisher sam: Information geometry and sharpness aware minimisation. In *International Conference on Machine Learning*, pages 11148–11161. PMLR, 2022.

[14] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *International Conference on Machine Learning*, pages 5905–5914. PMLR, 2021.

[15] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[16] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[17] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=zq1iJkNk3uN.

[18] Qiang Liu. Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30, 2017.

[19] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

[20] Yang Liu, Prajit Ramachandran, Qiang Liu, and Jian Peng. Stein variational policy gradient. *arXiv preprint arXiv:1704.02399*, 2017.

[21] Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12360–12370, 2022.

[22] Peng Mi, Li Shen, Tianhe Ren, Yiyi Zhou, Xiaoshuai Sun, Rongrong Ji, and Dacheng Tao. Make sharpness-aware minimization stronger: A sparsified perturbation approach. *arXiv preprint arXiv:2210.05177*, 2022.

[23] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 529–544. Springer, 2022.

[24] Maximilian Mueller, Tiffany Vlaar, David Rolnick, and Matthias Hein. Normalization layers are all that sharpness-aware minimization needs. *arXiv preprint arXiv:2306.04226*, 2023.

[25] Renkun Ni, Ping-yeh Chiang, Jonas Geiping, Micah Goldblum, Andrew Gordon Wilson, and Tom Goldstein. K-sam: Sharpness-aware minimization at the speed of sgd. *arXiv preprint arXiv:2210.12864*, 2022.

[26] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[28] Dilin Wang and Qiang Liu. Learning to draw samples: With application to amortized mle for generative adversarial learning. *arXiv preprint arXiv:1611.01722*, 2016.

[29] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cpDhcsEDC2.

[30] Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *Advances in neural information processing systems*, 31, 2018.

[31] Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8156–8165, 2021.

[32] Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, sekhar tatikonda, James s Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=edONMAnhLu-.

Appendix A. Related Works

Vision-language pretrained models Recently CLIP [27] has attracted lots of attention due to its ability to learn powerful representations of images and texts for downstream tasks from millions of data. It uses the contrastive learning [26] to align the cross-modal representation of image and text to learn these features. ALIGN [10] was also developed concurrently using the similar idea. Later, many works have been proposed to improve the performance of CLIP under the similar framework [6, 16, 17, 23, 29]. Unlike previous works that only provides one representation per data, our method is able to provide multiple representations for each data that capture more relevant information.

Stein Variational Gradient Descent Stein Variational Gradient Descent (SVGD) is a particle-based variational inference algorithm that aims to sample from an unknown target distribution [19]. Specifically, it uses the Stein's method to design an iterative method that transports a set of particles by performing a type of functional gradient descent on the KL divergence. SVGD has been used in many applications in deep learning, such as generative models [28], reinforcement learning [8, 20] and meta learning [30].

Sharpness-aware minimization Sharpness-aware minimization [5] is a recently proposed optimization method that improves the generalization of models by minimizing the loss in the local neighborhood in the parameter space. Similar method was also proposed in Zheng et al. [31] concurrently. Since then, several variants of SAM have been proposed to improve either the performance [13, 14, 32] or the computational efficiency [4, 21, 22, 24, 25]. Bahri et al. [1], Chen et al. [2] showed that SAM is able to improve the performance in various tasks in computer vision and natural language processing. In this work, we also leverage the idea of SAM to design the closeness term in our method.

Appendix B. Derivation for (5)

Recall SVGD uses a set of m particles $\{\theta_i\}_{i=1}^m$ and moves them according to the update rule below to approximate the target distribution p_*: for each particle θ_i, we have

$$\theta_i^{t+1} \leftarrow \theta_i^t + \eta_t \phi(\theta_i^t), \quad \phi(\theta) = \frac{1}{m}\sum_{j=1}^{m}[k(\theta_j^t, \theta)\nabla \log p_*(\theta_j^t) + \nabla_{\theta_j^t} k(\theta_j^t, \theta)],$$

where η_t is the stepsize and $k(x, y)$ is a kernel.

Often in the classification setting, we model the loss function $L(\theta)$ as negative log-likelihood of training data $L(\theta) = (1/N)\sum_i -\log p(x_i|\theta)$. Thus, by the Bayes' theorem we have the posterior becomes

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p_0(\theta) = e^{-NL(\theta)}p_0(\theta),$$

where $p_0(\theta)$ is the prior. Choosing the prior $p_0(\theta)$ as Gaussian distribution $N(0, \sigma^2 I)$, we know the SVGD update above becomes

$$\phi(\theta) = -\frac{N}{m}\sum_{j=1}^{m}[k(\theta_j^t, \theta)\nabla \tilde{L}(\theta_j^t)] + \frac{1}{m}\sum_{j=1}^{m}[\nabla_{\theta_j^t} k(\theta_j^t, \theta)],$$

where $\tilde{L}(\theta) := L(\theta) + \|\theta\|_2^2 / 2\sigma^2 N$ is the train loss with ℓ_2 regularization. In particular, the first term above can be viewed as the weighted sum of the gradient of m particles, and the second term is encouraging the diversity of these particles.

Appendix C. Experiment Details

For the CC3M dataset, all three models are trained from scratch for 30 epochs. We use a batch size of 128 and the cosine learning rate schedule with 10000 warmup steps. In the case of the model trained with adamW, we set the initialize learning rate to 1e-3, while setting the weight decay to 0.1. As for the model trained with SAM, we use a weight decay of 1e-4 and set $\rho = 0.1$. In the case of multi-head CLIP, we adopt a learning rate of 2e-3 and a weight decay of 0.2, use 5 heads. Moreover, we set the regularization coefficients $R_{close} = 0.05$ and $R_{div} = 0.05$.

Similarly, for the YFCC15M dataset, we train all three models for 32 epochs. We use a batch size of 128 and the cosine learning rate schedule with 10000 warmup steps. In the case of the model trained with adamW, we set the initialize learning rate to 5e-4, while setting the weight decay to 0.2. As for the model trained with SAM, we use a learning rate of 0.1 and a weight decay of 1e-4 and

set $\rho = 0.1$. In the case of multi-head CLIP, we adopt a learning rate of $8\text{e-}4$ and a weight decay of 0.33, use 5 heads. Moreover, we set the regularization coefficients $R_{close} = 0.02$ and $R_{div} = 0.006$. We also use a smaller logit scale of $\log(1/0.3)$ instead of $\log(1/0.07)$ in the original CLIP.