Improving CLIP Training

Team Members: Omar Khater, Michael Norman
Course: CSCE 636 Deep Learning, Fall 2024

Contents

Abstract

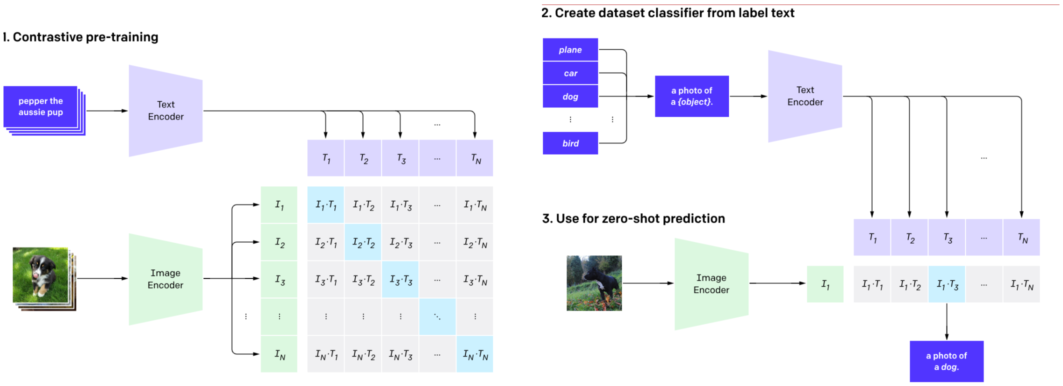
Global contrastive learning has gained significant attention for its ability to leverage dataset-wide statistics, offering robust representation learning in multimodal settings such as image-text pairs. This project focuses on optimizing global contrastive loss functions, including stochastic optimization for global contrastive learning (SogCLR) and indvidual SogCLR (iSogCLR) variants, to enhance performance in multimodal self-supervised learning tasks.

Through systematic evaluation and optimization using advanced optimizers such as AdamW and RAdam, we achieve significant improvements over both the benchmark CLIP model and the provided codebase (*SogCLR*). Specifically, our proposed workflow improves MSCOCO TR@1 by 25%, MSCOCO IR@1 by 23.6%, and ImageNet ACC@1 by 33.7% compared to the CLIP model, resulting in a 28.9% improvement in the overall average score. Furthermore, compared to the provided *SogCLR* implementation, our method achieves gains of 4.3% in MSCOCO TR@1, 7.4% in MSCOCO IR@1, and 16.3% in ImageNet ACC@1, leading to a 10.8% improvement in the overall average.

This work is publicly available at this GitHub repository.

1 Introduction

Contrastive learning has revolutionized self-supervised learning by leveraging similarity and dissimilarity comparisons within data to generate high-quality feature representations. In multimodal domains, particularly image-text pairings, contrastive learning has enabled breakthroughs in representation quality, supporting tasks such as zero-shot classification and cross-modal retrieval. CLIP (Contrastive Language-Image Pretraining) [16] exemplifies this paradigm, utilizing local contrastive loss to align visual and textual embeddings within a batch. The architecture for CLIP model is shown in figure 1.


I. Contrastive pre-training
pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ⋮ I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
2. Create dataset classifier from label text
plane
car
dog
bird
a photo of a {object}.
Text Encoder
3. Use for zero-shot prediction
Image Encoder
I_1
T_1 T_2 T_3 ... T_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
a photo of a dog.

(a) Step 1: Contrastive pre-training.

(b) Step 2, 3: Inference.

Figure 1: Overview of the CLIP architecture and workflow, including pretraining and inference stages.

This project advances multimodal self-supervised learning by focusing on global contrastive loss functions, which extend contrastive learning beyond batch-level comparisons to include dataset-wide statistics. These methods offer the potential for more robust feature learning and improved generalization.

In addition to exploring advanced loss functions such as *SogCLR* and *iSogCLR*, we also incorporate state-of-the-art Bayesian optimization techniques to tune hyperparameters.

The objective of this project is to enhance the training of CLIP-like models through the development and evaluation of better algorithms for global contrastive loss optimization. We aim to:

- Explore global loss functions, including state-of-the-art techniques such as *SogCLR* and *iSogCLR*, alongside their optimizations.

- Investigate the effectiveness of Bayesian optimization and advanced optimizers, such as AdamW, RAdam in improving the convergence and performance of these global losses.
- Evaluate performance on the prescribed datasets using standard metrics, with rigorous adherence to the given constraints.

2 Background

2.1 Self-Supervised Learning and Contrastive Learning

Self-Supervised Learning (SSL) has become a foundational paradigm for pre-training deep neural networks, demonstrating effectiveness across a wide range of domains, including Natural Language Processing (NLP) [14, 4, 10] and Computer Vision (CV) [5, 20, 12]. SSL aims to learn generalizable data representations from unlabelled data, which can then be transferred to various downstream tasks.

Contrastive Learning (CL) has emerged as a simple yet powerful framework within SSL. It aligns representations of "positive" pairs (e.g., augmented views of the same image) while pushing apart "negative" pairs (e.g., augmented views of different images). CL has achieved state-of-the-art results in image and text representation learning [3, 7, 18, 8]. Extending CL to multimodal data, where images and text are treated as different views of the same concept, has enabled models like CLIP [16] to achieve remarkable performance on various visual understanding tasks.

Despite its success, CLIP's reliance on batch-level contrastive loss introduces challenges, such as sensitivity to mini-batch size and slow convergence, especially in large-scale multimodal datasets. These limitations motivate the development of global contrastive loss functions.

2.2 Local and Global Contrastive Loss

The standard CLIP training methodology uses a mini-batch-based contrastive loss. Let $\mathcal{B}$ be a mini-batch of m image-text pairs (x_i, z_i). For each pair, the contrastive loss contrasts the similarity scores between matching pairs (x_i, z_i) and non-matching pairs (x_i, z_j) and (z_i, x_j) within the batch. The local contrastive loss for the mini-batch is computed as:

$$L(w, \tau, \mathcal{B}) = \frac{1}{m} \sum_{(x_i, z_i) \in \mathcal{B}} \log \left(\frac{\exp(h_i(w)^\top e_i(w)/\tau)}{\sum_{z_j \in \mathcal{B}} \exp(h_i(w)^\top e_j(w)/\tau)} \right) + \text{(symmetric term)},$$

where τ is the temperature parameter, and $h_i(w)$ and $e_i(w)$ are the normalized image and text embeddings, respectively.

While effective for small-scale datasets, the local loss's reliance on mini-batch negatives can lead to suboptimal performance on larger datasets. Global Contrastive Loss (GCL) addresses this issue by contrasting embeddings against the entire dataset:

$$L(w, \tau, \mathcal{D}) = \frac{1}{n} \sum_{x_i \in \mathcal{D}} \log \left(\frac{\exp(h_i(w)^\top e_i(w)/\tau)}{\sum_{z_j \neq z_i} \exp(h_i(w)^\top e_j(w)/\tau)} \right) + \text{(symmetric term)},$$

where $\mathcal{D}$ is the full dataset. By leveraging dataset-wide negatives, GCL improves convergence and representation quality.

2.3 SogCLR and iSogCLR

Training models with GCL is computationally expensive for large datasets. To mitigate this, SogCLR [19] introduces a memory-efficient stochastic optimization approach. The key idea is to maintain moving averages of negative pair contributions:

$$u_{1,i,t} = (1 - \gamma) u_{1,i,t-1} + \gamma g_1(w_t, x_i, \mathcal{B}_{1t}^-),$$
$$u_{2,i,t} = (1 - \gamma) u_{2,i,t-1} + \gamma g_2(w_t, z_i, \mathcal{B}_{2t}^-),$$

where γ is a smoothing parameter, and g_1 and g_2 are contrastive terms computed over mini-batches. The model parameters are updated using:

$$G_t = \frac{1}{m} \sum_{x_i \in \mathcal{B}_t} \frac{\nabla g_1(w_t, x_i, \mathcal{B}_t)}{\varepsilon + u_{1,i,t}} + \frac{1}{m} \sum_{z_i \in \mathcal{B}_t} \frac{\nabla g_2(w_t, z_i, \mathcal{B}_t)}{\varepsilon + u_{2,i,t}}.$$

This approach enables efficient optimization of the global objective without requiring full dataset computations.

iSogCLR [15] extends SogCLR by dynamically optimizing the temperature parameter τ, treating it as a trainable variable. This is achieved through a robust optimization framework:

$$\min_{w,\tau_1,\tau_2} \frac{1}{n} \sum_{x_i \in \mathcal{D}} L_1(w, \tau_{i1}, x_i, \mathcal{D}) + \frac{1}{n} \sum_{z_i \in \mathcal{D}} L_2(w, \tau_{i2}, z_i, \mathcal{D}) + \rho \sum_{i=1}^{n} (\tau_{i1} + \tau_{i2}),$$

where ρ is a regularization parameter. This formulation improves the robustness of learned representations, particularly for imbalanced datasets.

2.4 Datasets and Metrics

This project adheres to fixed datasets and evaluation criteria to ensure consistent and fair evaluation:

Datasets:

- **Training:** A 100k subset of the Conceptual Captions 3M (CC3M) dataset, which contains diverse and noisy image-text pairs derived from web data. It is designed to train models for learning robust image-text alignments.
- **Validation:**
  - **MSCOCO:** A large-scale dataset of high-quality image-text pairs focused on common objects in context, commonly used for image-to-text and text-to-image retrieval tasks.
  - **ImageNet:** A benchmark dataset comprising labeled images across 1,000 categories, used here for evaluating zero-shot image classification.

Metrics: The performance of models is evaluated based on three metrics:

- **Image-to-Text Recall at Rank 1 (IR@1):** Measures the percentage of images whose corresponding text is ranked first in a retrieval task. Higher IR@1 indicates better alignment between image and text embeddings.
- **Text-to-Image Recall at Rank 1 (TR@1):** Measures the percentage of text queries whose corresponding images are ranked first in a retrieval task. Higher TR@1 reflects improved cross-modal retrieval accuracy.
- **Top-1 Accuracy (ACC@1):** For zero-shot classification, ACC@1 computes the proportion of test samples where the predicted class (from image embeddings) matches the ground-truth label. It indicates the model's ability to generalize to unseen classes.

The overall evaluation metric is the average of IR@1, TR@1, and ACC@1, providing a unified measure of retrieval and classification performance.

2.5 Bayesian Optimization for Hyperparameter Tuning

Efficient hyperparameter tuning is crucial for improving model performance, particularly in complex optimization problems like global contrastive loss. Bayesian optimization is a probabilistic model-based approach that optimizes a target function by iteratively constructing a surrogate model. This model predicts the objective's behavior based on prior evaluations, guiding subsequent hyperparameter selections to balance exploration and exploitation [17].

Bayesian optimization employs acquisition functions, such as Expected Improvement (EI) or Upper Confidence Bound (UCB), to determine the next set of hyperparameters to evaluate. Unlike grid or random search, which sample hyperparameters indiscriminately, Bayesian optimization strategically narrows the search space, significantly reducing computation time while improving results. This efficiency has led to its adoption in various machine learning domains espicially large-scale neural network training [17, 2, 9].

In this project, Bayesian optimization is applied to tune critical hyperparameters, such as the learning rate, temperature parameter τ, and optimizer configurations, to enhance the training of CLIP-like models with global contrastive loss.

3 Related Work

Self-supervised learning (SSL) has revolutionized representation learning by leveraging unlabeled data to learn robust features. Contrastive learning, a prominent approach within SSL, trains models by contrasting positive pairs against negative pairs. Early works like SimCLR [3] and MoCo [7] established the foundations of contrastive learning, primarily focusing on single-modality tasks.

Multimodal contrastive learning extends these concepts to multiple data modalities, such as image and text, as demonstrated by CLIP [16]. CLIP optimizes a local contrastive loss over mini-batches, achieving remarkable zero-shot capabilities. However, its reliance on mini-batch statistics limits scalability and convergence for large datasets.

To address these limitations, global contrastive loss functions have been introduced:

- **SogCLR** [19]: Maintains dataset-level statistics to overcome the dependence on mini-batch size.
- **iSogCLR**: Extends SogCLR by incorporating temperature learning and advanced regularization techniques, including DRO mechanisms.
- **Global Objectives**: Other global loss methods explore dataset-wide invariance and variance constraints, such as VICReg [1].

Parallelly, advancements in optimizers like AdamW [13], NovoGrad [6], and RAdam [11] have improved optimization dynamics for large-scale learning tasks. This project builds upon these developments by systematically evaluating global contrastive losses and their interaction with advanced optimizers.

4 Modeling

The provided modeling framework is designed to flexibly support training and evaluation of multimodal contrastive learning models, with a specific focus on optimizing loss functions and hyperparameters. At its core, the pipeline implements a CLIP-like architecture, combining a pretrained image encoder and text encoder to learn aligned representations for images and text.

4.1 Modular Design for Loss Functions and Optimizers

The codebase is refactored to enable seamless experimentation with various global contrastive loss functions, including *SogCLR*, *iSogCLR*, and their variants. Each loss function is integrated as a modular component, allowing dynamic configuration based on the desired training objective. Similarly, the optimizer setup supports a wide range of algorithms such as AdamW, Novograd, and their fused counterparts, ensuring compatibility with diverse optimization strategies.

4.2 Model Architecture

The architecture includes:

- **Image Encoder:** A ResNet-50 backbone from the `timm` library, pretrained on ImageNet, processes image inputs.
- **Text Encoder:** A DistilBERT model from the `huggingface` library, pretrained on BookCorpus and Wikipedia, encodes text inputs.
- **Projection Layers:** Linear layers project image and text embeddings into a shared latent space for contrastive learning.

4.3 Training and Evaluation Pipeline

The pipeline supports both training and evaluation workflows:

- **Training:** The framework implements distributed training with support for gradient scaling (for mixed precision) and dynamic learning rate scheduling. Loss gradients are backpropagated and parameters updated based on the selected loss function and optimizer.

- **Evaluation:** Validation includes retrieval tasks (Image-to-Text and Text-to-Image) and zero-shot classification, using pre-defined datasets such as MSCOCO and ImageNet. Based on an objective score evaluated on the validation sets, the best model is selected.

  The objective value is described in 1:

$$\text{Objective Value} = \frac{1}{3}(\text{IR@1} + \text{TR@1} + \text{ACC@1}), \tag{1}$$

where:

- IR@1: Image-to-Text Recall at Rank 1, measuring the proportion of images whose corresponding text is ranked first.
- TR@1: Text-to-Image Recall at Rank 1, quantifying the percentage of text queries whose matching images are ranked first.
- ACC@1: Top-1 Accuracy, indicating the proportion of correct classifications for zero-shot classification tasks.

4.4 Customization for Experimentation

The framework is designed with high flexibility:

- **Loss Function Optimization:** Parameters specific to each loss function, such as temperature τ or regularization terms, are tunable.
- **Optimizer Configurations:** A wide range of optimizer settings can be tested, enabling thorough exploration of optimization dynamics.
- **Checkpointing and Resuming:** Training states can be saved and resumed, ensuring efficient experimentation.

This modular and extensible design facilitates systematic exploration of loss functions, optimizers, and hyperparameters, aligning with the project's goal of optimizing global contrastive learning frameworks.

5 Experiments & Results

This section describes the experimental setup, evaluation protocol, and results for optimizing global contrastive loss functions across three phases. Each phase incrementally narrows the hyperparameter search space, focusing on identifying and tuning the best-performing configurations.

5.1 Experimental Setup

The experiments are conducted in three phases:

1. **Phase 1: Initial Search.** A Bayesian search explores combinations of loss functions and optimizers over a wide search space. Experiments run for 10 epochs using 20% of the training data.
2. **Phase 2: Grid Search.** Top combinations identified in Phase 1 are trained on the full dataset for 15 epochs. A grid search refines the hyperparameter space for selected loss functions and optimizers.
3. **Phase 3: Fine-Tuning.** The best-performing configuration from Phase 2 undergoes a comprehensive hyperparameter search and is trained for 30 epochs with the full training dataset.

5.2 Results

5.2.1 Phase 1: Initial Search

In this phase, we explored 10 optimizers and 3 loss functions, as shown in the search space below:

- **Loss Functions:** SogCLR, iSogCLR_New, iSogCLR_New_v2.
- **Optimizers:** Adam, AdamW, RAdam, NovoGrad, FusedAdam, FusedLAMB, FusedNovoGrad, RMSProp, Momentum, NVNovoGrad.

The primary goal was to identify sensitivity of the objective metric to different combinations of loss functions and optimizers. The results are visualized in Figure 2 and Figure 3.

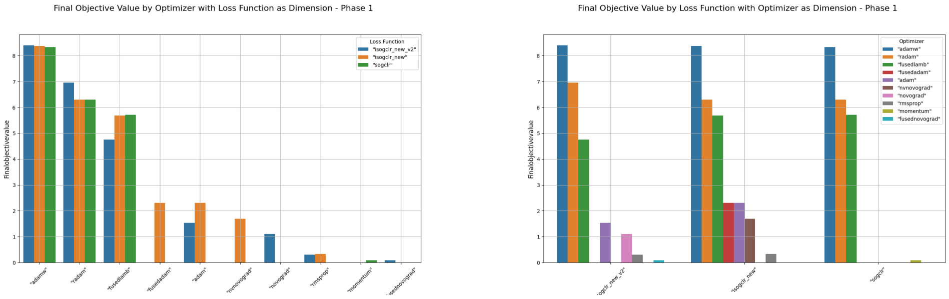


Figure 2: (Left) Validation objective values by optimizer, with loss function as dimension. (Right) Validation objective values by loss function, with optimizer as dimension (Phase 1).

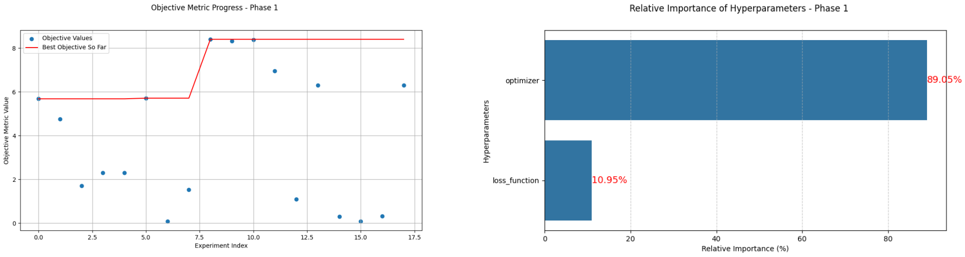


Figure 3: (Left) Progression of objective metric. (Right) Relative importance of hyperparameters (Phase 1).

5.2.2 Phase 2: Grid Search

Based on insights from Phase 1, we selected the top 3 optimizers (AdamW, RAdam, FusedLAMB) and 3 loss functions (SogCLR, iSogCLR_New, iSogCLR_New_v2) for grid search on the full dataset. The search space was refined to focus on specific hyperparameters relevant to these configurations.

The results for this phase are visualized in Figure 4 and Figure 5, while Table 1 summarizes the important hyperparameters used for both phase 1 and 2.

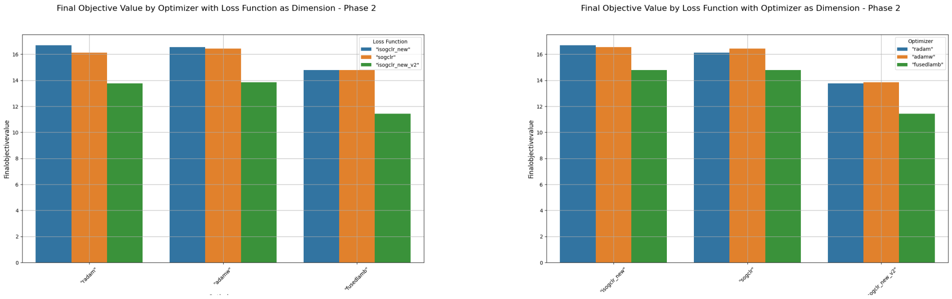


Figure 4: (Left) Validation objective values by optimizer, with loss function as dimension. (Right) Validation objective values by loss function, with optimizer as dimension (Phase 2).

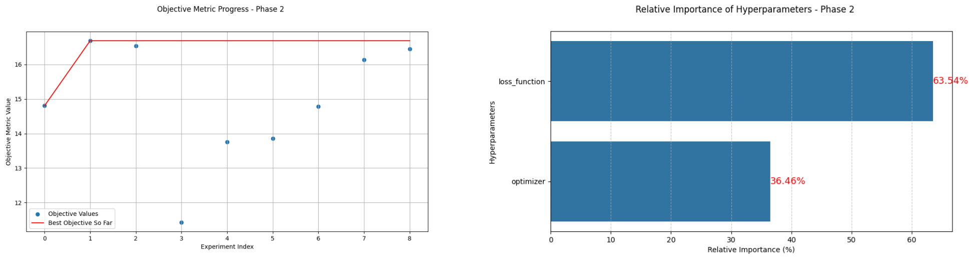


Figure 5: (Left) Progression of objective metric. (Right) Relative importance of hyperparameters (Phase 2).

Parameter	**Value**	**Description**
`embed_dim`	256	Dimensionality of the embedding space for image and text representations.
`lr`	0.0002	Learning rate for the optimizer.
`warmup_epochs`	5	Number of warmup epochs where the learning rate gradually increases.
`cooldown_epochs`	0	Number of cooldown epochs after training where learning rate gradually decreases.
`batch_size_train`	128	Batch size used during training.
`sogclr_gamma`	0.8	Momentum coefficient for the SogCLR loss function.
`rho_I`	8.0	Regularization coefficient for image embeddings in iSogCLR_New.
`rho_T`	8.0	Regularization coefficient for text embeddings in iSogCLR_New.
`eta_init`	0.001	Initial step size for specific loss function updates.
`tau_init`	0.01	Initial temperature for contrastive loss.
`personalized_tau`	0	Boolean indicating whether to use personalized temperatures for each sample.
`learnable_temp`	0	Boolean indicating whether the temperature parameter is learnable.

Table 1: Important hyperparameters and their descriptions for Phases 1 and 2.

5.2.3 Phase 3: Fine-Tuning

In the final phase, we trained the best-performing configuration (`iSogCLR_New` with `RAdam`) on the full dataset for 30 epochs. The hyperparameters tuned in this phase included learning rate, weight decay, and loss-specific parameters such as temperature and ρ_I. The progression of the objective metric is shown in Figure 6. The winning parameters are summarized in **??**.

Loss Function	**Optimizer**	**IR@1**	**TR@1**	**ACC@1**	**Average**
iSogCLR_New	RAdam	11.52	15.00	28.54	18.33

Table 2: Final validation results after fine-tuning (Phase 3).

5.2.4 Best Training Job Analysis

In figure 7, we show the learning progress for the best-performing job (tabulated in table 3). It can be noticed the rapid progress achieved in the first 20 epochs (best epoch) before the performance almost saturates. This suggests using early stopping in the future to reduce the training cost.

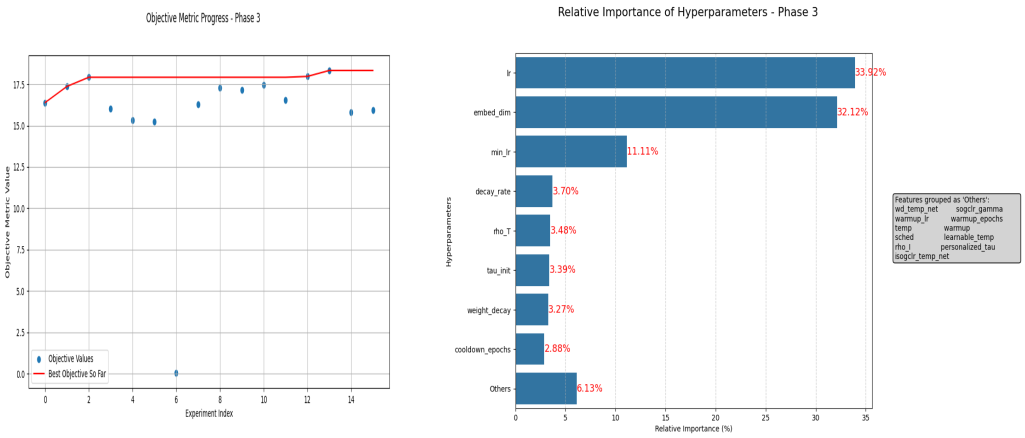


Figure 6: (Left) Progression of objective metric. (Right) Relative importance of hyperparameters (Phase 3).

Hyperparameter	**Value**	**Description**
`lr`	0.000164	Learning rate for the RAdam optimizer.
`weight_decay`	0.000373	Weight decay for the RAdam optimizer.
`embed_dim`	512	Dimensionality of the shared image-text embedding space.
`sogclr_gamma`	0.966	Momentum coefficient for updating moving averages in the SogCLR loss function.
`rho_I`	9.96	Regularization coefficient for image embeddings in *iSogCLR_New*.
`rho_T`	7.47	Regularization coefficient for text embeddings in *iSogCLR_New*.
`tau_init`	0.01	Initial temperature for the contrastive loss.
`temp`	0.0585	Initial global temperature parameter for contrastive loss.
`wd_temp_net`	5.85e-6	Weight decay for the temperature network parameters.
`decay_rate`	0.5679	Decay rate for the learning rate scheduler.
`min_lr`	1e-5	Minimum learning rate during learning rate scheduling.
`cooldown_epochs`	1	Number of cooldown epochs for gradual learning rate reduction.
`warmup_epochs`	5	Number of warmup epochs where the learning rate gradually increases.
`warmup_lr`	0.001	Initial learning rate during warmup phase.
`sched`	`tanh`	Learning rate scheduler type.

Table 3: Tuned hyperparameters and their values from Phase 3 with descriptions.

5.3 Discussion

From Phase 1 experiments, the Bayesian search efficiently explored the search space giving us important insight about the critical role of the optimizer (89%) over the loss function (11%). Phase 2 helped giving a direction for which configuration to fully tune. Finally, Phase 3 results 6 demonstrates the relative importance played by each dimension to make the Bayesian search select the next experiment. It worth emphasizing the critical role for the learning rate and embedding dimension for training this CLIP based model.

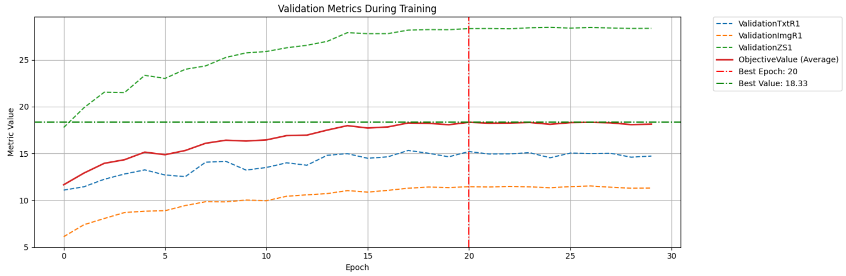


Figure 7: Validation metrics progression for the best-performing configuration during Phase 3 fine-tuning.

6 Conclusion

Our experiments demonstrate that both *SogCLR* and *iSogCLR_New* loss functions outperform the benchmark CLIP model across key metrics on MSCOCO and ImageNet datasets. Notably, *iSogCLR_New* achieves the highest average score of 18.33 (28.9% , 16.3% improvement against compared settings) , surpassing both the CLIP and SogCLR models. These improvements highlight the effectiveness of optimizing global contrastive loss functions with advanced hyperparameter tuning and structured training workflows.

The table below compares the results of our work (*iSogCLR_New*) against the benchmark CLIP and SogCLR models. The best results for each metric are bolded for clarity.

Method	MSCOCO TR@1	MSCOCO IR@1	ImageNet ACC@1	Average
CLIP (Benchmark)	12.00	9.32	21.35	14.22
SogCLR (Provided Codebase)	14.38	10.73	24.54	16.55
iSogCLR_New (Ours)	**15.00**	**11.52**	**26.63**	**18.33**

Table 4: Performance comparison of previous work and our work across MSCOCO and ImageNet datasets. Best results are bolded.

These results highlight the substantial advancements achieved by our work, particularly with the *iSogCLR_New* loss function, showcasing its potential for multimodal self-supervised learning tasks.

Team Contributions

This section outlines the contributions of each team member to the project.

Omar Khater

- Initiated the project by interacting with the provided codebase.
- Planned the work strategy and outlined a systematic experimental approach.
- Project Report
- Training for all phases
- Results analysis

Michael Norman

To be completed.

Declaration

We acknowledge that generative AI technologies, including OpenAI's ChatGPT, were utilized to assist in drafting this report. The AI tools provided support in organizing ideas, structuring sections, and refining the language of the document.

The authors, Omar Khater and Michael Norman, have thoroughly reviewed the report and confirm its content. We take full responsibility for the ideas, analyses, and conclusions presented in this report.

References

[1] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[2] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546–2554, 2011.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607, 2020.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Boris Ginsburg. Stochastic gradient methods with layer-wise adaptive moments for training of deep networks. *arXiv preprint arXiv:1905.11286*, 2019.

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the 33rd IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[8] Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *arXiv preprint arXiv:2207.13532*, 2022.

[9] Omar Khater, Ali Khater, Ashar Seif Al-Nasr, Samir Abozyd, Bassem Mortada, and Yasser M. Sabry. Advancing near-infrared spectroscopy: A synergistic approach through bayesian optimization and model stacking. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 318:124492, 2024.

[10] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[11] Liangchen Liu and Wu Jianlong. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[14] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[15] Zi-Hao Qiu, Quanqi Hu, Zhuoning Yuan, Denny Zhou, Lijun Zhang, and Tianbao Yang. Not all semantics are created equal: Contrastive self-supervised learning with automatic temperature individualization. *arXiv preprint arXiv:2305.11965*, 2023.

[16] 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 *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021.

[17] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

[18] Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? *arXiv preprint arXiv:2201.05119*, 2022.

[19] Zhuoning Yuan, Yuexin Wu, Zi-Hao Qiu, Xianzhi Du, Lijun Zhang, Denny Zhou, and Tianbao Yang. Provable stochastic optimization for global contrastive learning: Small batch does not harm performance. In *International Conference on Machine Learning*, pages 25760–25782. PMLR, 2022.

[20] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.