Fine-tuning Pre-trained Models for Multimodal Sentiment Analysis: A Comprehensive Analysis of Visual-Textual Emotion Recognition : CS 7643

Fattaneh Ameri Mahabadian, Mark Lights, Yanzhe Yang, Mehrnaz Hooshmand
Georgia Institute of Technology
famerim@gatech.edu, mlights3@gatech.edu, yyang3112@gatech.edu ,mhooshmand@gatech.edu

Abstract

Multimodal sentiment analysis requires models to understand both visual content and natural language to accurately identify tone. While pre-trained models like CLIP and BLIP-2 achieve strong performance on standard vision-language tasks, their ability to interpret and reason about sentiment across modalities remains underexplored. This project investigates how parameter- efficient fine-tuning methods (LoRA, Adapters) can improve multimodal tone classification while maintaining computational efficiency. We fine-tune pre-trained CLIP, BLIP-2, LLaVA and TinyLLaVA models on a sentiment-classification dataset, then conduct comprehensive analysis of how these methods affect the model's ability to detect tone, understand cross-modal sentiment alignment, and interpret multimodal content in social media posts. Our work addresses fundamental challenges in sentiment AI and provides insights for building more robust and interpretable multimodal tone-classification systems.

1. Introduction/Background/Motivation

Multimodal sentiment analysis asks whether we can reliably detect the emotional tone of a post when people use both an image and a short caption. In this project, we focus on a concrete question: can lightweight fine-tuning make commonly used vision–language models better at picking up subtle and sometimes confusing emotional cues in both text and image, while still achieving high accuracy? To study this, we fine-tune four pre-trained multimodal models : CLIP, BLIP-2, LLaVA, and TinyLLaVA -using parameter-efficient methods such as LoRA and adapters [5, 4]. Our goal is to measure how model structure and fine-tuning strategy influence each model's ability to detect tone, align sentiment across text and images, and provide interpretable multimodal decisions [15, 8, 10].

Existing multimodal sentiment systems have mostly relied on earlier CNN+LSTM pipelines that process images and text separately and then merge their features, which can struggle with noisy social-media posts and nuanced emotions [14, 3]. More recent vision–language models such as CLIP and BLIP-2, and instruction-tuned models such as LLaVA, have demonstrated strong results on general multimodal understanding [15, 8, 10]. However, their behavior on fine-grained sentiment tasks , especially when image and text provide weak, subtle, or conflicting emotional signals, remains underexplored. Improving this capability matters for applications such as content moderation, safety monitoring, and brand or public-opinion analysis, where understanding not only what a post contains but also how it feels is important.

We used the MVSA-Single static dataset from the Kaggle [12] website. This dataset was developed by T. Niu, S. A. Zhu, L. Pang, and A. El Saddik [13] at the Multimedia Communications Research Lab, University of Ottawa. It contains 4,869 Twitter (now known as X) posts with a total size of 213 MB. Each post includes an image file in .jpg format, a caption file in .txt format, and sentiment annotations stored in labelResultAll.txt (positive, negative, or neutral). The data was collected from social media platforms prior to 2016 using crawling techniques and curated by humans for filtering and sentiment labeling. The dataset is intended solely for research purposes. MVSA-Single was created to support multimodal sentiment analysis research, specifically addressing the gap in datasets that combine images and text captions with sentiment labels. No official train/validation/test splits are provided with the dataset; in our work, we applied an 60/20/20 split for training, validation, and testing.We also filtered dataset to retain only entries with identical text and image labels.Because the classes are imbalanced (positive-majority), we report both accuracy and macro/weighted F1; the agreement subset (about 2,600 samples) reduces annotation noise from conflicting labels.

2. Approach

2.1. LLaVA

We selected LLaVA (Large Language and Vision Assistant) [9], a state-of-the-art multimodal model that combines a vision encoder (CLIP ViT-L/14) with a large language model (LLaMA-7B) through a learned vision-language connector. Unlike CLIP's contrastive learning approach, LLaVA uses instruction tuning to enable natural language understanding of visual content, making it particularly well-suited for sentiment analysis tasks that require reasoning about emotional content in images and text. We chose the LLaVA-1.5-7B variant for its strong performance on vision-language understanding tasks while remaining computationally feasible for fine-tuning with parameter-efficient methods.

2.2. TinyLLaVA

We selected TinyLLaVA [19], a family of small-scale multimodal models that integrate vision and language, designed to deliver LLaVA-style capabilities with significantly fewer parameters. This smaller variant was chosen to balance performance with computational efficiency, making it well-suited for our limited GPU resources. To assess effectiveness, we evaluated classification performance on the imbalanced MVSA-Single dataset using weighted F1-score, accuracy, confusion matrices, and detailed classification reports, enabling analysis of model behavior across sentiment classes.

2.3. BLIP-2

BLIP2 serves as the third model in this experiment and builds on a widely adopted framework for vision–language pre-training [8]. BLIP2 follows a two stage design in which a frozen image encoder and a lightweight Q Former extract a compact visual representation that is then passed to a frozen large language model [8]. This structure makes BLIP2 suitable for multimodal classification, as the Q Former offers a stable and efficient feature interface while the language model provides strong contextual reasoning once the task specific head is fine tuned [1]. The corresponding architectural diagram is included in the appendix(Figure 17).

Since the expression classification task is relatively constrained and does not require the expressive power of very large language models, a moderate OPT model with 2.7 billion parameters was selected for the language model component [18]. This size provides sufficient semantic capacity for multimodal reasoning while keeping the computational cost of fine tuning manageable. To adapt BLIP2 for classification, a two layer classifier head was added on top of the BLIP2 output, and the number of hidden units (either 1024 or 512) was varied across experiments to balance learning capacity and generalization.

Several fine tuning methods were explored, including training only the Q Former and classifier head while keeping the encoder and language model frozen, as well as adding low rank adaptation (LoRA) [5] modules to the frozen OPT model and training LoRA alongside the Q Former and classifier head.Training the Q Former is particularly effective because it serves as the bottleneck between visual features and the language model, and its relatively small parameter count allows efficient task specific adaptation without modifying the much larger encoder or LLM components [7]. LoRA further enables lightweight refinement of the language model using a small number of low rank parameters, providing task specific flexibility without the cost of full transformer fine tuning [5].

2.4. Clip

Another model chosen for this experiment is CLIP ViT-B/32 [16]. This version is well suited for multimodal inputs, as it processes both an image and its corresponding text description. It also represents the lightest variant within the CLIP family, making it compatible with smaller GPUs and reasonably efficient on CPUs. The model includes a vision encoder and a text encoder that project into the same embedding dimension, enabling cosine similarity–based alignment.

3. Experiments and Results

3.1. LLaVA

We fine-tuned LLaVA-1.5-7B using Low-Rank Adaptation (LoRA) [6] for parameter-efficient adaptation. We applied LoRA adapters to four attention projection layers (q_proj, k_proj, v_proj, o_proj) in the language model, as these layers are critical for cross-modal understanding. We chose LoRA over full fine-tuning to manage memory constraints on our 32GB V100 GPU while maintaining adaptation flexibility. Our implementation builds on HuggingFace Transformers [17] and PEFT library [11], with custom modifications for label masking and conversation formatting.

Initial attempts with FP16 mixed precision training encountered gradient scaler conflicts when freezing/unfreezing parameters, causing "Attempting to unscale FP16 gradients" errors. We resolved this by loading models in FP32, applying LoRA and freezing strategies, then converting to FP16 for training. We also encountered memory constraints, addressed through gradient checkpointing, increased gradient accumulation (steps=8), and careful batch size management. Additionally, using `device_map="auto"` caused gradient computation issues; we fixed this by explicitly placing the model on a single device.

Zero-shot evaluation achieved 66.28% accuracy with

macro F1-score of 0.50, showing poor neutral class performance (F1: 0.08). We systematically explored hyperparameter space: LoRA ranks (r=8, r=16), scaling factors (alpha=16-64), learning rates (5e-5 to 5e-4), dropout (0.05-0.1), and fine-tuning strategies (LoRA-only vs. freezing vision components). Training used AdamW optimizer with cosine learning rate schedule, gradient accumulation (steps=4-8), and checkpointing.

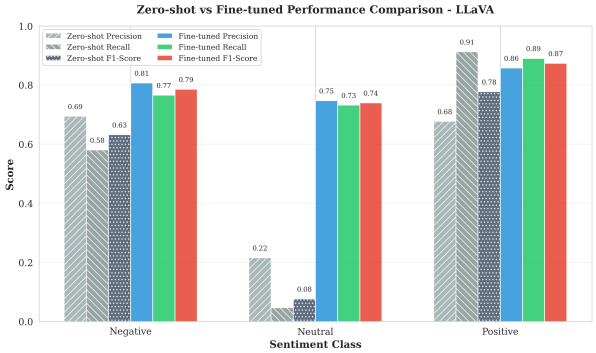

Figure 1. Zero-shot vs fine-tuned LLaVA: Precision, Recall, and F1-scores across sentiment classes.

Figure 1 shows balanced performance: positive (F1: 0.87), negative (F1: 0.79), neutral (F1: 0.74), with neutral improving from 0.08 to 0.74.

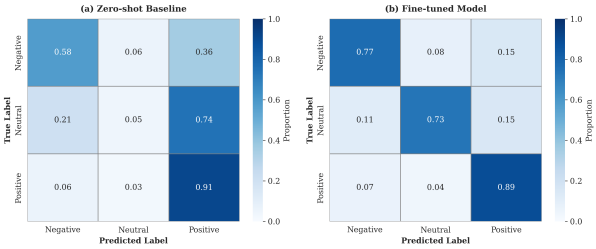

Figure 2. Confusion matrices: (a) Zero-shot baseline, (b) Fine-tuned model.

Figure 2 shows improved diagonal elements and reduced confusion, particularly for neutral class.

Configuration	Accuracy
Zero-shot	66.28%
LoRA (r=8, α=16)	74.37%
LoRA (r=16, α=32, lr=5e-5)	78.99%
LoRA (r=16, α=32, drop=0.1, lr=5e-4)	82.27%
Freeze Vision + LoRA	71.68%

Table 1. LLaVA fine-tuning experiments and results on MVSA-Single test set

Key findings: (1) LoRA-only fine-tuning (82.27%) significantly outperformed freeze-vision strategies (71-72%), indicating joint adaptation of vision-language connector and language model is crucial. (2) Increasing LoRA rank from r=8 to r=16 improved accuracy from 74.37% to 78.99%, demonstrating that higher capacity adapters better capture sentiment patterns. (3) Higher learning rate (5e-4) and dropout (0.1) further improved performance from 79.77% to 82.27%, showing that proper regularization and learning rate tuning enhance results. The best model achieved 82.27% accuracy (+15.99% over baseline) with macro F1-score of 0.80, demonstrating successful adaptation to sentiment analysis while maintaining computational efficiency.

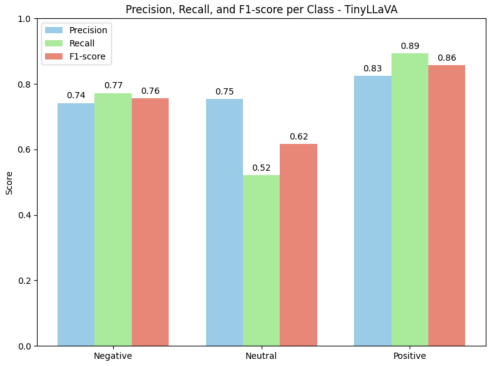

Figure 3. Per-Class Precision, Recall, and F1-Score – TinyLLaVA on MVSA-Single

3.2. TinyLLaVA

The TinyLLaVA model, benchmarked (Figure 4) on 2000 randomly selected MVSA-Single samples, achieved a weighted F1-score of 0.2922, reflecting limited zero-shot sentiment classification performance. The confusion matrix highlights frequent misclassifications, particularly with Neutral and Positive samples often predicted as Positive. This suggests the model has difficulty distinguishing sentiment classes in the absence of fine-tuning.

We used PyTorch framework in this deep learning workflow.We fine-tuned TinyLLaVA using a parameter-efficient approach with LoRA adapters rather than full model training. In our workflow, the pretrained backbone (bczhou/tiny-llava-v1-hf) from HuggingFace was loaded and frozen so its weights remained unchanged, while lightweight LoRA modules were inserted into the attention projection layers (q_proj and v_proj) to allow targeted adaptation. These adapters were configured with low rank (r=2), dropout (0.3), and scaling (lora_alpha=16) to balance efficiency and regularization. On top of this adapted backbone, we added a classifier head consisting of dropout and a linear layer to map pooled hidden states into three sentiment classes (negative, neutral, positive). This setup let us leverage HuggingFace's pretrained multimodal model while only fine-tuning

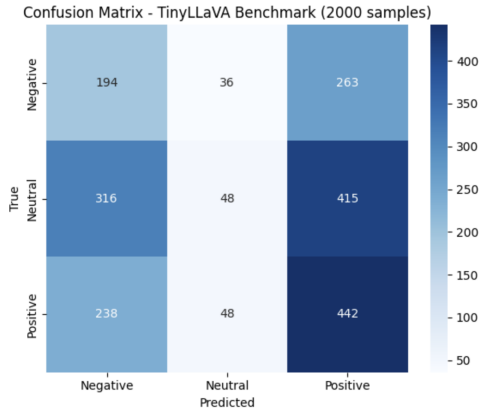

Figure 4. TinyLLaVA Benchmark

a small fraction of parameters, making training faster, less memory-intensive, and well-suited for our GPU resources.

Training uses the AdamW optimizer with a learning rate of 5e-6 and weight decay of 1e-4. The loss function is cross-entropy with label smoothing.Each epoch consists of a training phase, where forward passes, backpropagation, optimizer steps, and gradient resets are performed, and a validation phase, where evaluation is done without gradient updates. Early stopping monitors validation loss with patience set to 4, restoring the best weights if no improvement occurs.

The training and validation loss curves (Figure 5)both show a consistent downward trend over 10 epochs, indicating effective learning. The gap between the two curves remains modest, suggesting good generalization without overfitting. This supports the stability and robustness of the TinyLLaVA model during fine-tuning.

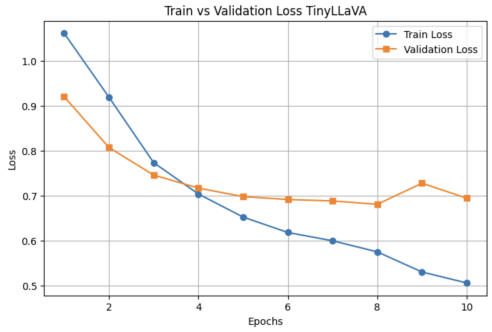

Figure 5. Training vs. Validation Loss – TinyLLaVA Sentiment Classification on MVSA-Single

The confusion matrix (Figure 6) shows TinyLLaVA performs well overall, with strong accuracy for Neutral and Positive classes, but more misclassifications for Negative, often predicted as Neutral. This indicates effective multimodal fusion for expressive sentiments, while subtle cues in Negative samples remain challenging, suggesting a bias toward Neutral when uncertain.

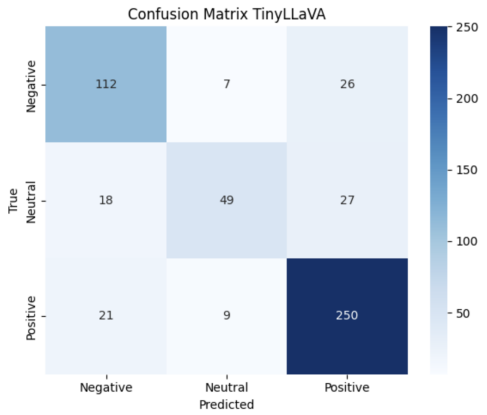

Figure 6. Confusion Matrix – TinyLLaVA Sentiment Classification on MVSA-Single

The final test evaluation of TinyLLaVA on the MVSA-Single agreement subset demonstrates strong overall performance (Figure 3), achieving a weighted F1-score of 0.7857 and an accuracy of 79% across 519 samples. The model performs best on the Positive class (F1 = 0.86), with both high precision (0.83) and recall (0.89), followed by the Negative class (F1 = 0.76), which shows balanced precision and recall. In contrast, the Neutral class remains the most challenging, with a lower F1-score of 0.62 due to reduced recall (0.52), indicating difficulty in correctly identifying neutral sentiment. The macro averages (F1 = 0.74) highlight this imbalance, while the weighted averages (F1 = 0.79) reflect the dominance of the positive class in the dataset. Overall, these results confirm that the model generalizes well on clear sentiment cues but struggles with ambiguous neutral cases, a common challenge in multimodal sentiment analysis.

3.3. BLIP-2

To ensure consistency, the zero-shot model used the same classifier head as the fine-tuned models, with all classifier parameters randomly initialized. When the pre-trained BLIP-2 model was applied in a simple prompt-based zero-shot setting, its test accuracy remained low (19%). We therefore selected the highest-performing configuration as our zero-shot reference for the remainder of our analysis. The dataset was imbalanced, with more than 60% of samples belonging to a single class, which affected both the optimization process and the per-class performance observed across all BLIP-2 experiments.

3.3.1 Experiment 1: Training the Q-Former with a 512-Unit Classifier Head

In the first configuration, a classifier head with 512 hidden units was added while the image encoder and OPT language model were frozen, and only the Q-Former and classifier head were trained. This setup improved zero-shot accuracy by 44% after six epochs, reaching approximately 75% test accuracy.

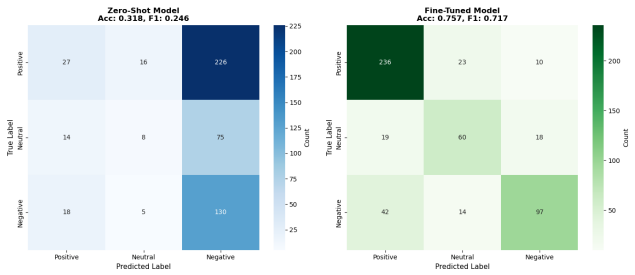

Figure 7. Confusion matrix for Experiment 1.

Most gains occurred in the first epoch, after which validation accuracy oscillated and plateaued, reflecting the limited adaptation capacity of the Q-Former bottleneck when the backbone is frozen.(Figure 19) Adjusting the learning rate and dropout delayed this plateau but did not change the final accuracy, and higher dropout slightly reduced performance due to underfitting. Increasing batch size introduced additional training instability without improving accuracy.

The dataset imbalance also shaped outcomes: the minority class (less than 18% of training samples) reached roughly 62% accuracy, while the majority class achieved accuracy and F1-scores near 80%, consistent with a classifier biased toward the dominant class.(Figure 18)

3.3.2 Experiment 2: LoRA-Adapted OPT with a 512-Unit Classifier Head and Trainable Q-Former

In the second configuration, standard LoRA [5] was applied to the OPT language model to allow the text representation to adapt to the classification task, while the image encoder remained frozen. The Q-Former and the 512-unit classifier head were kept trainable, and the same data split as in Experiment 1 was used. Standard LoRA was selected because the task is moderately complex and computational resources were sufficient, making more specialized variants such as DoRA or QLoRA unnecessary in this context. With a moderate LoRA capacity (rank $r = 8$, $\alpha_{\text{lora}} = 16$), test accuracy improved from 31.8% in the zero-shot baseline to 84.6%. Reducing LoRA capacity (e.g., $r = 4$, $\alpha_{\text{lora}} = 8$) lowered overall accuracy by roughly 4 percentage points and reduced minority-class performance by about 10 points, indicating insufficient adaptation. Increasing LoRA capacity beyond the moderate range reduced accuracy by approximately 10 points, suggesting early overfitting in the imbalanced setting. Training behaviour was similar to Experiment 1, with rapid gains in the initial epochs followed by a stable validation plateau. Increasing dropout or weight decay delayed the onset of this plateau but did not improve the final validation accuracy, and the overfitting pattern remained present.

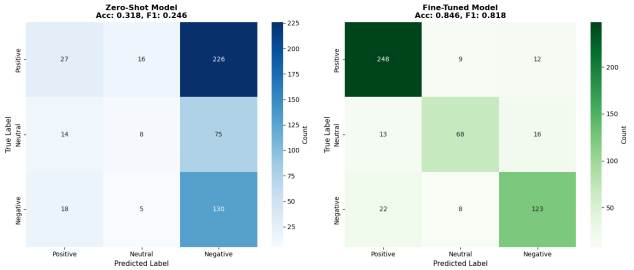

Figure 8. Confusion matrix for Experiment 2.

3.3.3 Experiment 3: Increased Classifier Capacity with Regularized Training

In the third experiment, the classifier head was redesigned from the single-layer structure used in Experiment 2 (one 1024-unit hidden layer) to a deeper two-layer configuration with 521 and 256 hidden units, respectively. The activation function was changed from ReLU to GELU, and an additional dropout layer was introduced between the new hidden layers to increase regularization. The rest of the model configuration followed Experiment 2. To mitigate the increased risk of overfitting associated with the deeper classifier, dropout was raised from 0.1 to 0.5 and weight decay from 0.01 to 0.1, and the LoRA capacity was reduced to a lighter configuration ($r = 4$, $\alpha_{\text{lora}} = 8$). These adjustments were intended to preserve training stability while allowing the expanded classifier head to capture finer class-specific distinctions. .

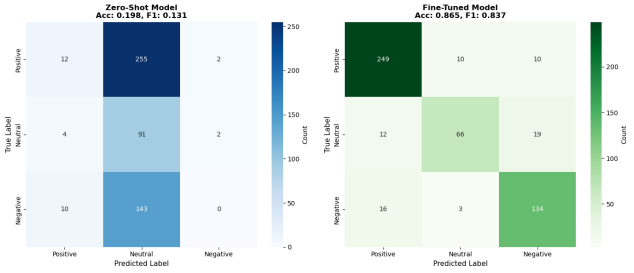

Figure 9. Confusion matrix for Experiment 3.

Under this configuration, the best model achieved an overall accuracy of 86%, representing the strongest perfor-

mance across all experiments. These findings indicate that, for this multimodal classification task, increasing classifier capacity can be effective when combined with stronger regularization and moderate LoRA adaptation, , enabling the model to benefit from additional representational flexibility while limiting excessive overfitting.

3.4. Clip

We determined the baseline of the clip model on the MVSA dataset by just testing the model on 1000 examples and getting the accuracy, f1-scores, recall, and precision. The accuracy for the baseline with parameters frozen was .4166. The f1-scores are pretty low as seen in (figure.10). The baseline also was pretty confused on the the negative and neutral examples but did better on positive examples (Figure 11)

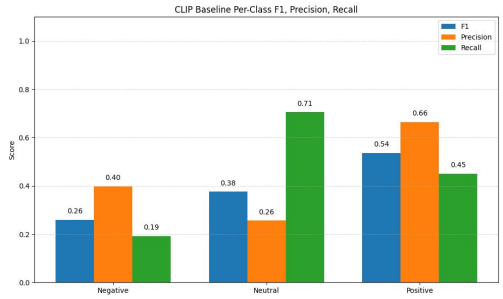

Figure 10. Per-Class Precision, Recall, and F1-Score – Baseline CLIP on MVSA-Single

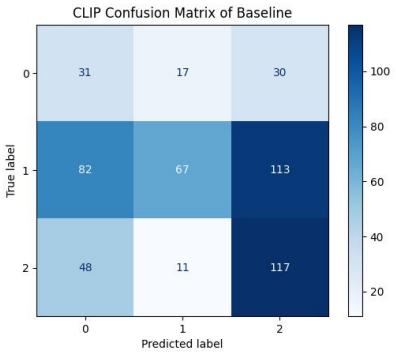

Figure 11. Confusion Matrix – baseline CLIP Sentiment Classification on MVSA-Single

Then for the first step of performance improvements on the data we added a classifier head to the model. Trained it for 10 epochs on 60% of the data and did validation testing for each epoch. Using AdamW optimizer with a learning rate of .001. This increased the test set accuracy to .7848. The metric scores drastically improved also and so did the confusion matrix. (Reference Figures 12 13 14)

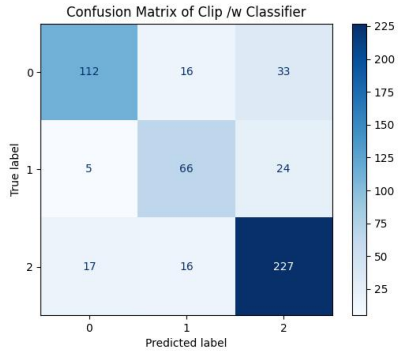

Figure 12. Confusion Matrix – classifier CLIP Head Sentiment Classification on MVSA-Single

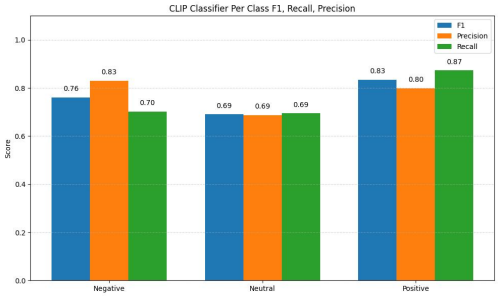

Figure 13. Per-Class Precision, Recall, and F1-Score – Classifier Head CLIP on MVSA-Single

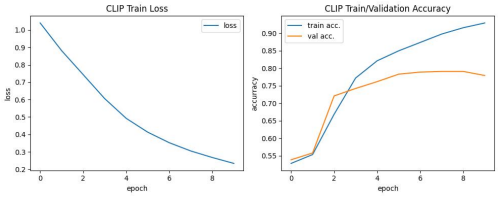

Figure 14. Training Results Classifier Head CLIP on MVSA-Single

Finally, for the next step of improvements on the data we added LoRA adapters to the classifier clip model to try and boost accuracy on test data more. I tried training this model with more than 5 epochs but it started to overfit the data. The validation accuracy started to decrease. This didn't increase accuracy, metric scores, nor confusion on new data enough for the increase of model size. The accuracy only increased to .812 form .7848 from the classifier head only model. The LoRA configuration for this was r=1, lora_alpha=1, targeted only the self attention layers q_proj and v_proj, and a dropout of .05. I tried increases r and alpha but the performance improved wasn't enough to justify it.

model	Accuracy
Clip baseline	.4166
Clip w/ Classifier	.7848
Clip w/ classifier and LoRA	.812

Table 2. Accuracy across Clip enhancements

Through the experiments on the baseline, classifier, and LoRA + classifier. The test set accuracy was .4166, .7848, and .812 respectively as in (Table 2).

4. Future Work

Several independent avenues can extend this work.

Dataset complexity. Future studies may move beyond three-class tone classification by using datasets such as MVSA-Multiple, which introduce multi-label sentiment and emotion categories. This also creates opportunities to evaluate text and image augmentation strategies.

LoRA design in TinyLLaVA. TinyLLaVA currently applies LoRA only to `q_proj` and `v_proj`. Extending adaptation to `k_proj` and `o_proj` could provide richer control over the attention mechanism and potentially improve multimodal sentiment modeling.

Training stability for BLIP-2 with imbalanced data. BLIP-2 exhibited early plateau and susceptibility to imbalance-driven overfitting. Techniques such as focal loss, curriculum rebalancing, or targeted sampling could support more stable and generalizable optimization.

References

[1] W. Dai, R. Xu, K. Zhang, S. Yan, S. Xu, et al. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023. 2

[2] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *arXiv:1803.09010*, 2021. Published in Communications of the ACM. 8

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014. 1

[4] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, prefix=de useprefix=false family=Laroussilhe, given=Quentin, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. 1

[5] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1, 2, 5

[6] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. 2

[7] Sungkyung Kim, Adam Lee, Junyoung Park, Andrew Chung, Jusang Oh, and Jay-Yoon Lee. Towards efficient visual-language alignment of the q-former for visual reasoning tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, page 15155–15165, 2024. 2

[8] Jiasen Li, Dongxu Li, Junnan Li, Zhecan Zhang, Weijie Su, Xingchao Yang, Linjie Zhang, Jue Yang, Luowei Yuan, Ce Liu, et al. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. 1, 2

[9] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual Instruction Tuning. 2

[10] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023. 1

[11] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022. 2

[12] Vince Marcs. Mvsa-single. https://www.kaggle.com/datasets/vincemarcs/mvsasingle, 2016. Sentiment Analysis on Multi-view Social Data. Contains 4869 image-text pairs with sentiment labels. 1

[13] T. Niu, S. Zhu, L. Pang, and A. El Saddik. Sentiment analysis on multi-view social data. *MultiMedia Modeling*, Part II:15–27, 2016. 1

[14] Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 873–883, 2017. 1

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021. 1

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamila Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv:2103.00020*, 2021. Published in ICML. 2

[17] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing. (arXiv:1910.03771), July 2020. 2

[18] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Oanh Dewan, Mona Diab,

Student Name	Contributed Aspects	Details
All	Report	Discuss, write and review
Fattaneh AmeriMahabadian	Project report template	Title, Abstract, References, contributions table
	Review datasheet dataset paper	Write datasheet [2] for dataset MVSA-Single
	Coding TinyLLaVA model	Benchmarking, fine tuning and experiments
	Write the report	Model TinyLLaVA experiments
Mark Lights	Initiate GitHub account	CLIP starter code
	Coding CLIP model	Benchmarking, fine tuning and experiments
	Write the report	Model CLIP experiments
Yanzhe Yang	Coding LLaVA model	Benchmarking, fine tuning and experiments
	Write the report	Model LLaVA experiments, results, background
	Proposal	Write the Project Proposal
Mehrnaz Hooshmand	Coding BLIP-2 model	Benchmarking, fine tuning and experiments
	Write the report	BLIP 2 approach and discussion, future work, Background

Table 3. Contributions of team members.

Emily Dinan, Joan Girbau, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 2

[19] B. Zhou. tiny-llava-v1-hf, 2025. https://huggingface.co/bczhou/tiny-llava-v1-hf. 2

[20] B. Zhou. Tinyllava: A framework of small-scale large multimodal models, 2025. Architecture figure (Section 3, Figure 2) available at https://arxiv.org/html/2402.14289v1#S3.F2. 8

A. Appendix

B. Project Code Repository

The GitHub repository we used as the starter code is: https://github.com/OpenAI/CLIP

The Github repository for our final project is at: https://github.gatech.edu/mlights3/DeepLearningFinalProject

C. TinyLLaVA Architecture

The architecture of TinyLLaVA [20] (Figure 15) consists of a small-scale (size 1.1B) LLM F_θ, a vision encoder V_ϕ, and a connector P_φ, where θ, ϕ, and φ are the learnable parameters respectively. This architecture can model various multimodal understanding tasks that take as input a pair of image and text sequence and output a text sequence.

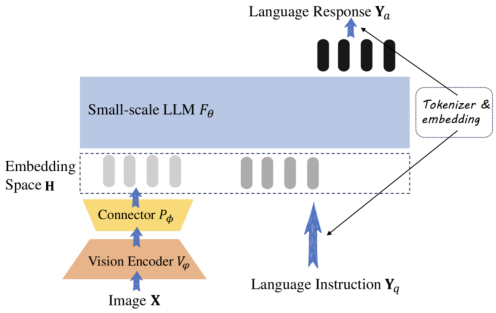

Figure 15. TinyLLaVA Architecture

D. Training vs. Validation Loss TinyLLaVA for the full dataset MVSA-Single

The training vs. validation loss (Figure 16) for the full dataset size of 4689 samples from MVSA-Single demon-

strates that fine-tuning helps mitigate overfitting.

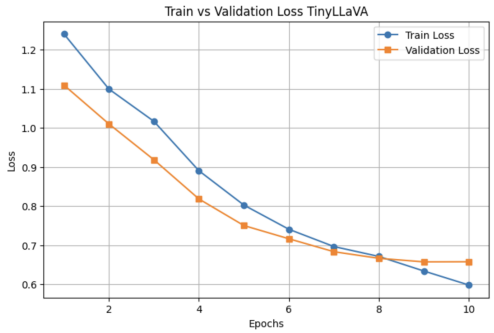

Figure 16. Training vs. Validation Loss – TinyLLaVA Sentiment Classification on MVSA-Single

E. BLIP-2

E.1. BLIP-2 Architecture

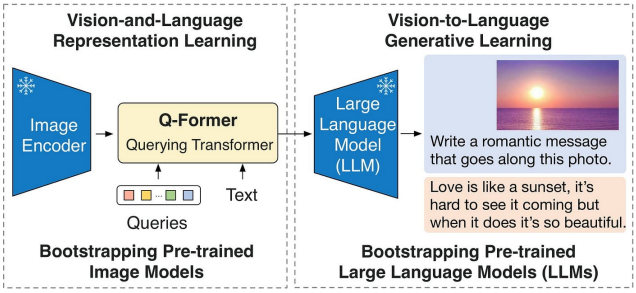

Figure 17. BLIP-2 Framework

E.1.1 BLIP-2 Experiment 1

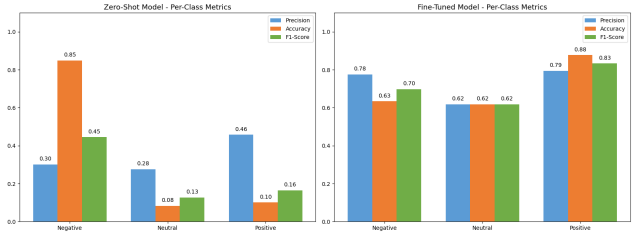

Figure 18. Per-Class Metrics for Experiment 1.

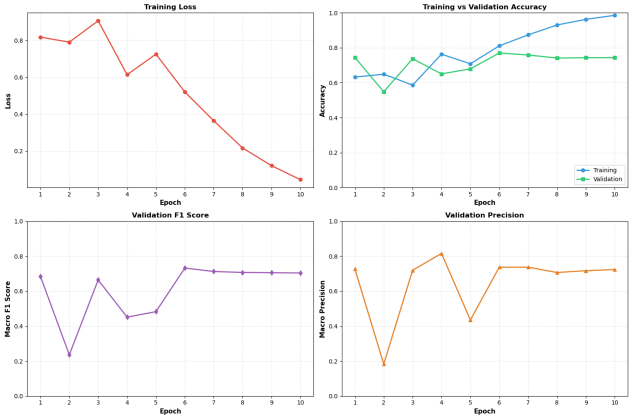

Figure 19. Training Metrics per epoch for Experiment 1.

E.1.2 BLIP-2 Experiment 2

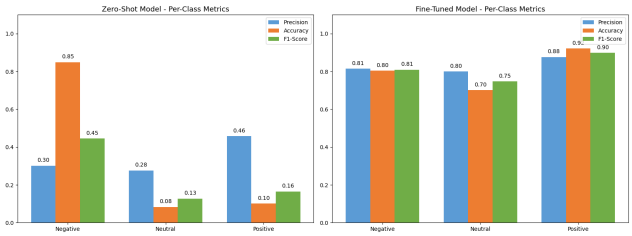

Figure 20. Per-Class Metrics for Experiment 2.

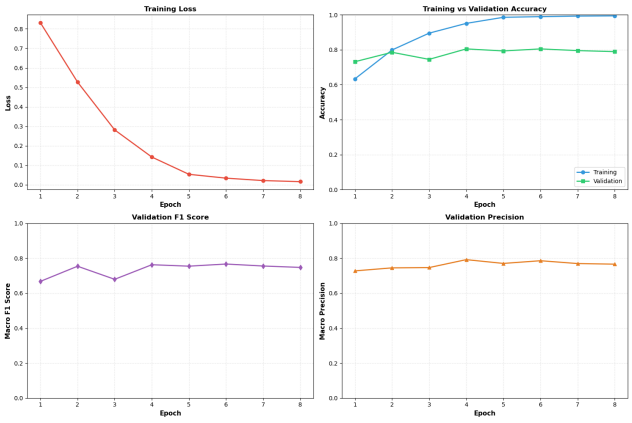

Figure 21. Training Metrics per epoch for Experiment 2.

E.1.3 BLIP-2 Experiment 3

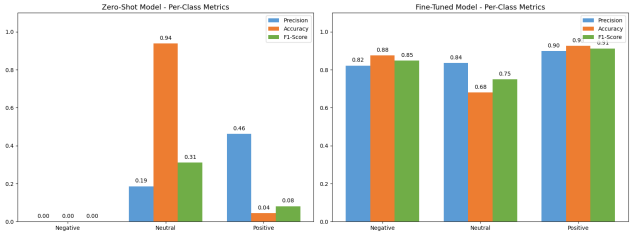

Figure 22. Per-Class Metrics for Experiment 3.

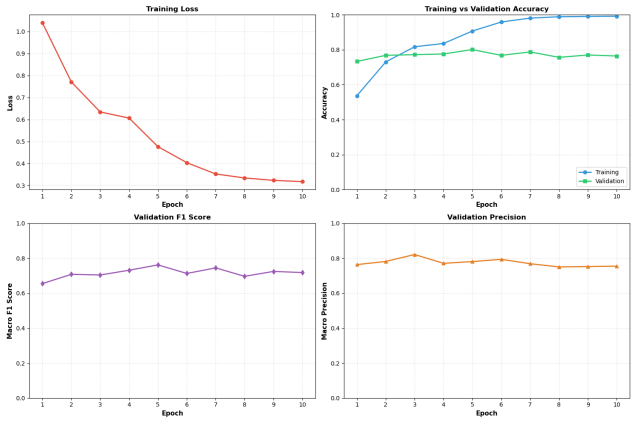

Figure 23. Training Metrics per epoch for Experiment 3.

F. CLIP Architecture

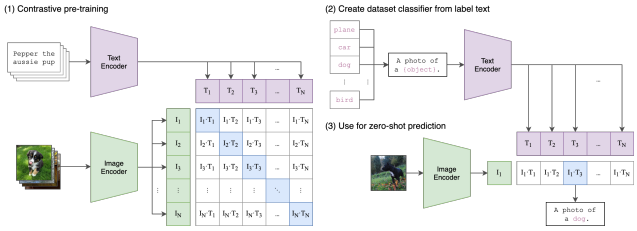

Figure 24. CLIP Architecture