CROSS MODAL DOMAIN ADAPTATION

10-701 FINAL REPORT

JS Dandurand[*]
Department of Machine Learning
Carnegie Mellon University
Pittsburgh, PA 15213
jdandura@andrew.cmu.edu

Husain Raja[*]
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
horaja@cs.cmu.edu

Kaylee Xu[*]
Department of Machine Learning
Carnegie Mellon University
Pittsburgh, PA 15213
kayleex@cs.cmu.edu

January 8, 2026

1 Introduction

Modern machine learning models are usually built for a single data modality, language models for text, and vision models for images. However, recent studies suggest that the representational power of large language models (LLMs) may be effectively applied beyond text. This opens the possibility of adapting LLMs to process non-text data, such as images, without retraining them from scratch. Our project investigates whether pretrained language models can be effectively adapted for image classification (on CIFAR10) through learned embedding and projection layers. Specifically, our objective is to determine whether the attention and feature extraction mechanisms learned from text can generalize to visual inputs when images are converted into token-like representations. By comparing simple frozen-backbone adaptation methods against more advanced distribution alignment and parameter-efficient fine-tuning techniques, we will assess the viability of cross-modal transfer from language to vision.

2 Literature Review

Research on cross-modal learning has increasingly explored whether pretrained language models (LLMs) can be repurposed for non-text modalities, particularly vision. Early multimodal work focused on learning shared embedding spaces between images and text, enabling models trained on one modality to generalize to tasks in the other. A seminal example is CLIP Radford et al. [2021], which jointly trains an image encoder and a text encoder using a large-scale contrastive objective. By aligning visual and textual representations, CLIP demonstrated strong zero-shot classification capabilities, as well as a surprising transfer to a wide range of vision tasks beyond. This shows that language-supervised embedding spaces captures rich, generalizable visual concepts. It establishes the significance of well-aligned cross-modal embedding spaces, motivating later work, including our own, that aims to align non-textual inputs to a pretrained language model's embedding space without expensive multimodal pretraining.

Complementing this, the *Frozen Pretrained Transformer* (FPT), introduced by Lu et al. [2021], shows that a transformer pretrained on natural language can serve as a universal processors for sequential data in diverse modalities. Their cross-domain experiments (e.g., vision, bit tasks, and homology) found that the frozen transformer, with small trainable input/output projection layers, can achieve comparable accuracy to a transformer trained from scratch on those tasks. This result suggest that the representation learned by self-attention and feedforward layers on natural language data carries a kind of general-purpose inductive bias, which is enough to process generic sequences regardless of modality. Thus, this framework strongly informs our baseline method, where we split an image into patches and then feed into a frozen LLM using small trainable embedding layers and classification head.

Building upon the use of LLMs as universal computation engines, Hao et al. [2022] propose treating LLMs as general-purpose reasoning modules that can interface with modality-specific encoders . their MetaLM architecture attaches pretrained, modality-specific encoders (e.g. for vision or audio) to a frozen LLM backbone and trains via a semi-causal language-modeling objective. Their competitive results across both language-only and vision-language benchmarks

[*] Equal Contribution

provides conceptual evidence that LLM internal representations can support non-text modalities as general reasoning modules, once properly aligned, which further strengthens the theoretical premise behind our design.

Another recent work in cross-modal alignment is highly related to our method. Shen et al. propose ORCA Shen et al. [2023], a two-stage "Align then Refine" paradigm. In the alignment stage, a trainable embedding network maps the target modality (e.g., images) into the feature distribution of the language model's native embedding space, optimized using distance metrics such as Maximum Mean Discrepancy (MMD) or Optimal Transport Dataset Distance (OTDD). In the refinement stage, the model is trained on the downstream task (e.g image classification). Their experiments across 10 different datasets, with modalities ranging from images to sound show that explicitly reducing the distribution gap between source and target embeddings can improve cross-modal transfer performance. This motivates our extension to the FPT baseline, we incorporate an alignment phase using MSE and MMD to better match image patch embeddings to the LLM's pretrained embedding space before fine-tuning.

Another promising direction involves aligning image inputs to LLMs at the *token* level. Zhu et al. propose a Vision-to-Language (V2L) tokenizer Zhu et al. [2024], which converts images into sequences of discrete tokens drawn from the LLM's vocabulary. A frozen LLM can then process these tokens auto-regressively, enabling tasks such as image classification and caption generation without updating the LLM's parameters. Compared to feature-level alignment, token-level alignment ensures that the LLM receives inputs in a form consistent with its pretraining, which may enhance semantic compatibility. However, this requires a learned visual tokenizer, which is more complex than the continuous embedding layers used in our method. The V2L approach illustrates a different but complementary perspective, that careful input-space alignment alone can unlock visual capabilities in LLMs.

Finally, parameter-efficient fine-tuning (PEFT) has become a widely adopted strategy for adapting large models without modifying most of their parameters. LoRA Hu et al. [2021], for example, adds low-rank adaptation matrices into the attention layers of a frozen transformer. Despite dramatically reducing the number of trained parameters (less than 1% of the model parameters), LoRA often matches or even exceeds the performance of full fine-tuning. PEFT techniques provide a natural extension to our project, after alignment, we can apply LoRA to selectively refine the internal representations of the frozen LLM with minimal computational overhead, combining efficiency and adaptability in a lightweight, practical way.

In summary, the literature indicates two converging insights. First, modality gaps can be bridged in either embedding space (e.g., FPT, ORCA) or token space (e.g., V2L), with each approach offering different benefits in terms of simplicity, semantic alignment, and performance. Second, large pretrained language models offer strong general-purpose priors that extend to vision tasks when equipped with small adapter or alignment modules. Our project positions itself within this landscape by combining the simplicity of FPT with the stronger theoretical guarantees of distribution alignment and the practical benefits of parameter-efficient tuning. The reviewed works collectively inform our methodological choices and highlight key challenges, particularly alignment quality, that our proposed extensions aim to address.

3 Experimental Results

3.1 Experimental Setup

We evaluate our cross-modal adaptation methods on CIFAR-10, a standard benchmark for image classification consisting of 32×32 RGB images across 10 classes. Our experiments systematically compare multiple training paradigms: (1) a frozen-backbone baseline (FPT), (2) distribution alignment followed by FPT training, (3) full parameter fine-tuning with and without alignment, and (4) LoRA-based fine-tuning with and without alignment.

3.1.1 Dataset and Preprocessing

The CIFAR-10 dataset Krizhevsky et al. [2009] contains 50,000 training and 10,000 test images. Following best practices for hyperparameter tuning, we reserve 20% of the training set (10,000 images) as a held-out validation set, ensuring that test set performance is only evaluated on the final best configuration. Data augmentation is performed on the training data with random horizontal flips (probability 0.5), random cropping with 4-pixel padding, and random rotations up to 20 degrees. All images are normalized using CIFAR-10 statistics: mean (0.4914, 0.4822, 0.4465) and standard deviation (0.2023, 0.1994, 0.2010).

3.1.2 Model Architecture

We use TinyLlama-110M Zhang et al. [2024] as our pretrained language model backbone. Images are tokenized into 4×4 patches, resulting in 64 patches per 32×32 image. Each patch is linearly projected to the model's embedding dimension (784 for TinyLlama-110M). Learnable positional encodings are added to patch embeddings before being passed to the transformer backbone. The model uses mean pooling over all patch tokens, followed by a dropout layer and a linear classification head mapping to 10 classes.

3.1.3 Training Paradigms

We evaluate four distinct training paradigms:

FPT (Frozen Pretrained Transformer) Baseline: Only layer normalization parameters and newly introduced components (patch embedding, positional encoding, classification head) are trainable, keeping the transformer backbone completely frozen. This establishes a lower bound for cross-modal transfer performance.

FPT with Distribution Alignment: Prior to task-specific training, we perform an embedding alignment stage where the patch embedding and positional encoding layers are optimized to minimize distribution divergence between image patch embeddings and text token embeddings sampled from the pretrained model's vocabulary. This stage runs for 15-20 epochs using one of three distance metrics: Mean Squared Error (MSE), Cosine Distance, or Maximum Mean Discrepancy (MMD).

Full Parameter Fine-tuning: All model parameters are updated during training. This includes both configurations with and without the distribution alignment stage, allowing us to assess whether alignment provides complementary benefits when full model capacity is available.

LoRA Fine-tuning: We apply Low-Rank Adaptation (LoRA) Hu et al. [2021] to the attention layers (query, key, value, and output projections) as well as the MLP layers (up and down projections) while keeping the base model frozen. LoRA introduces trainable low-rank matrices that approximate weight updates, significantly reducing the number of trainable parameters. We explore LoRA ranks of 8, 16, and 32, with alpha scaling factors of 8, 16, 32, and 64, and dropout rates of 0.0, 0.1, and 0.2.

3.1.4 Hyperparameter Tuning with ASHA

To ensure fair comparison and optimal performance for each method, we employ the Asynchronous Successive Halving Algorithm (ASHA) Li et al. [2020] for hyperparameter optimization. ASHA is a principled early-stopping method that progressively eliminates poor configurations by training them for fewer epochs at early stages (rungs) and only continuing promising configurations to full training. This approach is computationally efficient while maintaining statistical rigor.

For each training paradigm, we define a hyperparameter search space:

- **Learning rate:** 10^{-5} to 10^{-3} (log-uniform sampling)
- **Batch size:** 16, 32, 64, or 128
- **Weight decay:** 0, 10^{-4}, or 10^{-2}
- **Optimizer:** Adam or AdamW
- **Learning rate scheduler:** Cosine annealing or linear decay
- **Dropout rate:** 0.0, 0.1, or 0.2
- **Alignment learning rate:** 10^{-5}, 10^{-4}, or 5×10^{-4} (when alignment is enabled)
- **Alignment batch size:** 8, 16, or 32 (when alignment is enabled)
- **LoRA-specific:** rank $\in \{4, 8, 16, 32\}$, alpha $\in \{8, 16, 32, 64\}$, dropout $\in \{0.0, 0.1, 0.2\}$ (for LoRA mode only)
- **Max Gradient Norm (Grad Clipping):** 1

ASHA is configured with 4 rungs and a reduction factor of 2, meaning that at each rung, only the top 50% of configurations are promoted to the next rung. Early rungs train for fewer epochs (scaled proportionally), while the final rung trains for the full 50 task epochs. All hyperparameter selection is performed exclusively on the validation set, with the test set reserved for final evaluation of the best configuration.

3.1.5 Distribution Alignment Implementation

When alignment is enabled, we first train the patch embedding and positional encoding layers to minimize distribution divergence. We sample 50,000 random text token sequences from the pretrained model's vocabulary and compare two distance metrics:

- **MSE:** Mean Squared Error between mean-pooled embeddings
- **MMD:** Maximum Mean Discrepancy using Gaussian kernels with multiple bandwidths

Theoretical Motivation: Both metrics align distributions, but at different levels. MSE minimizes the L_2 distance between the first moments (means) of the image and text embedding distributions. By minimizing $\|\mu_X - \mu_Y\|^2$ where μ_X and μ_Y are the mean embeddings, we ensure the image embeddings are centered similarly to text embeddings in the shared space. However, MSE only captures mean alignment and ignores higher-order statistics.

MMD provides a more comprehensive alignment by comparing distributions in a reproducing kernel Hilbert space (RKHS). The MMD between distributions P and Q is defined as:

$$\text{MMD}^2(P, Q) = \mathbb{E}_{x,x' \sim P}[k(x, x')] + \mathbb{E}_{y,y' \sim Q}[k(y, y')] - 2\mathbb{E}_{x \sim P, y \sim Q}[k(x, y)]$$

where $k(\cdot, \cdot)$ is a Gaussian kernel. When MMD is zero, the distributions are identical. By using multi-scale Gaussian kernels with different bandwidths, MMD captures not only mean alignment but also variance and higher-order moments, providing a more robust distribution matching that better preserves the structure of the text embedding space.

The alignment stage runs for 15-20 epochs (tuned via ASHA) before task-specific training begins. Only the embedding layers are updated during alignment; the transformer backbone remains frozen.

Note: The code implementation for MMD is adapted from Shen et al. [2023]. We cite the official implementation: `https://github.com/sjunhongshen/ORCA`

3.1.6 Evaluation Metrics

We report test set accuracy as our primary performance metric, defined as the percentage of correctly classified images. We also track training and validation loss (cross-entropy) throughout training to monitor convergence and overfitting. All experiments are logged using MLflow for reproducibility and visualization of training curves.

3.2 Results

3.2.1 Baseline Performance (FPT without Alignment)

Our frozen-backbone FPT baseline achieves 67.06% test accuracy on CIFAR-10 after 50 epochs of training. This establishes a lower bound for cross-modal transfer, demonstrating that even with minimal adaptation (only layer norms and embedding layers trainable), the pretrained language model can extract useful features for image classification. The baseline performance is much lower than vision-specific models (e.g., ResNet achieves ∼95% on CIFAR-10), which is expected given that we are adapting a language model rather than training a vision model from scratch. However, the fact that performance significantly exceeds random chance (10%) suggests that transformer architectures learn generalizable attention patterns that transfer across modalities when appropriate embedding interfaces are provided.

Table 1: Test accuracy comparison across methods.

Method	Test Accuracy (%)	Δ from Baseline
FPT Baseline	67.06	–
FPT + Alignment (MSE)	66.74	-0.32
FPT + Alignment (MMD)	66.96	-0.10
Full Fine-tuning	85.26	18.20
Full Fine-tuning + Alignment (MSE)	85.48	18.42
Full Fine-tuning + Alignment (MMD)	87.39	20.33
LoRA Fine-tuning	81.42	14.36
LoRA Fine-tuning + Alignment (MSE)	81.87	14.81
LoRA Fine-tuning + Alignment (MMD)	83.86	16.80

3.2.2 Effect of Distribution Alignment on FPT

Table 1 compares performance across different alignment strategies applied to all of the finetuning methods. We observe that embedding alignment applied to the FPT baseline does not achieve a noticable improvement with either distance metrics, with the unaligned baseline actually achieving the highest test accuracy at 67.06%. This results suggests that the frozen transformer backbone already provides representations that are sufficiently compatible with the downstream task, and the limited trainable parameters (only layer norms and the classification head) cannot effectively leverage the additional alignment signal. Alternatively, the alignment may be operating at a representation level that does not interact meaningfully with the frozen backbone's fixed feature space, indicating that FPT's constraint of keeping the backbone frozen limits the potential benefits of cross-modal alignment.

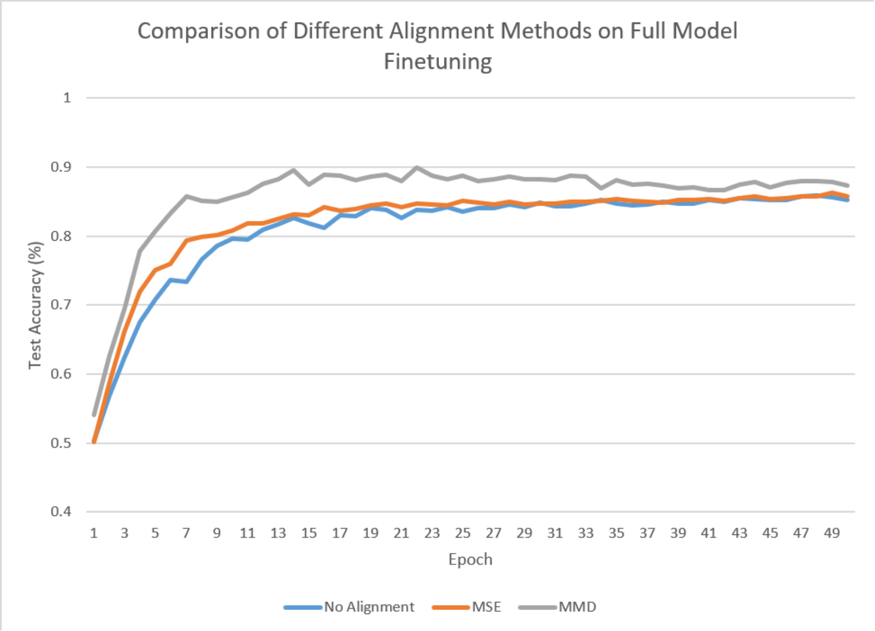

Figure 1: Alignment Comparison Under Full Fine-tuning

3.2.3 Full Parameter Fine-tuning

Full fine-tuning of all model parameters achieves 85.26% test accuracy, representing a 18.20% improvement over the FPT baseline. When combined with distribution alignment, performance further improves to 85.48%, indicating that finetuning the entire model allows it to better take advantage of the pretrained embedding alignment. These results are consistent with Shen et al. [2023], which came to a similar conclusion.

However, Figure 1 shows that this improvement from distribution alignment pretraining comes from the MMD distance metric, while the MSE metric only provides a minor increase in accuracy as well as slightly faster convergence in the early epochs of training.

Finally, full fine-tuning experiments exhibit clear overfitting, as shown by the U-shaped test loss curve. As observable in Figure 2, while test accuracy plateaus around 85%, test loss decreases for the first 15-20 epochs before steadily rising, indicating the model becomes overconfident on correct predictions while increasing loss on misclassified examples. This behavior stems from the mismatch between model capacity and dataset complexity: TinyLlama-110M's 100+ million parameters far exceed what CIFAR-10's 50,000 training images require, causing the model to memorize rather than generalize. On larger benchmarks like ImageNet, which consist of a larger number of classes for classification, we would expect this overfitting gap to narrow as dataset complexity better matches model capacity.

3.2.4 LoRA Fine-tuning Results

LoRA fine-tuning provides an attractive middle ground between FPT and full fine-tuning. With optimal hyperparameters (rank=16, alpha=32, dropout=0.1), LoRA achieves 81.42% test accuracy while updating only about 2 million parameters (approximately 2% of the model). This represents over 14 percentage points improvement over FPT baseline while requiring over 50× fewer trainable parameters than full fine-tuning.

When combined with distribution alignment, LoRA performance improves to 81.87% (MSE) and 83.86% (MMD), demonstrating that alignment benefits are preserved in the parameter-efficient setting. Similar to full model fine-tuning, we observe that MMD is much more successful in improving model performance than MSE. The performance gap between LoRA+MMD Alignment and Full Fine-tuning+MMD Alignment is 3.53 percentage points, suggesting that LoRA can capture most of the benefits of full fine-tuning at a fraction of the computational cost.

Lastly, a noteworthy observation is that LoRA methods show convergence behavior similar to FPT but with much lower loss and higher accuracy, rather than the aggressive overfitting observable in full model fine-tuning, likely due to the constrained parameter space.

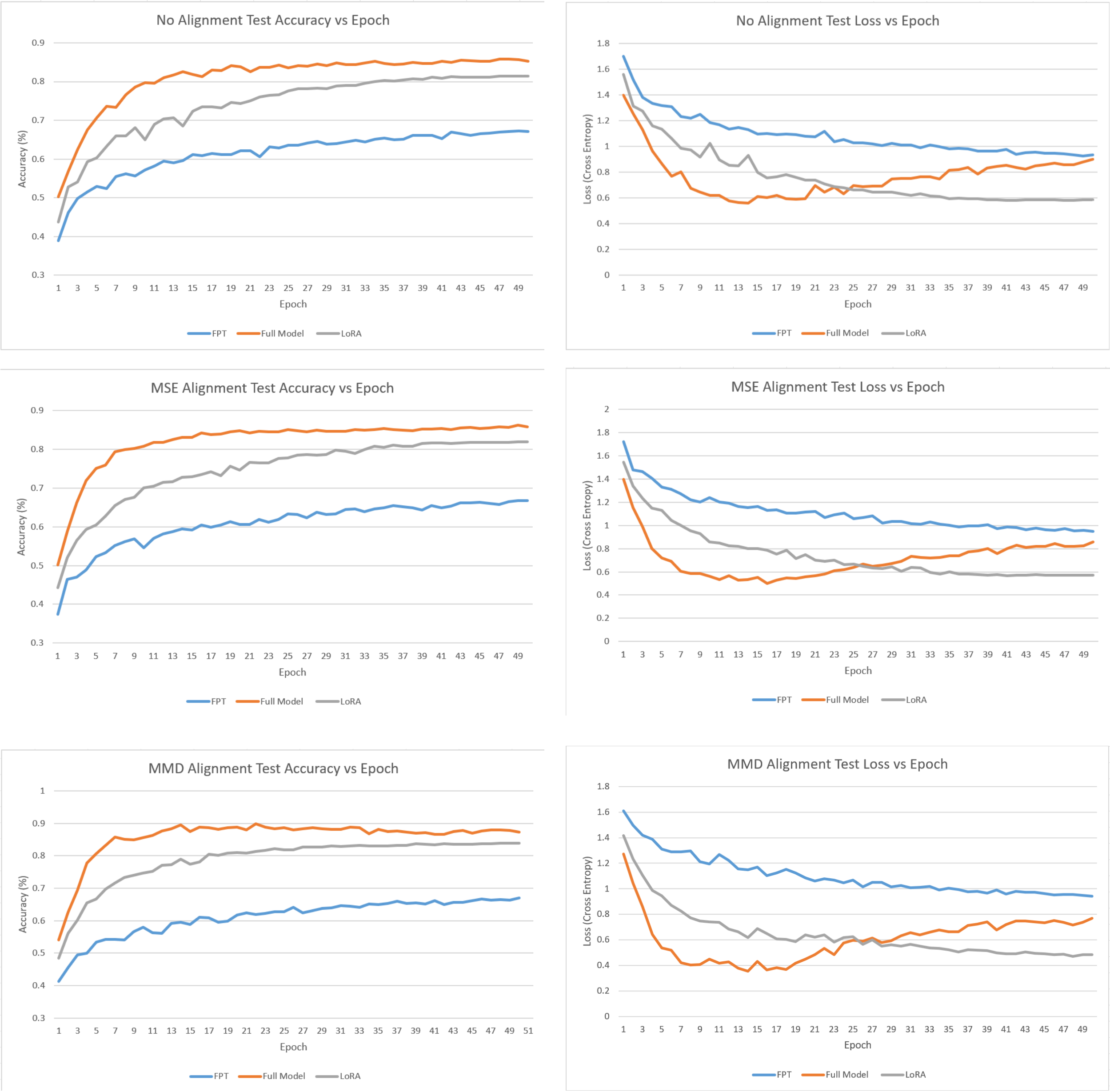

Figure 2: Test Accuracy and Loss Under Three Alignments Across Methods

3.2.5 Computational Efficiency

Table 2 compares the computational requirements of each method. FPT with alignment provides the best accuracy-to-parameter ratio, requiring only 0.1% of parameters to be updated while achieving 67% accuracy. This makes it an effective method in low-resource settings.

LoRA with alignment achieves only 3% lower performance than full fine-tuning performance while updating only 2.2% of parameters, representing an excellent trade-off between performance and efficiency. Full fine-tuning, while achieving the highest absolute performance, requires 2.5× more memory and 1.4× longer training time compared to FPT. An important observation is that the FPT method, while only having 0.1% of the full model's trainable parameters, does not save proportionally on training time and memory compared to full model fine-tuning. This is because despite having a small number of trainable parameters, the entire model must still be loaded in memory, while gradients must still back-propagate through the entire model to update these parameters.

Table 2: Computational efficiency comparison. Trainable parameters exclude frozen backbone weights. Training time is measured in GPU hours on a single NVIDIA A100.

Method	Trainable Params (M)	Training Time (hrs)	Memory (GB)
FPT Baseline	0.11	0.9	4.8
FPT + Alignment	0.11	1.0	5.2
LoRA (rank=16)	2.45	1.0	5.3
LoRA + Alignment	2.45	1.1	5.8
Full Fine-tuning	109.6	1.3	11.1
Full Fine-tuning + Alignment	109.6	1.3	13.8

4 Discussion

4.1 Analysis and Conclusion

Our experiments offer a concrete perspective on how pretrained language models behave when repurposed for vision via embedding-space alignment.

The basic Frozen Pretrained Transformer (FPT) approach achieves 67.06% test accuracy on CIFAR-10. Though this falls short of vision-specific models, this performance is still noteworthy, given that the backbone never sees gradients from image data. The result implies that the structural computation mechanisms (attention, feed-forward layers) learned on text can generalize beyond. In other words, the Transformer architecture encodes modality-agnostic inductive biases.

This observation motivates us to add distribution alignment (MSE or MMD) on top of FPT, seeing that the performance does not improve. This suggests that matching global feature distributions alone is not sufficient for downstream visual classification. That is, distribution-level alignment does not automatically yield the semantic structure or class-discriminative power that classification tasks need.

In contrast, with full-parameter fine-tuning, the test accuracy greatly increases to over 87%. This exciting result indicates that the model effectively "learns to see." However, this comes with a high computational cost and risks of catastrophic forgetting of original linguistic ability.

Finally, we experiment with parameter-efficient fine-tuning via LoRA. With a test accuracy over 83%, though slightly lower than full fine-tuning, it shows a noteworthy efficiency in adapting the model with manageable computation cost. This method leverages some of the pretrained backbone's generalization while gaining flexibility.

As the model is allowed to adapt its internal representations when fine-tuning (both full and PEFT), the alignment methods of MSE and MMD function well to enhance the accuracy by 0.2-2.5%. This shows that even slight alignment of the input embedding distribution to the pretrained model's native space can yield modest but consistent benefits.

Based on our findings, we conclude that LLMs (or pretrained Transformers) indeed carry a general-purpose computation backbone. Even without any vision-specific training, they possess a nontrivial capacity to process structured non-text input. However, embedding-space distribution alignment alone is too weak for high-performance vision tasks, but it does function when applied to fine-tuning methods. Full fine-tuning remains the most effective way to adapt a language-model backbone for vision applications, since it allows the model reshape internal representations to better suit the structure of image data. Besides, parameter-efficient tuning (LoRA + alignment) is a promising compromise with its cost efficiency and mild drop in classification accuracy. To summarize, our work suggests a practical path for lightweight cross-modal adaptation: align input embeddings, then selectively refine internal parameters.

4.2 Limitations and Future Work

Our work reveals several important limitations of the current method, as well as promising directions for future work.

First, our current alignment objective relies entirely on global distribution matching (e.g., via MSE or MMD). It can bring the overall feature distributions closer, but does not guarantee that embeddings from the same class cluster together or that different classes remain separable in embedding space. Future work should investigate more structured alignment objectives. Distance measures such as the Optimal Transport Dataset Distance (OTDD), which align datasets by considering both feature and label distributions thus preserving clustering structure, guides a promising direction.

Second, our evaluation is constrained to a single, relatively small, low-resolution dataset: CIFAR-10. While this dataset serves as a convenient testbed, its simplicity and limited diversity make it a poor proxy for real-world visual tasks. To better assess the scalability and generality of LLM-based cross-modal adaptation, future work should extend experiments to larger, higher-resolutiondatasets (e.g. ImageNet, COCO, or real-world image collections), where semantic complexity, inter-class variation, and fine-grained features pose bigger challenges.

Third, our current scope is limited to image classification: a relatively straightforward vision task. In more complex applications such as object detection, semantic segmentation, captioning, or multimodal retrieval and generation, the requirements on embedding structure, spatial reasoning, and semantic alignment are far greater. It remains an open question whether embedding-space alignment (with or without fine-tuning) will suffice in those contexts. Therefore, future work should explore these more demanding tasks.

Finally, there is room for improving our adaptation and architectural design. For example, the embedding network used to map image patches into LLM embedding space can become a deeper, more expressive projection network to encode richer visual structure.

Overall, this study represents a promising starting point for further research into cross-modal adaptation and alignment in the rapidly evolving multimodal field.

4.3 Future of This Field

Beyond the scope of this project, the broader field of LLM-centered multimodal learning is undergoing a clear shift toward more unified and semantically grounded representation frameworks. One promising direction is the development of modality-agnostic embedding spaces, where images, text, audio, and video share a common token vocabulary or latent geometry. This trend is already visible in recent multimodal foundation models, which increasingly rely on shared embedding layers or unified encoders rather than maintaining separate modality-specific backbones.

In parallel, semantic alignment is expected to become the dominant mechanism for cross-modal integration. Contrastive vision–language pretraining, instruction-driven multimodal supervision, and synthetic captioning pipelines all point to a future where LLMs learn visual semantics through grounded interactions rather than statistical matching alone. As multimodal datasets grow larger and more diverse, semantic alignment is poised to replace low-level embedding matching as a more robust and scalable alternative.

To summarize, multimodal models are steadily evolving toward unified, semantically grounded architectures that integrate vision and language within the same representational system.

References

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Pretrained transformers as universal computation engines. In *Advances in Neural Information Processing Systems*, 2021.

Yanshuai Hao, David Muir, Hao Wang, et al. Language models are general-purpose interfaces. In *International Conference on Learning Representations*, 2022.

Zheyan Shen et al. Cross-modal fine-tuning: Align then refine. *arXiv preprint arXiv:2302.05738*, 2023.

Zhengfan Zhu, Beiwen Sun, Wei Zhang, et al. Beyond text: Frozen large language models in visual signal comprehension. *arXiv preprint arXiv:2401.00000*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. CIFAR-10 dataset.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model, 2024.

Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. A system for massively parallel hyperparameter tuning, 2020.