Model Reprogramming Outperforms Fine-tuning on Out-of-distribution Data in Text-Image Encoders

***Abstract*—When evaluating the performance of a pre-trained model transferred to a downstream task, it is imperative to assess not only the in-distribution (ID) accuracy of the downstream model but also its capacity to generalize and identify out-of-distribution (OOD) samples. In this paper, we unveil the hidden costs associated with intrusive fine-tuning techniques. Specifically, we demonstrate that commonly used fine-tuning methods not only distort the representations necessary for generalizing to covariate-shifted OOD samples (OOD generalization) but also distort the representations necessary for detecting semantically-shifted OOD samples (OOD detection). To address these challenges, we introduce a new model reprogramming approach for fine-tuning, which we name REPROGRAMMER. REPROGRAMMER aims to improve the holistic performance of the downstream model across ID, OOD generalization, and OOD detection tasks. Our empirical evidence reveals that REPROGRAMMER is less intrusive and yields superior downstream models. Furthermore, we demonstrate that by appending an additional representation residual connection to REPROGRAMMER, we can further preserve pre-training representations, resulting in an even more safe and robust downstream model capable of excelling in many ID classification, OOD generalization, and OOD detection settings.**

***Index Terms*—Fine-tuning, Robustness, Out-of-distribution Detection, Distribution Shift**

I. INTRODUCTION

As pre-trained models become increasingly adopted for addressing complex downstream tasks, it has become progressively more important to ensure not only the in-distribution accuracy of the downstream model but also its robustness and safety when confronted with distribution shifts. In real-world applications, models often encounter samples that deviate to varying degrees from the expected in-distribution dataset. For samples exhibiting covariate shifts (non-semantic) from the in-distribution, we assess robustness by measuring the OOD generalization, where a robust model should consistently maintain high accuracy across all covariate-shifted OOD samples. Alternatively, for samples exhibiting semantic shifts from the in-distribution, we evaluate safety through OOD detection, where a safe and robust model should be capable of distinguishing semantically shifted OOD samples from the ID samples. Recently, both of these problems have been rigorously studied with a plethora of new and exciting literature aimed at addressing these issues [1]–[15].

However, several fundamental challenges still impede researchers from improving ID, OOD generalization, and OOD detection performances. These challenges range from difficulties in encapsulating covariant (domain) shifts, to overconfidence when predicting semantically shifted samples [10], [14], [16]. One framework, for training deep learning models, that has demonstrated strong performance in both ID classification and OOD generalization settings is large text-image supervised pre-trained models [13], [17], [18]. However, it has recently become apparent that common fine-tuning methods can distort the robust representations acquired during multi-modal pre-training, which can result in a decline in the fine-tuned model's OOD generalization performance [4], [15], [19]. Moreover, it also remains unclear whether these distortions, induced by fine-tuning, will adversely affect OOD detection tasks in the same way observed in OOD generalization.

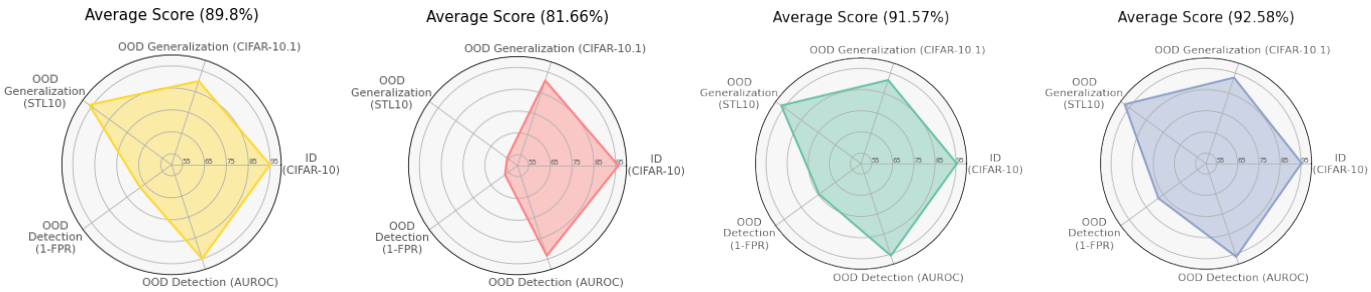
In this paper, we present evidence demonstrating that common fine-tuning techniques, such as linear-probing (optimizing only the classification head), full fine-tuning (optimizing all model parameters), LP-FT (optimizing classification head first before full fine-tuning) [4], regularized fine-tuning (optimizing all model parameters while applying regularization to the zero-shot weights) [20], model soups (average the weights of zero-shot and fine-tuned models) [15], and prompt learning (optimizing adjustable tokens in the caption) [21], can not only degrade OOD generalization performance but also compromise OOD detection capabilities. Furthermore, we illustrate that each of these common fine-tuning techniques possesses distinct strengths and hidden costs associated with their ID, OOD generalization, and OOD detection capabilities. This raises the question *can we develop an alternative fine-tuning technique that is less intrusive, more robust, and safer on both covariate and semantically shifted OOD samples?*

We tackle this question by exploring and altering an alternative approach to transfer learning called *model reprogramming* [22]. By leveraging and altering key components of *model reprogramming*, we present a new method for reprogramming a text-image pre-trained model to a downstream ID task. We also show that due to the less intrusive nature of *model reprogramming* (no adjustments to the pre-trained model parameter) our approach is less distortive, leading to improved OOD generalization and OOD detection performances. In addition, our findings also reveal that by incorporating a representation residual connection into REPROGRAMMER, we can further promote the retention of pre-training representations. Our operating hypotheses are

[H1] Traditional fine-tuning techniques can degrade both OOD generalization and OOD detection performances in CLIP-like models, leading to worse OOD performance when compared to the untuned zero-shot model.

[H2] By solely employing *model reprogramming* techniques, which are less intrusive and do not impose any changes to the pre-trained model parameters, our REPROGRAMMER will lead to more pre-training representations being maintained throughout the fine-tuning process.

[H3] The addition of a representation residual connection to the zero-shot model can further maintain pre-training



(a) Linear-probing (b) Full Fine-tuning (c) Reprogrammer (d) Residual Reprogrammer

Fig. 1: Radar charts illustrating the trade-offs between ID, OOD generalization, and OOD detection performances across linear-probing, full fine-tuning, REPROGRAMMER, and RESIDUAL REPROGRAMMER. All results are based on the CIFAR benchmarks. To quantify the cost-performance trade-offs, we report the average scores normalized across all metrics.

representations, leading to enhanced OOD generalization and detection performances.

[H4] Reprogramming the image encoder on ID samples enables REPROGRAMMER to more effectively align covariate-shifted OOD samples with the in-distribution space during inference, consequently resulting in enhanced ID classification and OOD generalization.

More formally, we introduce REPROGRAMMER and RESIDUAL REPROGRAMMER, a pair of *model reprogramming* techniques that leverage two distinct modalities of reprogramming functions to simultaneously reprogram the image encoder and the text encoder. Subsequently, we conduct a comprehensive set of evaluations demonstrating the superiority of our REPROGRAMMER and RESIDUAL REPROGRAMMER methods. To the best of our knowledge, we are the first to venture into applying *model reprogramming* techniques to multi-modal joint text-image encoder models. An illustration depicting the trade-offs between cost and performance associated with intrusive fine-tuning techniques can be found in Figure 1. Our **key results and contributions** are summarized as follows:

- We demonstrate that common fine-tuning techniques can degrade OOD performances, resulting in trade-offs between ID, OOD generalization, and OOD detection. A visual comparison of these trade-offs in terms of cost and performance is provided in Figure 1.
- We introduce REPROGRAMMER and RESIDUAL REPROGRAMMER, a pair of simple yet effective fine-tuning techniques designed to fully maintain and harness pre-training representations in CLIP-like models.
- Our results show that RESIDUAL REPROGRAMMER consistently outperforms all other methods holistically when evaluating ID, OOD generalization, and OOD detection tasks. Improving the aggregated performance by $+2.78\%$ on CIFAR benchmarks and $+0.69\%$ on `ImageNet-1k` benchmarks when compared to the next best method.
- Additionally, we conduct supporting ablations to improve our understanding of REPROGRAMMER under (1) varying degrees of reprogramming strength and (2) visualizing the reprogrammed feature space under covariate shifts.

II. Background and Related Work

Pre-trained and CLIP-like Models: Pre-trained models, trained on vast and diverse datasets, have become a popular technique for constructing robust machine learning models capable of efficient transfer to downstream tasks [23]–[32]. In this paper, we primarily focus on the Contrastive Language-Image Pre-training (CLIP) model [13]. CLIP is a multi-modal model pre-trained on a large dataset of 400 million image-caption pairs collected from the web. More specifically, given a set of image-caption pairs $D = \{(X_1, T_1)..., (X_n, T_n)\}$, CLIP-like models train an image-encoder f and a text-encoder g such that the cosine similarity between the features $f(x_k)$ and $h(t_k)$ are maximized with respect to each pair k.

Out-of-distribution Generalization: To assess the OOD generalization performance of our downstream models, we fine-tune and compare the accuracy of our tuned models using two distinct yet interconnected datasets D_{in} and D_{out}. The dataset D_{in} corresponds to the in-distribution dataset to which our pre-trained model is tuned on. The OOD dataset D_{out} represents a covariate (domain) shifted out-of-distribution dataset, comprising samples that share the same semantic classifications as those in the in-distribution dataset D_{in} but manifested under different domains. These domains can include sketches, origami, and other variations of the in-distribution classes [33]–[36].

In an OOD generalization context, the goal of an effective fine-tuning technique is to attain high accuracy across both D_{in} and D_{out}. Being able to achieve high accuracy across both datasets is paramount, as an intelligent and robust model should be agnostic to the covariate shifts of a given sample.

Out-of-distribution Detection: Out-of-distribution detection can be formulated as a binary classification problem where, given some classifier $\tilde{f}$ tasked on the in-distribution dataset D_{in}, our objective is to design a function estimator

$$h(\hat{x}) = \begin{cases} \text{in}, & \text{if } S(\hat{x}) \geq \gamma \\ \text{out}, & \text{if } S(\hat{x}) < \gamma, \end{cases}$$

such that $h(\hat{x})$ can determine whether a sample $\hat{x}$ is in-distribution D_{in} or out-of-distribution Q_{out}.

Critically, in the OOD detection setting, our goal is to detect semantically shifted samples. For instance, if the in-

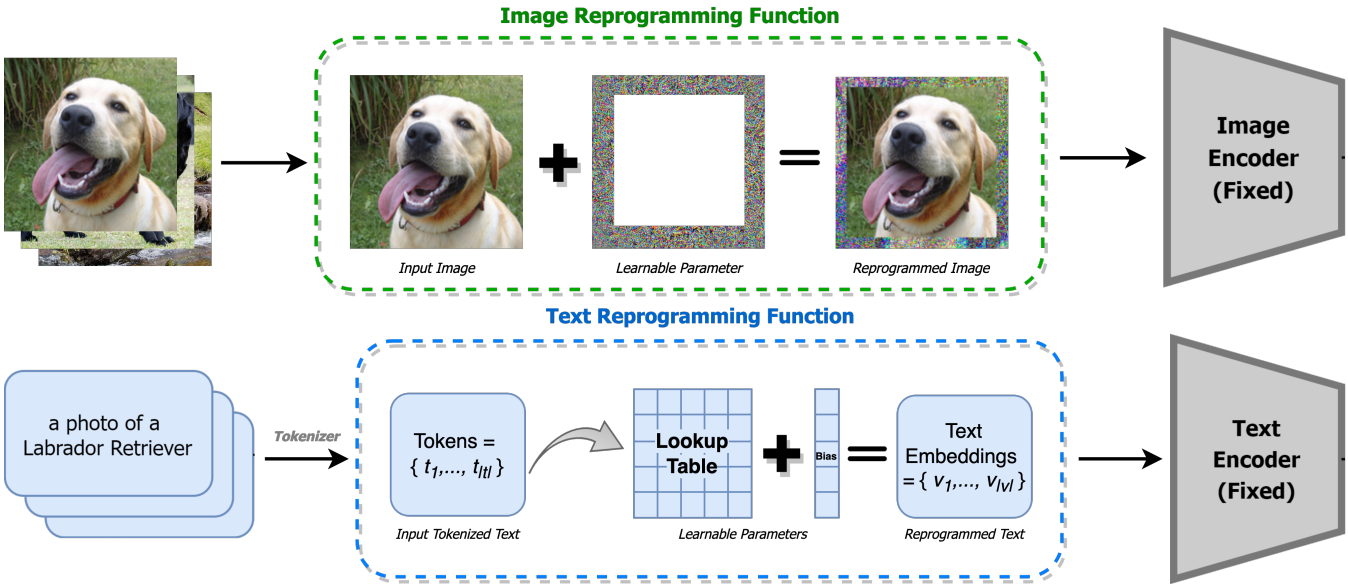


Fig. 2: Visual diagrams illustrating the image reprogramming and text reprogramming functions. In the image reprogramming function, an input image undergoes resizing and padding, followed by the addition of a learnable edge perturbation. Similarly, in the text reprogramming function, an input caption is tokenized before a lookup table and bias embedding are applied. Subsequently, both the reprogrammed image and caption embeddings are passed through the fixed image-text encoder during a model forward pass.

distribution encapsulates samples of $\{$"*cats*", "*dogs*"$\}$ then the goal of our detector h, given a "*car*" sample $\hat{x}$, is to detect that the $\hat{x}$ sample does not belong to the in-distribution set $\hat{x} \notin D_{in}$, or equivalently that the sample is out-of-distribution $\hat{x} \in Q_{out}$. To evaluate OOD detection, we employ the commonly used maximum softmax probability (`msp`) detector h_{msp} [37], which measures the confidence of our classifier $\tilde{f}$ towards a given input $\hat{x}$. The goal, of a strong fine-tuning method, is to produce a downstream model $\tilde{f}$ that is more uncertainty aware. Specifically, we want the downstream model $\tilde{f}$ to not confidently classify on semantically shifted OOD samples, whilst maintaining confidence when predicting ID samples. This goal is again immediately apparent, as we want a safe and robust model to not (overconfidently) find a semantically dissociated OOD sample to be indistinguishable from ID samples [2], [7], [9], [37].

Model Reprogramming: *Model reprogramming* is a resource-efficient, cross-domain, framework used to re-purpose models for different task-specific scenarios [22]. The framework draws significant inspiration from adversarial reprogramming, which was first introduced by Elsayed et al [38]. The aim of *model reprogramming* is to re-use and re-align the data representation, from an existing model, for a separate task without fundamental changes to the model's parameters [39]. In particular, *model reprogramming* utilizes a trainable input transformation (reprogramming function) that maps the input to a new form for the model to ingest. Following a forward pass, *model reprogramming* employs a label mapping function to generate final classification predictions. It is important to note that reprogramming functions are not specific to any singular input; instead, the reprogramming function is consistently applied to all inputs. Traditionally, *model reprogramming* methods operate by training an image/audio reprogramming function to optimally transform continuous input data, such that the output of the model can be used to perform some other desired task [38], [39]. *Model reprogramming* methods have also been proven to be successful in both white-box and black-box settings [40]. Additionally, Neekhara et al [41] presented a reprogramming method for sequence classification models, by utilizing a context-based vocabulary remapping function [41], [42]. To the best of our knowledge, this paper is the first *model reprogramming* method tackling joint text-image encoders in a multi-modal setting.

III. Methodology

In this section, we begin by introducing the image and text reprogramming modules as presented in Figure 2, followed by the full REPROGRAMMER and RESIDUAL REPROGRAMMER fine-tuning techniques. Additionally, we offer further details on our methodology in Appendix B.

A. Image Reprogramming

Consider just the CLIP image encoder $f : I \rightarrow \mathbb{R}^{b \times k}$ where b is the input image batch size and $k = 512$ is the CLIP feature size. To apply reprogramming, we leverage the commonly used adversarial program first described by Elsayed et al [38], which we define as the reprogramming function ψ. The reprogramming function ψ is applied to the input image pre-forward pass through the CLIP image encoder f. Critically, the reprogramming function ψ is not specific to any

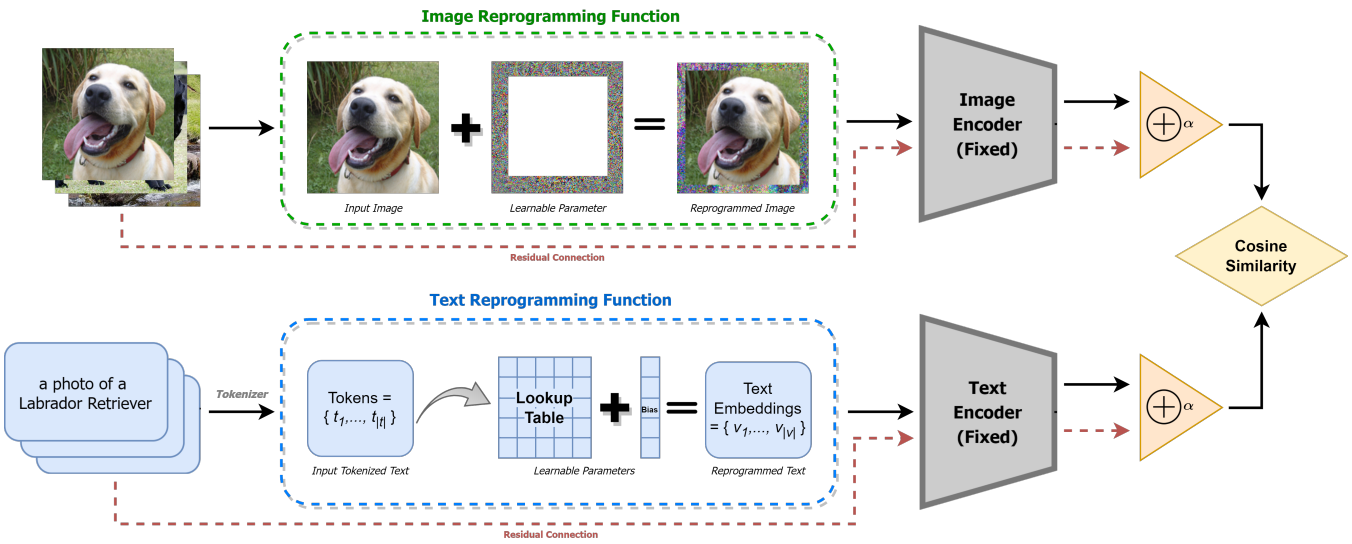


Fig. 3: Visual diagram illustrating the REPROGRAMMER and RESIDUAL REPROGRAMMER training schema based on the CLIP joint image and text encoder setting. During REPROGRAMMER training, an image and caption pair each independently undergoes their respective reprogramming functions before being passed into the CLIP image and text encoders. A loss is then computed based on the cosine similarity of the two reprogrammed features. Then we subsequently backpropagate and optimize each parameter associated with the image and text reprogramming function. During inference time, RESIDUAL REPROGRAMMER leverages a residual connection that combines the reprogrammed representation and zero-shot representations.

singular input image, rather ψ will be consistently applied to all images. We define our reprogramming function ψ as

$$\psi(X) = \mathcal{U}(X) + \tanh(W \odot M) \tag{1}$$

where $\mathcal{U}$ denotes an image up-sampling then zero-padding function, $W \in \mathbb{R}^{d \times d \times 3}$ is the image reprogrammer parameters that is to be learned, d is the size of CLIP's input width and height, $\odot$ denotes the Hadamard product, and M is a binary masking matrix. We define the binary masking matrix M as 0 for positions where we wish to implant the original image, and 1 for positions that we choose to reprogram.

B. Text Reprogramming

Now we consider the CLIP text encoder $g : S \rightarrow \mathbb{R}^{b \times k}$ where b is the input text batch size and $k = 512$ is the CLIP feature size. Additionally, we define our text input s as a sequence of tokens $s = \{s_1, ..., s_{|s|}\}$ where s_i is the vocabulary index of the ith token in the vocabulary list V_S. To apply reprogramming to a text input, we leverage and alter a version of the adversarial program as first described by Neekhara et al [41].

Formally, we define our text reprogramming function as $\Phi_{\theta,b}$ where $\Phi_{\theta,b}$ is a simple look-up embedding and bias on the tokens $\{s_i\}$ that can be parameterized by the learnable embedding tensor θ and the bias parameter b. Specifically, we define our $\theta \in \mathbb{R}^{|V_S| \times d}$ and $b \in \mathbb{R}^d$ where our default vocabulary size is $|V_S| = 49408$, which is the expected vocabulary size for the CLIP text encoder. Similarly, as with all reprogramming functions, the text reprogramming function is not specific to any singular text input, rather $\Phi_{\theta,b}$ will be consistently applied to all text inputs.

An example of the text reprogramming function goes as follows. First, we set $s_i =$ "a photo of a $\{c_i\}$" where c_i is the given sample class label. Our text reprogramming then tokenizes the string $s =$ "a photo of a Labrador Retriever" into tokens t_s. Subsequently, the tokens t_s are passed into the Φ_θ function to embed the tokens into a matrix $v'_s \in \mathbb{R}^{|t_s| \times e}$ where each token in t_s becomes an embedding vector of size e. Then we apply a bias parameter b to v'_s in the form of $v_s = v'_s + b$, before finally passing the vector v_s through the CLIP text encoder g to get the reprogrammed text features.

C. Reprogrammer

Finally, to train our given image and text reprogramming functions ψ and $\Phi_{\theta,b}$, we define our training objective as

$$W^*, \theta^*, b^* = \underset{W,\theta,b}{\operatorname{argmax}} \left(\operatorname{sim}(f(\psi_W(x)),\ g(\Phi_{\theta,b}(s)))\right) \tag{2}$$

where (x, s) is an image and caption pair obtained from our training set D_{in}, f and g are the CLIP image and text encoders respectively, sim is the cosine-similarity function, and W, θ, b are the learnable parameters encapsulating our reprogramming functions $\psi_W, \Phi_{\theta,b}$. In practice, rather than directly optimizing for cosine similarity, we follow the optimization schema of a symmetric cross-entropy loss as implemented in CLIP pre-training [13]. It is important to note that throughout the REPROGRAMMER training process, we impose no adjustments to any pre-trained model parameters. Thereby fundamentally limiting any distortion to the pre-training representations.

After tuning our REPROGRAMMER parameters W, θ, b we perform classification, on an input image $\hat{x}$ with m class

	Method	CIFAR Benchmark Aggregate (↑)	ImageNet Benchmark Aggregate (↑)
No Tuning	ZS	88.22	51.54
Fine-tuned	LP	89.80	55.13
	FFT	81.66	51.88
	LP-FT	89.85	55.06
	L2-SP	87.49	53.39
	WiSE-FT	89.58	55.75
	CoOp	83.13	54.35
	RP	91.44	55.11
	RRP	**92.69**	**56.64**

TABLE I: **Aggregate Results** of the fine-tuned downstream model's performance across ID classification, OOD generalization, and OOD detection tasks. To quantify the holistic performance, we report the average score normalized across all benchmarks as described in the Experimental Setup in Section IV-A. All values are percentages and **bold** values are the superior results.

D_{in}	Method	CIFAR-10 Accuracy (↑)	CIFAR10.1 Accuracy (↑)	STL10 Accuracy (↑)
No Tuning	ZS	89.23	83.30	97.40
CIFAR-10	LP	94.89	90.05	96.34
	FFT	96.24	91.05	55.90
	LP-FT	96.38	91.53	95.93
	L2-SP	95.46	90.71	87.59
	WiSE-FT	**97.63**	92.65	91.27
	CoOp	94.50	90.45	68.94
	RP	$95.23^{\pm 0.1}$	$91.42^{\pm 0.1}$	$96.58^{\pm 0.3}$
	RRP	$95.56^{\pm 0.1}$	$\mathbf{92.67}^{\pm 0.1}$	$\mathbf{97.86}^{\pm 0.1}$

TABLE II: **CIFAR Generalization Results** OOD generalization performance comparison between zero-shot (ZS), linear-probing (LP), full fine-tuning (FFT), LP-FT, L2-SP, WiSE-FT, CoOp, REPROGRAMMER (RP), and RESIDUAL REPROGRAMMER (RRP) methods with `CIFAR-10` as the in-distribution dataset. RP and RRP results are averaged over 3 random seeds. Values are percentages and **bold** values are the superior results.

labels $C = \{c_1, ..., c_m\}$, similar to that of zero-shot CLIP. Specifically, we make a prediction y through

$$y = \underset{i}{\operatorname{argmax}}(\operatorname{sim}(f(\psi_{W^*}(\hat{x})), g(\Phi_{\theta^*,b^*}(s_i)) \tag{3}$$

where s_i is the class-wise captions such that s_i = "a photo of a $\{c_i\}$" and ψ_{W^*} and Φ_{θ^*,b^*} are our learned reprogramming functions parameterized by W^*, θ^*, and b^*.

D. Residual Reprogrammer

To further retain pre-training representations, we propose RESIDUAL REPROGRAMMER which seeks to fuse prior pre-training representations alongside our reprogrammed representations. These representation residual connections have been utilized in different formulations and settings [43], [44] however they have never been applied in the context of *model reprogramming*. Additionally, it is important to note that the representation residual connections employed in RESIDUAL REPROGRAMMER are not the residual connections defined in the ResNet architecture [45]. Furthermore, we provide some intuition in Appendix C showcasing how RESIDUAL REPROGRAMMER can be interpreted as an inference time regularizer for our image and text reprogramming functions.

Consider a tuned REPROGRAMMER model with parameters W, θ, b and an input image $\hat{x}$ with m class labels $\{c_1, ..., c_m\}$. We define our residual reprogramming functions as

$$F(\hat{x}) = (1-\alpha) f(\psi_{W^*}(\hat{x})) + \alpha f(\hat{x}) \tag{4}$$

$$G(s_i) = (1-\alpha) g(\Phi_{\theta^*,b^*}(s_i)) + \alpha g(s_i) \tag{5}$$

where ψ_{W^*} and Φ_{θ^*,b^*} are our learned reprogramming functions parameterized by W^*, θ^*, and b^*. Subsequently, during inference time, we perform classification with RESIDUAL REPROGRAMMER through

$$y = \underset{i}{\operatorname{argmax}}(\operatorname{sim}(F(\hat{x}), G(s_i)) \tag{6}$$

where s_i is the class-wise captions such that s_i = "a photo of a $\{c_i\}$".

IV. EXPERIMENTS

In this section, we first outline our experimental setup for OOD generalization and OOD detection in Section IV-A, before evaluating our REPROGRAMMER and RESIDUAL REPROGRAMMER methods against other common fine-tuning techniques in Section IV-B. Furthermore, we conduct additional ablations in Section IV-C and provide supplementary experiments which can be found in Appendix E.

A. Experimental Setup

In-distribution dataset: We tune our model with `CIFAR-10` [46] and `ImageNet-1k` [47] as the in-distribution (ID) datasets. These datasets are widely employed as ID datasets for OOD generalization and OOD detection experiments. The `CIFAR-10` dataset contains labeled (32×32) resolution images covering a range of real-world objects such as horses, cats, and airplanes. The ImagetNet-1k dataset contains over 1.2 million training images spanning 1000 different real-world objects such as species of dogs and automotive vehicles.

Out-of-distribution Generalization: For models fine-tuned on `CIFAR-10`, we evaluate the OOD generalization performance on two standard covariate-shifted OOD datasets. Specifically, we evaluate generalization accuracy with the `CIFAR-10.1` [48] and `STL10` [49] datasets. For models fine-tuned with `ImageNet-1k`, we evaluate the OOD generalization performance across four widely used benchmarks. In particular, we evaluate generalization accuracy with `ImageNetV2` [35], `ImageNet-R` [33], `ImageNet-A` [34], and `ImageNet-Sketch` [36]. All of these datasets contain images derived from the same semantic labels as those in the ID dataset. For example, these datasets may encompass a sketched version of a Labrador Retriever, a cartoon depiction of a strawberry, or a photograph of a toy duck (for more details refer to Appendix A-A).

Out-of-distribution Detection: For models fine-tuned on `CIFAR-10`, we evaluate using the `msp` detector against

D_{in}	Method	ImageNet-1k	ImageNetV2	ImageNet-A	ImageNet-R	ImageNet-S
		Accuracy (↑)	Accuracy (↑)	Accuracy (↑)	Accuracy (↑)	Accuracy (↑)
No Tuning	Zero-shot (ZS)	59.44	52.79	11.82	43.48	38.61
ImageNet	Linear-probing (LP)	72.43	61.35	10.71	41.58	38.19
	Full Fine-tuning (FFT)	73.14	60.98	6.41	32.71	32.83
	LP-FT [4]	73.30	**62.04**	11.38	43.30	39.10
	L2-SP [20]	72.79	61.13	9.21	37.24	35.29
	WiSE-FT [15]	**73.86**	61.50	11.27	42.18	37.92
	CoOp [21]	70.64	58.12	**13.89**	43.26	39.28
	Reprogrammer (RP)	$72.02^{\pm 0.1}$	$61.15^{\pm 0.2}$	$12.61^{\pm 0.3}$	$44.18^{\pm 0.2}$	$39.64^{\pm 0.3}$
	Residual Reprogrammer (RRP)	$72.63^{\pm 0.1}$	$61.74^{\pm 0.1}$	$13.06^{\pm 0.3}$	$\mathbf{45.38}^{\pm 0.2}$	$\mathbf{40.12}^{\pm 0.2}$

TABLE III: **ImageNet Generalization Results.** OOD generalization performance comparison between zero-shot, linear-probing, full fine-tuning, L2-SP, WiSE-FT, CoOp, REPROGRAMMER, and RESIDUAL REPROGRAMMER methods. All methods utilize the CLIP B/32 architecture fine-tuned on `ImageNet-1k` as the in-distribution dataset. An additional description of the four covariate-shifted OOD datasets is provided in Appendix A-A. ↑ indicates larger values are better, while ↓ indicates smaller values are better. All values are percentages and **bold** values are the superior results.

four commonly used CIFAR OOD detection benchmarks. More specifically, we evaluate on the `iSUN` [50], `LSUN Resized` [51], `Places365` [52], and `Textures` [53] datasets. These OOD datasets span a wide range of objects including fine-grained images, scene images, and textural images. Importantly, these datasets are carefully chosen so that there is no semantic overlapping with respect to the `CIFAR-10` dataset. For models tuned with `ImageNet-1k`, we use the large-scale ImageNet OOD detection benchmark proposed by Huang et al [3]. Specifically, we evaluate on four OOD datasets which are subsets from the `iNaturalist` [54], `SUN` [55], `Places` [52], and `Textures` [53] datasets. These datasets are again carefully curated so that there is no semantic overlap with respect to the `ImageNet-1k` dataset (for more details refer to Appendix A-A).

Evaluation Metrics: In the context of OOD generalization, we measure all methods across the designated covariate-shifted OOD datasets using accuracy as the evaluation metric. For OOD detection, we measure all methods across each semantically shifted dataset using the false positive rate, when the true positive rate of ID samples is 95% (FPR95), and the area under the receiver operating characteristic curve (AUROC) as evaluation metrics.

Compared Methods: We compared our REPROGRAMMER (RP) and RESIDUAL REPROGRAMMER (RRP) fine-tuned models against zero-shot (ZS), linear-probed (LP), full fine-tuned (FFT), LP-FT [4], L2-SP [20], WiSE-FT [15], and CoOp [21] fine-tuned models. Each of these fine-tuning methods is commonly used in CLIP-based OOD evaluations. Zero-shot refers to applying the CLIP pre-trained model directly to the designated downstream task without making any alterations to the CLIP model. Linear probing involves optimizing a linear regression classifier directly on the frozen features extracted from the penultimate layer of the CLIP image encoder. To obtain a fully fine-tuned model, we fine-tuned all parameters in the image encoder and classification head to fit the in-distribution dataset. Subsequently, both L2-SP and WiSE-FT can be considered more sophisticated alternatives to full fine-tuning. Specifically, L2-SP is a regularized full fine-tuning method that applies L2 regularization to the model's parameters with respect to the zero-shot model parameters [20]. WiSE-FT is a model souping approach that combines the weights of a fully fine-tuned model with the zero-shot model [15]. Finally, CoOp is a prompt learning approach to fine-tuning where the learned parameters consist solely of class-wise captions as defined by s_i [21].

Learning Details: All presented experiments were conducted using the CLIP B/32 architecture unless otherwise specified. Additional experiments with higher capacity models can be found in Appendix E. We conducted a simple hyperparameter sweep for RESIDUAL REPROGRAMMER, varying α from $\{0.0, 0.1, \ldots, 1.0\}$, based on ID test accuracy. The final chosen value used during the evaluation was $\alpha = 0.4$. For images with a resolution higher than 128×128, we downscaled and cropped them to 128×128. We did this to accommodate CLIP's input size limitations, ensure paddable pixels, and maintain a fair comparison across all datasets, considering the information loss due to downsampling. Additional experiments with various degrees of downsampling, ranging from no downsampling to heavy downsampling, are presented in Appendix G. In all fine-tuning training processes, we initialized the model with the pre-trained CLIP B/32 model and performed a hyperparameter sweep over three learning rates using a cosine learning rate scheduler. For linear probing, we directly optimized a linear regression classifier on the frozen features extracted from the penultimate layer of the CLIP image encoder, sweeping over learning rates of $\{0.005, 0.002, 0.001\}$ for 5 epochs. For full fine-tuning, we initialized the classification head with text encoder features derived from class-wise captions, as specified by Wortsman [15]. We then conducted a sweep over learning rates of $\{0.00001, 0.00003, 0.0001\}$ for 5 epochs, optimizing all parameters in the image encoder and classification head. For LP-FT, we initialize the classification head using linear probing before full fine-tuning the model. Additionally, for WiSE-FT, we utilized $\alpha = 0.5$. For REPROGRAMMER, we randomly initialized both the image and text reprogramming functions and conducted a sweep over learning rates of $\{0.0005, 0.001, 0.005\}$ for 5 epochs. In all experiments, we set the batch size to 128 and included a warm-up period of 500 iterations. Further details regarding hyperparameter

D_{in}	Method	iSUN		LSUN Resize		Places365		Textures		Average	
		FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)
No Tuning	ZS	27.15	95.08	24.41	95.61	15.87	97.12	32.36	92.60	24.95	95.10
CIFAR-10	LP	36.74	94.57	28.38	95.75	24.65	96.73	39.67	92.93	32.36	94.99
	FFT	45.47	92.78	42.95	93.41	40.92	94.06	44.85	92.30	42.89	93.40
	LP-FT	37.24	94.69	36.80	95.23	28.77	95.85	40.43	93.06	35.81	94.71
	L2-SP	40.20	93.76	34.94	94.47	35.09	94.84	42.72	92.71	38.24	93.95
	WiSE-FT	37.93	94.22	31.23	95.10	34.73	95.06	40.36	93.15	36.06	94.38
	CoOp	35.38	94.48	30.53	95.37	57.77	86.72	44.72	93.52	42.10	92.52
	RP	$29.86^{\pm 0.7}$	$95.36^{\pm 0.5}$	$26.31^{\pm 0.6}$	$95.88^{\pm 0.4}$	$15.95^{\pm 0.5}$	$97.60^{\pm 0.3}$	$30.68^{\pm 0.8}$	$93.65^{\pm 0.5}$	$25.70^{\pm 0.7}$	$95.62^{\pm 0.4}$
	RRP	$\mathbf{24.87}^{\pm 0.6}$	$\mathbf{96.19}^{\pm 0.4}$	$\mathbf{20.52}^{\pm 0.6}$	$\mathbf{97.12}^{\pm 0.3}$	$\mathbf{15.22}^{\pm 0.5}$	$\mathbf{97.86}^{\pm 0.2}$	$\mathbf{26.37}^{\pm 0.6}$	$\mathbf{94.87}^{\pm 0.5}$	$\mathbf{21.75}^{\pm 0.6}$	$\mathbf{96.51}^{\pm 0.4}$

D_{in}	Method	iNaturalist		SUN		Places		Textures		Average	
		FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)
No Tuning	ZS	53.96	85.15	**64.89**	**81.26**	**65.76**	**79.30**	70.05	77.03	**63.67**	**80.69**
ImageNet	LP	**51.15**	88.25	78.68	74.58	76.42	75.15	70.25	**78.71**	69.12	79.17
	FFT	71.94	81.37	80.29	74.01	79.97	74.54	78.28	74.80	77.62	76.18
	LP-FT	59.28	85.51	78.74	74.56	76.51	75.01	73.88	76.67	72.10	77.94
	L2-SP	64.74	85.01	77.84	74.50	77.59	74.93	75.98	75.05	74.04	77.37
	WiSE-FT	52.21	**88.63**	77.08	74.89	75.70	75.91	71.05	77.92	69.01	79.34
	CoOp	60.51	82.91	77.63	73.88	76.12	73.71	**64.70**	78.68	69.74	77.30
	RP	$57.13^{\pm 1.1}$	$85.82^{\pm 0.7}$	$76.68^{\pm 1.5}$	$74.31^{\pm 1.0}$	$75.89^{\pm 1.8}$	$74.32^{\pm 1.1}$	$70.53^{\pm 1.6}$	$77.09^{\pm 1.0}$	$70.06^{\pm 1.5}$	$77.89^{\pm 1.0}$
	RRP	$51.46^{\pm 0.9}$	$87.82^{\pm 0.6}$	$69.95^{\pm 1.1}$	$77.29^{\pm 0.9}$	$70.93^{\pm 1.5}$	$76.58^{\pm 1.0}$	$69.10^{\pm 1.5}$	$77.48^{\pm 1.0}$	$65.36^{\pm 1.3}$	$79.79^{\pm 0.9}$

TABLE IV: **OOD Detection Results.** OOD detection performance comparison between zero-shot (ZS), linear-probing (LP), full fine-tuning (FFT), L2-SP, WiSE-FT, CoOp, REPROGRAMMER (RP), and RESIDUAL REPROGRAMMER (RRP) using the `msp` [37] detector. All methods utilize the CLIP B/32 architecture fine-tuned on `CIFAR-10` or `ImageNet-1k` as the in-distribution dataset. ↑ indicates larger values are better, while ↓ indicates smaller values are better. All values are percentages and **bold** values are the superior results.

settings can be found in the provided source code. In addition, when tuning on `CIFAR-10` and `ImageNet-1k`, we set the image up-sampling for REPROGRAMMER and RESIDUAL REPROGRAMMER to 160×160 and 224×224 pixels, respectively, with padding sizes of 64 and 32. As previously mentioned, images with resolutions higher than 128×128 were downscaled and cropped to 128×128 to accommodate CLIP's input size limitations and ensure consistent comparisons across datasets. Additional experiments featuring varying degrees of downsampling, from none to heavy downsampling, are presented in the Appendix G.

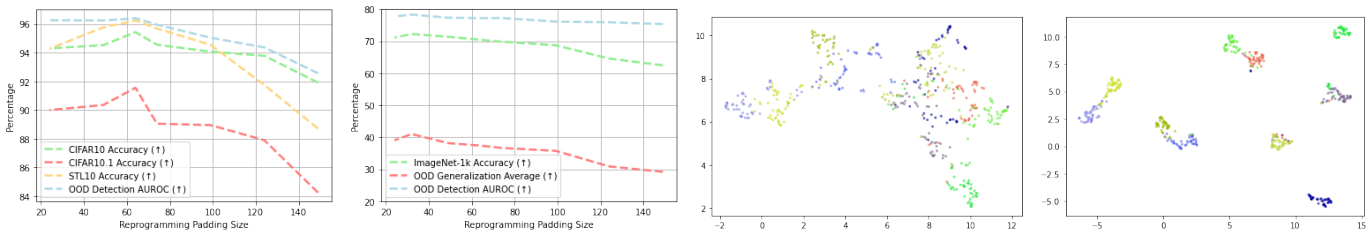
B. Results

Holistic Performance: We present a holistic evaluation in Table I, showcasing an aggregated score based on the average normalized performance across ID, OOD generalization, and OOD detection tasks for REPROGRAMMER, RESIDUAL REPROGRAMMER, and other common fine-tuning techniques. More specifically, these aggregated scores are presented in relation to the specified in-distribution dataset (`CIFAR-10` or `ImageNet-1k`) utilized for fine-tuning the pre-trained model.

We observe that the base REPROGRAMMER method generally outperforms all other compared fine-tuning methods in terms of its aggregated performance. Furthermore, it is evident that the RESIDUAL REPROGRAMMER method surpasses even REPROGRAMMER, enhancing the holistic aggregate score by $+1.01\%$ in our `CIFAR-10` benchmarks and $+1.13\%$ in our `ImageNet-1k` benchmarks. These aggregated performances provide strong support for our hypotheses that less intrusive fine-tuning techniques, such as REPROGRAMMER, will yield more holistically robust downstream models that are better equipped to handle covariate and semantically shifted OOD samples.

Out-of-distribution Generalization: We provide a detailed evaluation in Table II and Table III, focusing on the generalization accuracy of our REPROGRAMMER and RESIDUAL REPROGRAMMER methods after fine-tuning on the `CIFAR-10` and `ImageNet-1k` ID datasets, respectively. We note that WiSE-FT exceeds most other methods specifically in the ID classification task. This is in line with expectations set by prior works [15]. However, in the context of OOD generalization tasks, RESIDUAL REPROGRAMMER consistently outperforms all other fine-tuning techniques across all benchmarks. This observation substantiates our hypothesis that maintaining diverse pre-trained representations is crucial for effectively generalizing to covariate-shifted OOD samples. Furthermore, we notice that full fine-tuning in particular yields significantly worse results compared to all other fine-tuning techniques. This observation also aligns with prior works that have shown naive full fine-tuning to negatively distort the pre-training representations necessary for robust OOD generalization [4].

Out-of-distribution Detection: We provide a detailed evalu-



(a) Reprogrammer Padding Size (CIFAR Benchmarks) (b) Reprogrammer Padding Size (ImageNet Benchmarks) (c) Linear-probing Feature Space (d) Reprogrammer Feature Space

Fig. 4: **Ablation Studies.** Figures 4a, 4b illustrate the effectiveness of our REPROGRAMMER method as we adjust the image reprogramming padding size. A larger padding size indicates that more of the input image is being subjected to the reprogramming function. Additionally, we present UMAP visualization comparing the feature spaces between linear-probed and REPROGRAMMER models using 500 randomly sampled covariate shifted (`CIFAR-10.1`) images in Figures 4c, 4d.

ation of OOD detection in Table IV. Specifically, we present the OOD detection performances of our fine-tuned models across four semantically shifted OOD datasets, as well as the averaged performance across all four datasets in both the `CIFAR-10` and the `ImageNet-1k` ID settings. To ensure fair comparisons, we employ the commonly used baseline `msp` detector [37] across all experiments as a measure to assess the level of overconfidence exhibited by each downstream model when dealing with semantically shifted OOD samples. Firstly, it is very apparent that all non-reprogrammer fine-tuning techniques exhibit worse OOD detection performance in comparison to the zero-shot model. This observation reinforces our hypothesis that fine-tuning techniques have an adverse impact on the downstream model's ability to detect semantically shifted OOD samples.

Secondly, we also note that RESIDUAL REPROGRAMMER outperforms all other fine-tuning techniques in both `ImageNet-1k` and `CIFAR-10` OOD detection benchmarks. However, in the `ImageNet-1k` benchmarks, neither REPROGRAMMER nor RESIDUAL REPROGRAMMER manages to surpass the OOD detection capabilities of the zero-shot model. The superiority of the zero-shot model for OOD detection, when compared to fine-tuned models is expected as recent research has also demonstrated the effectiveness of the zero-shot CLIP model for OOD detection tasks [56]. This implies that while REPROGRAMMER and RESIDUAL REPROGRAMMER lead to improved downstream models compared to other fine-tuning techniques, there remains an inherent OOD detection cost associated with fine-tuning a pre-trained model. Subsequently, this hidden cost can further materialize as a trade-off between generalization capabilities (ID & OOD generalization) and detection capabilities (OOD detection) when fine-tuning a pre-trained model.

C. Ablation Studies

Reprogrammer Padding Size: In this ablation study, we evaluate the effectiveness of our REPROGRAMMER training as we adjust the image reprogramming padding size. The image reprogramming padding size refers to a set of hyperparameters within our image reprogramming module ψ that control the extent of border and padding perturbations applied to the image. Consequently, a larger image reprogramming padding size results in more extensive perturbations being applied through the image reprogramming module. We illustrate the effects of varying reprogramming padding sizes, increasing the permissible border pixels from 30 to 140, in Figure 4. More specifically, we present the impacts of padding size adjustments on both our `CIFAR-10` benchmarks (Figure 4a) and `ImageNet-1k` benchmarks (Figure 4b). Comparing the results of our ablation study, we observe that the optimal range of padding sizes for our `CIFAR-10` reprogrammed model tends to be larger than that for our `ImageNet-1k` reprogrammed model. We hypothesize that this discrepancy is due to the lower-resolution images present in the `CIFAR-10` dataset, which compels our REPROGRAMMER to adopt a more aggressive perturbation strategy to compensate for the lower-resolution samples.

Reprogrammed Feature Space: In this ablation study, we offer additional insights to illustrate how reprogramming can enhance the alignment of covariate-shifted OOD samples. Figures 4c and 4d present *UMAP* visualizations comparing the feature space between the linear-probed and reprogrammed models on covariate-shifted OOD samples [57]. Observing these visualizations, it becomes evident that REPROGRAMMER generates more class-conditionally distinct and compact clusters of covariate features. This observation further substantiates our hypothesis that *model reprogramming* techniques can effectively align OOD samples with the strongly tuned in-distribution space, thereby enabling REPROGRAMMER and RESIDUAL REPROGRAMMER to more accurately distinguish and classify covariate-shifted OOD samples.

V. Discussion

In this section, we discuss the limitations and computational efficiency of our REPROGRAMMER and RESIDUAL REPROGRAMMER methods. Specifically, we explore some of the inherent limitations relating to Text-Image Encoder architectures in Section V-A, discuss the challenges associated with learning under high output space dimensionality in Section V-B, discuss our work with relation to some recent advancements in Section V-C, and elaborate on the computational efficiency of our methods in Section V-D.

A. Text-Image Encoders

Our proposed REPROGRAMMER and RESIDUAL REPROGRAMMER methods aim to reprogram both the image and text encoders in CLIP-like models. Due in part to the necessity for there to be paired image and text encoders, our methods are limited to these joint text-image encoder models. However, diverging from our REPROGRAMMER and RESIDUAL REPROGRAMMER methods, we can harness components from our approach in combination with other fine-tuning techniques. Specifically, Kumar [4] demonstrated how to mitigate full fine-tuning distortions by initially tuning the classification head before proceeding with full fine-tuning of the model. Likewise, the image reprogramming function can be integrated with linear-probing or full fine-tuning to develop another set of fine-tuning techniques. These supplementary methods could potentially yield similar out-of-distribution benefits to those observed in our paper and may also be applicable to other non-CLIP pre-trained models. We leave this area open for future exploration.

B. Output Space Dimensionality

A potential limitation of REPROGRAMMER and RESIDUAL REPROGRAMMER is their effectiveness in tasks with higher output dimensionality. Specifically, *model reprogramming* techniques generally perform better on tasks with lower output dimensionality, such as the 10-way classification in `CIFAR-10`. However, when dealing with tasks with higher output dimensionality, like the 1000-way classification in `ImageNet-1k`, *model reprogramming* techniques typically require more extensive tuning and tend to be less capable of outperforming other fine-tuning techniques. This limitation is reflected in our methods as, although REPROGRAMMER and RESIDUAL REPROGRAMMER still demonstrate improvements in the showcased ImageNet benchmarks, these improvements are less significant when compared to the CIFAR benchmarks. An intuitive initial approach to addressing this issue would be to replace our traditional and simple reprogramming functions with more robust functions. However, this topic is beyond the scope of this paper, and we will leave this problem open for future *model reprogramming* research.

C. Comparison to MaPLe

Khattak et al. proposed a new prompt learning technique called MaPLe, specifically designed for multi-modal models [58]. MaPLe was developed concurrently with REPROGRAMMER and RESIDUAL REPROGRAMMER, and both methodologies address the same domain of multi-modal model fine-tuning.

However, there are significant methodological differences that distinguish REPROGRAMMER and RESIDUAL REPROGRAMMER from MaPLe. In particular, the Deep Vision Prompting in MaPLe utilizes multi-layer prompting, where each layer of the transformer undergoes an independent prompt learning module [58]. This approach differs from both traditional model reprogramming and RP, as they employ solely an input-level transformation function. Subsequently, this enables RP to be more lightweight, easier to implement in real-world applications, and versatile for settings like black-box optimization. Additionally, the Vision Language Prompt Coupling proposed in MaPLe establishes fixed prompting pairs between all layers of the Image and Text encoders [58]. This deviates from RP, where each modality is provided with its reprogramming function that is independently learned. This design proves to be particularly advantageous for diverse multi-modal models, where the paired modalities may not be image-text. For example, multi-modal models of text-audio will prove challenging for MaPLe to adapt to.

We would like to reiterate that MaPLe is specifically designed to enhance prompt tuning for In-Distribution (ID) classification and generalization settings. In contrast, our work focuses on addressing robustness through Out-of-Distribution (OOD) detection and generalization. Furthermore, our goal is to shed light on the challenging trade-off between In-Distribution (ID) and OOD performances that can arise during the fine-tuning process. Additionally, MaPLe focuses on ID classification and generalization settings, while our work specifically addresses the OOD generalization and OOD detection settings. Consequently, a substantial portion of our paper aims to shed light on the challenging trade-off between ID and OOD performance that can arise during fine-tuning, and we also provide an in-depth discussion on how to measure and address these trade-off concerns. In summary, there are significant methodological and setting differences between our work and MaPLe, making direct comparisons between the methods challenging.

D. Reprogramming Computational Efficiency

One of the key advantages of using *model reprogramming* techniques lies in their minimal resource and data requirements, as demonstrated by Tsai et al [40]. Specifically, the computational overhead introduced by the incorporation of REPROGRAMMER is minimal. The image reprogramming function can be broken down into a masking function involving matrix addition, and the text reprogramming function is a simple lookup operation with vector addition. As a result, the training process incurs negligible overhead due to the small set of parameters needed to be learned. Furthermore, the memory complexity associated with maintaining the reprogramming functions is also minimal. It only necessitates the storage of matrices proportional to the fixed padding and vocabulary sizes specified by the REPROGRAMMER methods. This efficiency in resource utilization makes REPROGRAMMER and RESIDUAL REPROGRAMMER an appealing choice for various real-world applications.

VI. Societal Impact

The goal of our project is to enhance the safety and robustness of fine-tuning techniques applied to large pre-trained machine learning models. We believe that these improvements can have a profound impact across various societal domains. Given that many modern real-world applications heavily depend on classification, addressing these safety and robustness concerns is of paramount importance, spanning from consumer

and business applications to autonomous vehicles and medical imaging. Through this endeavor, we aim to provide researchers with an additional tool to address these complex challenges. While we do not foresee any adverse consequences stemming from our work, we aspire to continue monitoring and building upon this method in the future.

VII. Conclusion

In this paper, we demonstrated that preserving pre-training representations is critical for improving the holistic capabilities (ID classification, robustness to covariate shifts, and safety under semantic shifts) of the downstream model. To this end, we introduced an alternative approach for fine-tuning text-image encoder models called REPROGRAMMER, which aims to minimize distortion to the model's pre-trained representations through *model reprogramming* techniques. Experimental results further highlight the effectiveness of both REPROGRAMMER and RESIDUAL REPROGRAMMER when compared to other common fine-tuning techniques. We hope that our study illuminates the hidden costs associated with common fine-tuning techniques and inspires future research to leverage reprogramming approaches for fine-tuning. Moreover, we hope that our study helps to underscore the importance of measuring holistic ID and OOD performances (Table I) when evaluating the effectiveness of different fine-tuning techniques.

References

[1] J. Chen, Y. Li, X. Wu, Y. Liang, and S. Jha, "Atom: Robustifying out-of-distribution detection using outlier mining," *In Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2021.

[2] Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 951–10 960.

[3] R. Huang and Y. Li, "Towards scaling out-of-distribution detection for large semantic space," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[4] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, "Fine-tuning can distort pretrained features and underperform out-of-distribution," *International Conference on Learning Representations*, 2022.

[5] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems*, 2017, pp. 6402–6413.

[6] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, 2018, pp. 7167–7177.

[7] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[8] Z. Lin, S. D. Roy, and Y. Li, "Mood: Multi-level out-of-distribution detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[9] W. Liu, X. Wang, J. Owens, and Y. Li, "Energy-based out-of-distribution detection," *Advances in Neural Information Processing Systems*, 2020.

[10] J. P. Miller, R. Taori, A. Raghunathan, S. Sagawa, P. W. Koh, V. Shankar, P. Liang, Y. Carmon, and L. Schmidt, "Accuracy on the line: on the strong correlation between out-ofdistribution and in-distribution generalization," *International Conference on Machine Learning*, 2021.

[11] S. Mohseni, M. Pitale, J. Yadawa, and Z. Wang, "Self-supervised learning for generalizable out-of-distribution detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5216–5223.

[12] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" in *International Conference on Learning Representations*, 2018.

[13] A. Radford, J. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," *International Conference on Machine Learning (ICML)*, vol. 139, pp. 8748–8763, 2021.

[14] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, "Measuring robustness to natural distribution shifts in image classification," *Advances in Neural Information Processing Systems*, 2020.

[15] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, and L. Schmidt, "Robust fine-tuning of zero-shot models," *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2109.01903

[16] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 427–436.

[17] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," *International Conference on Machine Learning*, 2021.

[18] H. Pham, Z. Dai, G. Ghiasi, K. Kawaguchi, H. Liu, A. W. Yu, J. Yu, Y.-T. Chen, M.-T. Luong, Y. Wu, M. Tan, and Q. V. Le, "Combined scaling for open-vocabulary image classification," 2021. [Online]. Available: https://arxiv.org/abs/2111.10050

[19] A. Andreassen, Y. Bahri, B. Neyshabur, and R. Roelofs, "The evolution of out-of-distribution robustness throughout fine-tuning," *arXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2106.15831

[20] X. Li, Y. Grandvalet, and F. Davoine, "Explicit inductive bias for transfer learning with convolutional networks," in *International Conference on Machine Learning*, 2018.

[21] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision (IJCV)*, 2022.

[22] P.-Y. Chen, "Model reprogramming: Resource-efficient cross-domain machine learning," *arXiv*, 2022. [Online]. Available: https://arxiv.org/abs/2202.10629

[23] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, 2020.

[24] S. Chen, Y. Hou, Y. Cui, W. Che, T. Liu, and X. Yu, "Recall and learn: Fine-tuning deep pretrained language models with less forgetting," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

[25] K. Desai and J. Johnson, "Virtex: Learning visual representations from textual annotations," *Conference on Computer Vision and Pattern Recognition*, 2021.

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *International Conference on Learning Representations*, 2021.

[27] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," *International Conference on Machine Learning*, 2021.

[28] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," *European Conference on Computer Vision*, 2020.

[29] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *arXiv*, 2019.

[30] M. B. Sariyildiz, J. Perez, and D. Larlus, "Learning visual representations with caption annotations," *European Conference on Computer Vision*, 2020.

[31] X. Zhai, X. Wang, B. Mustafa, A. Steiner, D. Keysers, A. Kolesnikov, and L. Beyer, "Lit: Zero-shot transfer with locked-image text tuning," *Conference on Computer Vision and Pattern Recognition*, 2022.

[32] Y. Zhang, H. Jiang, Y. Miura, C. D. Manning, and C. P. Langlotz, "Contrastive learning of medical visual representations from paired images and text," *arXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2010.00747

[33] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer, "The many faces of robustness: A critical analysis of out-

of-distribution generalization," *International Conference on Computer Vision*, 2021.

[34] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," *Conference on Computer Vision and Pattern Recognition*, 2021.

[35] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do imagenet classifiers generalize to imagenet?" *International Conference on Machine Learning*, 2019.

[36] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, "Learning robust global representations by penalizing local predictive power," *Advances in Neural Information Processing Systems*, pp. 10 506–10 518, 2019.

[37] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[38] G. F. Elsayed, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial reprogramming of neural networks," *International Conference on Learning Representations*, 2019.

[39] C.-H. H. Yang, Y.-Y. Tsai, and P.-Y. Chen, "Voice2series: Reprogramming acoustic models for time series classification," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 808–11 819.

[40] Y.-Y. T. Tsai, P.-Y. Chen, and T.-Y. Ho, "Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources," *International Conference on Machine Learning*, 2020.

[41] P. Neekhara, S. Hussain, S. Dubnov, and F. Koushanfar, "Adversarial reprogramming of text classification neural networks," *Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, 2019.

[42] P. Neekhara, S. Hussain, J. Du, S. Dubnov, F. Koushanfar, and J. McAuley, "Cross-modal adversarial reprogramming," *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2427—-2435, 2022.

[43] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, and Y. Qiao, "Clip-adapter: Better vision-language models with feature adapters," 2021. [Online]. Available: https://arxiv.org/abs/2110.04544

[44] R. Zhang, Z. Wei, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, and H. Li, "Tip-adapter: Training-free adaption of clip for few-shot classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[46] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *arXiv*, 2009.

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[48] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[49] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," *Fourteenth International Conference on Artificial Intelligence and Statistics*, vol. 15, pp. 215—-223, 2011.

[50] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao, "Turkergaze: Crowdsourcing saliency with webcam based eye tracking," *CoRR*, vol. abs/1504.06755, 2015. [Online]. Available: http://arxiv.org/abs/1504.06755

[51] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[52] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1452–1464, 2017.

[53] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[54] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, "The inaturalist species classification and detection dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8769–8778.

[55] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 3485–3492.

[56] Y. Ming, Z. Cai, J. Gu, Y. Sun, W. Li, and Y. Li, "Delving into out-of-distribution detection with vision-language representations," in *Advances in Neural Information Processing Systems*, 2022.

[57] L. McInnes, J. Healy, N. Saul, and L. Grossberger, "Umap: Uniform manifold approximation and projection," *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[58] M. U. khattak, H. Rasheed, M. Maaz, S. Khan, and F. S. Khan, "Maple: Multi-modal prompt learning," *Conference on Computer Vision and Pattern Recognition*, 2023.

[59] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," in *International Conference on Learning Representations*, 2018.

[60] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[61] H. Bahng, A. Jahanian, S. Sankaranarayanan, and P. Isola, "Exploring visual prompts for adapting large-scale models," *arXiv preprint arXiv:2203.17274*, 2022.

APPENDIX A
DETAILS OF EXPERIMENTS

In this Appendix, we present a detailed description of the chosen OOD generalization and OOD detection datasets in Appendix A-A and a description of our software and hardware specifications in Appendix A-B.

A. Datasets

We present a detailed list of our OOD generalization and OOD detection evaluation datasets, along with a brief description of each dataset.

CIFAR-10 OOD Generalization Benchmarks::

- `CIFAR-10.1` [48] is a collection of over 2,000 test images, sampled from TinyImages, which are designed to be a minute distributional shift from the CIFAR-10 dataset.
- `STL10` [49] is a collection of over 8,000 test images, sampled from `ImageNet-1k`, that is commonly used in domain adaptation studies. We carefully curate the STL10 dataset to evaluate with only the 9 semantically overlapping classes, choosing to omit the semantically different "monkey" class.

ImageNet-1k OOD Generalization Benchmarks::

- `ImageNetV2` [35] is a collection of 10,000 test images with approximately 10 samples per class. The dataset was sampled utilizing the same semantic labels as defined in `ImageNet-1k` and obtained independently from any previous ImageNet models.
- `ImageNet-A` [34] is a collection of 7,500 naturally adversarial and challenging images that are sampled based on 200 semantically overlapping `ImageNet-1k` classes.
- `ImageNet-R` [33] is a collection of over 30,000 test images, based on 200 semantically overlapping `ImageNet-1k` classes, that contain images of art, cartoon, graffiti, embroidery, origami, toy, sculpture, sketch, tattoo, and other rendition of the `ImageNet-1k` classes.
- `ImageNet-Sketch` [36] is a collection of over 50,000 test images based on all 1000 of the `ImageNet-1k` classes with approximately 50 images per class. Each image is a black-and-white sketch variant of the `ImageNet-1k` class.

CIFAR-10 OOD Detection Benchmarks::

- `iSUN` [50] is a collection of over 8,925 natural scene images sampled from the SUN dataset. We include the full set of iSUN images when conducting OOD detection evaluations.
- `LSUN Resized` [51] is a collection of 10,000 testing images, sampled from the LSUN dataset, spanning across 10 different scenes with images down-sampled to the size of (32×32). We include the full set of LSUN Resized images when conducting OOD detection evaluations.
- `Places365` [52] contains large-scale photographs of scenes with 365 scene categories. There are 900 images per category in the test set and we again include the full test set for OOD detection evaluations.
- `Textures` [53], or Describable Textures Dataset, is a collection of 5,640 real-world texture images under 47 categories. We include the entire set of 5640 images for OOD detection evaluations.

ImageNet-1k OOD Detection Benchmarks::

- `iNaturalist` [54] is a collection of 859,000 plant and animal images spanning over 5,000 different species. Each image is resized to have a max dimension of 800 pixels and we evaluate 10,000 images randomly sampled from 110 classes that are carefully chosen to be semantically disjoint from the `ImageNet-1k` dataset.
- `SUN` [55] is a collection of over 130,000 images of scenes spanning 397 categories. We evaluate 10,000 randomly sampled images from 50 classes that are semantically disjoint from `ImageNet-1k` classes, as `SUN` and `ImageNet-1k` have overlapping semantic concepts.
- `Places` [52] is a collection of scene images with similar semantic coverage as SUN. We use a subset of 10,000 images across 50 classes that are semantically disjoint from the `ImageNet-1k` dataset.
- `Textures` [53], or Describable Textures Dataset, is a collection of 5,640 real-world texture images under 47 categories. We again include the entire set of 5640 images for OOD detection evaluations.

B. Software and Hardware

Software: We conducted all experiments with Python 3.8.12 and PyTorch 1.11.0.

Hardware: All experiments were conducted on NVIDIA GeForce RTX 2080Ti.

APPENDIX B
DETAILS OF METHODOLOGY

In this appendix, we present additional visualizations to help explain the individual components of REPROGRAMMER in detail. We first present additional image reprogramming details and visualizations, before moving on to detailing the text reprogramming component.

A. Image Reprogramming

We present a visual diagram of our image reprogramming in the top half of Figure 2. Additionally, we note that the image reprogramming can up-sample images to the input size of the pre-trained model. However, due to restrictions in the current open-sourced pre-trained CLIP models, our image reprogramming up-sampling is limited to being $3 \times 224 \times 224$ dimensions or less. Additionally, as part of the reprogramming function ψ, the size of the up-sampling/padding function $\mathcal{U}$ and binary masking matrix M are tunable hyperparameters.

B. Text Reprogramming

Similarly, we present a visual diagram of our text reprogramming function in the bottom half of Figure 2. We generate class-wise captions following closely with the experiments presented by Radford et al [13]. Specifically, we set $s_i =$ "a photo of a $\{c_i\}$" where c_i is the given sample class label. As an example, our text reprogramming follows the procedures where, given a text Labrador Retriever label, our text reprogramming first tokenizes the string $s =$ "a photo of a Labrador Retriever" into tokens t_s. Subsequently, the tokens t_s are passed into the Φ_θ function to embed the tokens into a vector v'_s. Then we apply a bias parameter b to the given vector v'_s in the form of $v_s = v'_s + b$, before finally passing the vector v_s through the CLIP text encoder g to get the reprogrammed text features.

APPENDIX C
ANALYZING REPRESENTATION RESIDUAL CONNECTION

For ease of notation, let us consider

$$
\begin{aligned}
f(x) &= f(x) \,/\, \text{norm}(f(x)) \\
g(s_i) &= g(s_i) \,/\, \text{norm}(g(s_i)) \\
\hat{f}(x) &= f(\psi_{W^*}(x)) \,/\, \text{norm}(f(\psi_{W^*}(x))) \\
\hat{g}(s_i) &= g(\Phi_{\theta^*,b^*}(s_i)) \,/\, \text{norm}(g(\Phi_{\theta^*,b^*}(s_i)))
\end{aligned}
$$

where $\text{norm}(\cdot)$ represents the vector l_2 norm.
Therefore, we can reduce our RESIDUAL REPROGRAMMER classification to:

$$
\begin{aligned}
y &= \underset{i}{\text{argmax}} \left[\text{sim}\left((1-\alpha)\hat{f}(x) + \alpha f(x), ((1-\alpha)\hat{g}(s_i) + \alpha g(s_i) \right) \right] && (7) \\
&= \underset{i}{\text{argmax}} [((1-\alpha)\hat{f}(x) + \alpha f(x))^\top ((1-\alpha)\hat{g}(s_i) + \alpha g(s_i))] && (8) \\
&= \underset{i}{\text{argmax}} [(1-\alpha)^2 \hat{f}(x)\hat{g}(s_i) + \alpha(1-\alpha)\hat{f}(x)g(s_i) + \alpha(1-\alpha)f(x)\hat{g}(s_i) + \alpha^2 f(x)g(s_i)] && (9) \\
&= \underset{i}{\text{argmax}} [(1-\alpha)^2 \hat{f}(x)\hat{g}(s_i) + \alpha(1-\alpha)(\hat{g}(s_i) + \epsilon)g(s_i) + \alpha(1-\alpha)f(x)(\hat{f}(x) - \epsilon) + \alpha^2 f(x)g(s_i)] && (10) \\
&= \underset{i}{\text{argmax}} [(1-\alpha)^2 \hat{f}(x)\hat{g}(s_i) + \alpha^2 f(x)g(s_i) + \alpha(1-\alpha)\hat{g}(s_i)g(s_i) \\
&\qquad + \alpha(1-\alpha)f(x)\hat{f}(x) + \alpha(1-\alpha)(\epsilon g(s_i) - \epsilon f(x))] && (11) \\
&= \underset{i}{\text{argmax}} [(1-\alpha)^2 \hat{f}(x)\hat{g}(s_i) + \alpha^2 f(x)g(s_i) + \alpha(1-\alpha)\hat{g}(s_i)g(s_i) \\
&\qquad + \alpha(1-\alpha)f(x)\hat{f}(x) + \alpha(1-\alpha)(\hat{f}(x) - \hat{g}(s_i))(g(s_i) - f(x))] && (12)
\end{aligned}
$$

where (10) holds given $\epsilon = \hat{f}(x) - \hat{g}(s_i)$.

Subsequently, we can now see that our RESIDUAL REPROGRAMMER is simply a complex combination of the REPROGRAMMER classification $(1-\alpha)^2 \hat{f}(x)\hat{g}(s_i)$, zero-shot classification $\alpha^2 f(x)g(s_i)$, some regularization by the zero-shot representation $\alpha(1-\alpha)\hat{g}(s_i)g(s_i)$, and some additional closeness regularization $\alpha(1-\alpha)(\hat{f}(x) - \hat{g}(s_i))(g(s_i) - f(x))$.

D_{in}	Method	ImageNet-1k Accuracy ↑	ImageNetV2 Accuracy ↑	ImageNet-A Accuracy ↑	ImageNet-R Accuracy ↑	ImageNet-S Accuracy ↑
No Tuning	Zero-shot (ZS)	75.26	64.13	27.98	52.42	39.80
ImageNet	Linear-probing (LP)	**83.90**	69.72	46.13	68.71	41.58
	Reprogrammer (RP)	83.42	70.36	51.62	70.20	43.80
	Residual Reprogrammer (RRP)	83.61	**70.92**	**52.45**	**72.97**	**44.28**

TABLE V: **CLIP L/14 ImageNet Generalization Results** OOD generalization performance comparison between zero-shot, linear-probing, REPROGRAMMER, and RESIDUAL REPROGRAMMER methods. All methods utilize the CLIP L/14 architecture fine-tuned on `ImageNet-1k` as the in-distribution dataset. The description of the four covariate shifted OOD datasets is provided in the Appendix. ↑ indicates larger values are better, while ↓ indicates smaller values are better. All values are percentages and **bold** values are the superior results.

APPENDIX D
ADDITIONAL DISCUSSION

In this appendix, we present some additional discussion regarding the use of differing architectures in Appendix D-A.

A. Differing Architectures

Within our evaluations, we leverage CLIP as the pre-trained model to which we apply our REPROGRAMMER and RESIDUAL REPROGRAMMER methods. Subsequently, a natural question arises asking how effective would our methods be when applied to other similar CLIP-like models with differing encoder architectures and training datasets such as ALIGN [17] and BASIC [18]. But, due in large part to a lack of available open-source pre-trained model parameters, we are unable to train or test with these comparable models. However, critically it is important to note that our REPROGRAMMER methodology is not inherently limited in any way to just the open-source CLIP models. Specifically, ALIGN and BASIC primarily differ from CLIP only in their scale, as both ALIGN and BASIC can be interpreted as CLIP but with larger capacity transformer architectures alongside a larger training dataset. Therefore, as both ALIGN and BASIC are fundamentally similar to CLIP, we hypothesize that REPROGRAMMER and RESIDUAL REPROGRAMMER should show similar effectiveness when applied to either ALIGN or BASIC. Subsequently, we leave this question open for future exploration and encourage researchers, with more readily available resources, to experiment with our proposed methodologies.

APPENDIX E
HIGHER CAPACITY CLIP EXPERIMENTS

In this appendix, we present additional experimental results showcasing the performance of REPROGRAMMER and RESIDUAL REPROGRAMMER when using higher capacity CLIP models. Specifically, we present OOD generalization performances and OOD detection performances with the larger CLIP L/14 model.

A. OOD Generalization

We present the ImageNet OOD generalization results in Table V and the CIFAR OOD generalization results in Table VI using the large pre-trained CLIP L/14 model. We choose to omit full-fine-tuning experiments due to limited limited computational resources. We observe that similar to our experimental observations with the B/32 CLIP model, REPROGRAMMER and RESIDUAL REPROGRAMMER consistently outperform both linear-probing and zero-shot on all of our OOD generalization benchmarks.

D_{in}	Method	CIFAR-10 Accuracy (↑)	CIFAR10.1 Accuracy (↑)	STL10 Accuracy (↑)
No Tuning	ZS	96.18	82.67	99.53
CIFAR-10	LP	98.04	94.63	86.29
	RP	98.32	95.72	98.79
	RRP	**98.64**	**96.16**	**99.86**

TABLE VI: **CLIP L/14 CIFAR Generalization Results** OOD generalization performance comparison between zero-shot (ZS), linear-probing (LP), and REPROGRAMMER (RP) methods utilizing CLIP ViT-L/14 tuned with `CIFAR-10` as the in-distribution dataset. ↑ indicates larger values are better, while ↓ indicates smaller values are better. All values are percentages and **bold** values are the superior results.

D_{in}	Method	iSUN		LSUN Resize		Places365		Textures		Average	
		FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
No Tuning	ZS	11.65	97.51	17.07	96.39	10.92	97.60	27.58	93.09	16.81	96.15
CIFAR-10	LP	19.58	96.47	25.96	96.08	15.94	97.43	30.13	93.12	22.90	96.04
	RP	10.31	97.20	15.84	96.53	12.08	97.63	20.51	94.75	14.68	96.53
	RRP	**7.10**	**98.24**	**11.90**	**97.95**	**10.57**	**98.01**	**16.81**	**97.06**	**11.60**	**97.81**

D_{in}	Method	iNaturalist		SUN		Places		Textures		Average	
		FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
No Tuning	ZS	**30.07**	**95.82**	**41.37**	**94.06**	**42.96**	**93.96**	**42.13**	**92.59**	**36.63**	**94.11**
ImageNet	LP	42.58	93.72	51.49	88.12	56.98	88.73	58.95	87.89	52.50	89.62
	RP	45.79	93.31	49.37	90.03	54.93	88.90	59.36	87.73	52.36	89.99
	RRP	40.46	94.18	42.81	91.55	50.62	90.75	50.20	90.06	46.02	91.64

TABLE VII: **CLIP L/14 OOD Detection Results** OOD detection performance comparison between zero-shot (ZS), linear-probing (LP), REPROGRAMMER (RP), and RESIDUAL REPROGRAMMER (RRP) methods using the `msp` [37] detector. All methods utilize the CLIP L/14 architecture fine-tuned on `CIFAR-10` or `ImageNet-1k` as the in-distribution dataset. ↑ indicates larger values are better, while ↓ indicates smaller values are better. All values are percentages and **bold** values are the superior results.

B. OOD Detection

We present our OOD detection in Table VII using the large pre-trained CLIP L/14 model. Specifically, in the top half of Table VII, we report the OOD detection performance with the CIFAR benchmarks, and in the bottom half of Table VII we report the OOD detection performance with the ImageNet benchmarks. Due to limited computational resources, we have chosen to omit full fine-tuning results. Again, for a fair comparison, we use the same commonly used baseline `msp` detector across all experiments as a way to gauge the level of overconfidence the zero-shot, linear-probed, REPROGRAMMER, and RESIDUAL REPROGRAMMER models has on semantically shifted OOD samples.

We can see that similar to the CLIP B/32 experiments, our RESIDUAL REPROGRAMMER outperforms all other fine-tuned models. However, again following observations with the CLIP B/32 experiments, we see that none of the fine-tuned downstream models were able to exceed the OOD detection capabilities of the zero-shot model. This again reaffirms the hypothesis that there is a hidden cost associated with fine-tuning a pre-trained model. In particular, this hidden cost seems to be most prominent when observing the capabilities of the downstream model on OOD detection tasks.

APPENDIX F
COMPARISON WITH OUTLIER EXPOSURE

In this section, we provide a brief comparison between REPROGRAMMER and RESIDUAL REPROGRAMMER with full fine-tuned outlier exposure [59]. We trained OE using the same full fine-tuning training setting as specified in Section IV-A alongside TinyImages [60] as the auxiliary OOD dataset and a $\beta = 0.5$ as specified by Hendrycks et al. [59]. Observing the aggregated results in Table IX, we note that RRP still surpasses OE by +2.90%. However, we want to clarify that comparing OE with RRP is not strictly fair. OOD regularization techniques like OE involve a distinct training regime, often needing an extra auxiliary OOD dataset. This is akin to unfairly providing one method with extra novel data while withholding such data from other methods. We present these empirical results as an additional point of reference for comparing RRP with existing OOD regularization techniques and not as an argument for the strict superiority of RRP.

D_{in}	Method	iSUN		LSUN Resize		Places365		Textures		Average	
		FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)
CIFAR-10	OE	**2.88**	**99.05**	**1.73**	**99.40**	**12.32**	**97.99**	**19.60**	**95.58**	**9.13**	**98.01**
	RP	$29.86^{\pm 0.7}$	$95.36^{\pm 0.5}$	$26.31^{\pm 0.6}$	$95.88^{\pm 0.4}$	$15.95^{\pm 0.5}$	$97.60^{\pm 0.3}$	$30.68^{\pm 0.8}$	$93.65^{\pm 0.5}$	$25.70^{\pm 0.7}$	$95.62^{\pm 0.4}$
	RRP	$24.87^{\pm 0.6}$	$96.19^{\pm 0.4}$	$20.52^{\pm 0.6}$	$97.12^{\pm 0.3}$	$15.22^{\pm 0.5}$	$97.86^{\pm 0.2}$	$26.37^{\pm 0.6}$	$94.87^{\pm 0.5}$	$21.75^{\pm 0.6}$	$96.51^{\pm 0.4}$

TABLE VIII: **OOD Detection Results.** OOD detection performance comparison with outlier exposure (OE) and `CIFAR-10` as the in-distribution dataset. All values are percentages and **bold** values are the superior results.

D_{in}	Method	CIFAR-10	CIFAR10.1	STL10	Aggregate
		Accuracy (↑)	Accuracy (↑)	Accuracy (↑)	Aggregate (↑)
CIFAR-10	OE	**95.62**	90.12	68.49	89.79
	RP	$95.23^{\pm 0.1}$	$91.42^{\pm 0.1}$	$96.58^{\pm 0.3}$	91.44
	RRP	$95.56^{\pm 0.1}$	$\mathbf{92.67}^{\pm 0.1}$	$\mathbf{97.86}^{\pm 0.1}$	**92.69**

TABLE IX: **Results.** OOD generalization and aggregate performance comparison with outlier exposure (OE) using `CIFAR-10` as the in-distribution dataset. Values are percentages and **bold** values are the superior results.

APPENDIX G
DOWN-SAMPLING EXPERIMENTS

In this appendix, we present our down-sampling experiments showcasing that our method isn't limited by the down-sampling step we implemented within our experiments. Specifically, in Table X we show the OOD generalization performance of REPROGRAMMER and in Table XI we show the OOD detection performance of REPROGRAMMER as we apply different degrees of down-sampling to training and testing datasets.

Down-sampling Size	Method	ImageNet-1k	ImageNetV2	ImageNet-A	ImageNet-R	ImageNet-S
		Accuracy ↑	Accuracy ↑	Accuracy ↑	Accuracy ↑	Accuracy ↑
$\mathbf{64 \times 64}$	Zero-shot	50.632	44.22	6.12	34.547	28.556
	Linear-probing	65.322	53.61	5.627	36.19	29.354
	Full Fine-tuning	**70.33**	**58.33**	5.28	30.55	29.018
	Reprogrammer	65.814	54.71	**7.08**	**36.923**	**30.653**
$\mathbf{96 \times 96}$	Zero-shot	57.284	50.59	9.813	41.043	35.587
	Linear-probing	70.464	58.89	9.16	39.973	35.517
	Full Fine-tuning	**72.228**	**60.68**	5.60	31.9	32.249
	Reprogrammer	70.244	59.27	**11.00**	**41.777**	**36.906**
$\mathbf{128 \times 128}$	Zero-shot	59.44	52.79	11.82	43.48	38.61
	Linear-probing	72.43	61.35	10.71	41.58	38.19
	Full Fine-tuning	**73.14**	60.98	6.41	32.71	32.83
	Reprogrammer	72.10	**61.28**	**12.58**	**44.30**	**39.40**
$\mathbf{160 \times 160}$	Zero-shot	60.14	53.46	12.893	44.36	39.932
	Linear-probing	73.142	61.86	11.787	42.167	39.27
	Full Fine-tuning	**73.39**	61.59	6.227	32.877	33.331
	Reprogrammer	72.934	**61.91**	**14.07**	**44.43**	**40.41**
$\mathbf{192 \times 192}$	Zero-shot	60.796	53.84	13.8	44.86	40.294
	Linear-probing	73.486	62.03	11.907	42.093	39.425
	Full Fine-tuning	**73.764**	61.86	6.427	32.137	32.135
	Reprogrammer	73.08	**62.36**	**14.73**	**44.57**	**40.73**
$\mathbf{224 \times 224}$	Zero-shot	61.896	54.71	15.267	46.713	40.83
	Linear-probing	74.882	62.45	12.6	42.217	39.944
	Full Fine-tuning	**75.071**	62.03	6.387	32.48	33.469
	Reprogrammer	74.118	**62.65**	**15.32**	**45.09**	**40.86**

TABLE X: **Down-sampling OOD Generalization Results** OOD generalization performance comparison between differing down-sampling severity. All methods utilize the CLIP B/32 architecture and were fine-tuned on the `ImageNet-1k` dataset down-sampled to the specified resolution. Similarly, the evaluation was completed using, if available, the validation dataset down-sampled to the specified resolution. ↑ indicates larger values are better, while ↓ indicates smaller values are better. All values are percentages and **bold** numbers are superior **fine-tuning** results.

Down-sampling Size	Method	iNaturalist		SUN		Places		Textures		Average	
		FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
64×64	ZS	63.57	81.3	75.1	77.8	74.34	76.14	73.67	75.82	71.67	77.77
	LP	65.44	**84.62**	84.61	70.23	83.48	70.79	**76.35**	**76.51**	77.47	75.54
	FFT	74.82	79.44	81.29	73	80.55	**73.45**	80.96	72.95	79.4	74.71
	RP	**65.42**	83.68	**79.96**	**72.05**	**80.13**	71.86	77.78	74.72	**75.82**	**75.58**
96×96	ZS	57.5	83.7	67.13	80.53	68	78.77	71.13	76.64	65.94	79.91
	LP	**54.21**	**87.46**	80.32	73.77	77.94	**74.18**	**72.36**	**78.28**	71.21	**78.42**
	FFT	73.33	80.55	80.89	73.43	80.03	74	80.09	73.75	78.58	75.43
	RP	56.76	85.62	**77.19**	**73.86**	**76.46**	73.91	73.07	76.78	**70.87**	77.54
128×128	ZS	53.96	85.15	64.89	81.26	65.76	79.30	70.05	77.03	63.67	80.69
	LP	**51.15**	**88.25**	78.68	74.58	76.42	**75.15**	**70.25**	**78.71**	**69.12**	**79.17**
	FFT	71.94	81.37	80.29	74.01	79.97	74.54	78.28	74.80	77.62	76.18
	RP	56.85	85.97	**75.68**	**74.99**	**74.80**	74.84	70.51	77.43	69.46	78.31
160×160	ZS	52.98	85.38	63.57	81.5	64.36	79.63	69.34	77.23	62.56	80.93
	LP	**50.42**	**88.37**	77.53	74.94	75.17	75.65	68.55	**79.01**	67.92	**79.49**
	FFT	71.83	81.15	81.55	73.26	80.35	74.2	78.79	74.43	78.13	75.76
	RP	56.25	85.92	**74.55**	**75.69**	**72.37**	**76.27**	**67.98**	77.93	**67.79**	78.95
192×192	ZS	53.57	85.22	63.75	81.37	63.04	80.16	68.99	77.44	62.34	81.05
	LP	**50.28**	**88.36**	78.02	74.89	74.62	76.25	**69.04**	**78.96**	67.99	**79.62**
	FFT	71.95	81.09	81.43	73.56	81.11	74.12	79.2	74.6	78.42	75.84
	RP	55.77	85.99	**74.48**	**75.31**	**71.30**	**76.79**	69.24	77.8	**67.70**	78.97
224×224	ZS	53.75	85.58	62.89	81.65	63.82	80.13	68.19	77.67	62.16	81.26
	LP	**50.88**	**88.18**	79.14	74.92	75.9	76.02	**67.73**	**79.35**	68.41	**79.62**
	FFT	72.14	80.89	80.98	74.06	80.69	74.58	79.04	74.98	78.21	76.13
	RP	55.56	85.82	**73.89**	**76.25**	**70.32**	**77.63**	68.05	78.28	**66.95**	79.5

TABLE XI: **Down-sampling OOD Detection Results** OOD detection performance comparison between differing down-sampling severity. All methods utilize the CLIP B/32 architecture fine-tuned on the **Image-1k** dataset down-sampled to the specified resolution. Similarly, all semantically shifted OOD datasets were also down-sampled to the specified resolution. ↑ indicates larger values are better, while ↓ indicates smaller values are better. All values are percentages and **bold** numbers are the superior **fine-tuning** results.

Reviews from Prior Submissions

In the following pages, we provide reviews from our prior submission with ICCV 2023. We also provide a brief description of how those reviews were addressed at the end of these reviews.

REVIEWER #1

Summary: Describe the key ideas, experiments, and their significance (preferably in 5-7 sentences):

The paper explores the challenges in evaluating pre-trained models for downstream tasks, especially in terms of their ability to generalize to and detect out-of-distribution (OOD) samples. The authors propose a novel ”reprogrammer” approach that is less intrusive and helps in maintaining pre-training representations, resulting in better holistic capabilities. The method involves optimizing the learnable parameters of image and text reprogramming functions to maximize the cosine similarity between CLIP image and text encoders after passing through their respective reprogramming functions. The authors show that their approach outperforms other common transfer learning techniques in terms of ID classification, OOD generalization, and OOD detection performances.

[Strengths] Consider the aspects of key ideas, experimental or theoretical validation, writing quality, and data contribution (if relevant). Explain clearly why these aspects of the paper are valuable. Short, unjustified review is NOT OK.:

1) The paper proposes a novel reprogramming approach that improves performance on in-distribution and out-of-distribution tasks.
2) The paper provides extensive evaluation and comparison of different transfer learning techniques.

[Weaknesses] Consider the aspects of key ideas, experimental or theoretical validation, writing quality, and data contribution (if relevant). Explain clearly why these are weak aspects of the paper.:

One major weakness of the paper is that it does not discuss the necessity of using a multimodal approach to reach its goal. Specifically, the paper proposes a novel reprogramming approach for joint text-image encoders, but it does not explore the possibility of using separate approaches for text and image encoders. For example, the paper does not discuss the performance of visual prompt tuning (VPT) or text prompt tuning (CoOp: Learning to Prompt for Vision-Language Models), which are techniques that are applicable to a single modality. Moreover, the paper does not provide a comparison with other recent approaches such as prompt tuning, which is a technique that fine-tunes pre-trained models with natural language prompts. While the proposed method is similar to prompt tuning in some ways, the paper did not conduct any discussion on this topic. Therefore, it would be useful to explore the benefits and limitations of a multimodal approach compared to a single-modality approach, and to compare the proposed reprogramming method with other recent approaches in more detail. Besides, a CVPR’23 paper of MaPLe: Multi-modal Prompt Learning has a similar achitecture, or a complexer version of the proposed method. It could be good to also discuss the topic of prompt tuning based on the mentioned work.

[Paper rating]: (3) Borderline

[Recommendation confidence]: Somewhat confident: I do not directly work on this topic, but my expertise and experience are sufficient to evaluate this paper.

[Justification of rating] Provide detailed justification of your rating. It should involve how you weigh the strengths and weaknesses of the paper.: See weakness

[Additional comments] Minor suggestions, questions, corrections, etc. that can help the authors improve the paper, if any.: NA

[Alterations in New Draft] A brief response to help clarify all the changes made to address Reviewer #1’s concerns:

To address Reviewer #1’s concerns we have updated/added the following new additions to our paper:

1) We have added a new discussion section, specifically addressing the needs and limitations of the multi-modal setting to which we frame our paper.
2) We have also added a completely new set of CoOp evaluations to our main paper in an effort to address this reviewer’s concerns regarding the lack of prompt tuning baselines. We would also like to draw attention to recent findings that indicate the comparative disadvantage of single-modality prompting when compared to multi-modality based methods [61].
3) Finally we have also added a section discussing the dissimilarities between our work and MaPLe in an effort to further clarify any future confusion any readers may have.

REVIEWER #2

Summary: Describe the key ideas, experiments, and their significance (preferably in 5-7 sentences):

This paper studies how the robustness of pre-trained classification models changes after fine-tuning, and how to improve the behavior. The paper considers two forms of robustness: "Out-of-distribution generalization" refers to the ability of a model to accurately classify images that have been distorted in a way that was not seen during training, but that does not change the semantic content of the image. "Out-of-distribution detection" refers to the ability of the model to detect images that are semantically unrelated to the training set, e.g. an image of a car in for a model trained on animals. The paper first shows that fine-tuning, while improving performance on in-distribution data, can reduce performance on OOD generalization and detection. The paper then proposes to mitigate this by using a different adaptation method called "model reprogramming". In contrast to fine-tuning, which adjusts a model's weights, reprogramming adds learnable parameters to the *input*, e.g. in the form of extra pixels appended to an image. This extra input, which is fixed for all examples, is optimized to improve performance on the target tasks. Optionally, a skip connection can be added, such that the model can interpolate between the original input and the "reprogrammed" input. The paper evaluates the proposed "Reprogrammer" method on several OOD generalization and detection benchmarks and compares to WiSE-finetuning, a prior method that consist of simply averaging the weights of the pretrained and the fine-tuned model. Reprogrammer succeeds in improving OOD performance over fine-tuning, but at the cost of slightly reduced in-distribution performance. In contrast, WiSE-finetuning tends to improve both in-distribution and OOD performance compared to fine-tuning, although the OOD improvements are often a bit smaller than for Reprogrammer.

[Strengths] Consider the aspects of key ideas, experimental or theoretical validation, writing quality, and data contribution (if relevant). Explain clearly why these aspects of the paper are valuable. Short, unjustified review is NOT OK.:

1) The proposed method generally provides significant Improvements on the presented OOD benchmarks.
2) The "model reprogramming" through changing the input image is a little-used alternative to changing the model weights, and might open new possibilities for adapting models.

[Weaknesses] Consider the aspects of key ideas, experimental or theoretical validation, writing quality, and data contribution (if relevant). Explain clearly why these are weak aspects of the paper.:

1) Performance tradeoff. According to the presented results, the proposed method involves a tradeoff: It generally improves OOD performance, but reduces in-distribution performance, compared to full fine-tuning. This poses a difficult decision to the practitioner, who now has to weigh ID and OOD performance against each other. This is a significant disadvantage of the proposed method compared to WiSE-FT, which improves *both* ID and OOD performance, and is therefore strictly preferable to full fine-tuning. This makes WiSE-FT an easy choice, even if its OOD improvements are somewhat smaller than those of the proposed method. The present paper should make suggestions for how practitioners can address the tradeoff – how can you make a principled, quantitative decision about the relative benefit of improved OOD compared to the reduction in ID performance? Alternatively, the paper should test if the proposed method can provide benefits when applied *in addition* to WiSE-FT.
2) Weak motivation of "reprogramming": It is not obvious that changing the model input is generally less "intrusive", or in other ways better, than changing model parameters; there is probably an equivalent model parameter change for any input change. Careful tuning of the fine-tuning approach, e.g. fine-tuning only part of the model, could perhaps achieve similar results. The paper should discuss if there are specific advantages of reprogramming that are not available when changing model weights.

[Paper rating]: (4) Weak reject

[Recommendation confidence]: Very confident: I am an expert on this topic.

[Justification of rating] Provide detailed justification of your rating. It should involve how you weigh the strengths and weaknesses of the paper.: The primary weakness of the proposed method is that it burdens the practitioner with a tradeoff between ID and OOD performance that other methods, specifically WiSE-FT, do not have. This needs to be addressed before the method is sufficiently practical for publication.

[Additional comments] Minor suggestions, questions, corrections, etc. that can help the authors improve the paper, if any.: See "Weaknesses".

[Alterations in New Draft] A brief response to help clarify all the changes made to address Reviewer #2's concerns:

To address Reviewer #2's concerns we have updated/added the following new additions to our paper:

1) We have reformulated a large chunk of the introduction, and have added an additional discussion section, in an effort to clarify the underlying motivation for reprogrammer. Additionally, we have also properly cited our definition of intrusive which we based on prior works.
2) The trade-off serves as a key motivation for our paper, as we aimed to demonstrate that these aspects are not necessarily aligned with each other. Specifically, we showcased how certain methods such as WiSE-FT, may excel in ID classification while underperforming in terms of OOD Generalization and OOD Detection when compared to other fine-tuning techniques. However, determining the relative importance of each aspect is a challenging problem that depends in part on the model's specific application. To help clarify this to future readers we've added a new holistic evaluation section

which means to answer the reviewer's concerns regarding how to view these ID and OOD tradeoffs and how to better compare methods like WiSE-FT and RP. We also want to note that prioritizing ID classification is not always the optimal choice. In real-world applications, involving highly sensitive tasks, the model's safety and robustness to OOD samples are sometimes more important. For example, in a self-driving context, accurate OOD Detection becomes crucial as it can prevent catastrophic mistakes caused by failure to detect distributional shifts.

REVIEWER #3

Summary: Describe the key ideas, experiments, and their significance (preferably in 5-7 sentences):

This paper addresses the issue of lost generalizability of CLIP-like models after fine-tuning on a target dataset. They introduce a re-programming technique to remap the inputs and learn the re-programming technique using vision-language pairs. Additionally, a residual version is introduced that softly combines the original representation and the re-programmed version. Then they conduct evaluation on OOD generalization and detection on datasets like CIFAR and ImageNet.

[Strengths] Consider the aspects of key ideas, experimental or theoretical validation, writing quality, and data contribution (if relevant). Explain clearly why these aspects of the paper are valuable. Short, unjustified review is NOT OK.:

1) The paper deals with an important problem of OOD generalization and detection of foundation VLM models.
2) The method is simple and easy to implement, and as advertised, computationally efficient.
3) The experimental results cover good number of datasets.

[Weaknesses] Consider the aspects of key ideas, experimental or theoretical validation, writing quality, and data contribution (if relevant). Explain clearly why these are weak aspects of the paper.:

1) My biggest concern from the paper is that there is no insight provided as to why the method works, or why it should work. For example, what specific properties in linear-probing/fine-tuning are responsible for the drop in OOD performance - and which of these limitations does re-programing successfully address which improves performance? Without knowing exactly why re-programming works, it is difficult to assess its impact for OOD performance.
2) It would be helpful to provide some more information into what exactly is learnt after re-programming. How are the parameters helping in improving OOD performance.
3) Is the network fine-tuned or linear-probed after training with re-programming? Does this choice impact the performance?
4) Does the re-programming trick also work for vision-only models and language-only models? For example, it is well-known that self-supervised vision-only models fail to generalize after fine-tuning, does this method also work well in those settings?
5) Can the re-programming be also applied to the feature space rather than the input space?
6) More information is needed on the effect of alpha in addition to noting that a value of 0.4 works best.

[Paper rating]: (4) Weak reject

[Recommendation confidence]: Somewhat confident: I do not directly work on this topic, but my expertise and experience are sufficient to evaluate this paper.

[Justification of rating] Provide detailed justification of your rating. It should involve how you weigh the strengths and weaknesses of the paper.: While the problem considered and the approach seem reasonable, I think there is not adequate insights and explanation provided in the paper that illustrates the benefits of the proposed approach. Please answer my questions above and I am ready to update my score.

[Additional comments] Minor suggestions, questions, corrections, etc. that can help the authors improve the paper, if any.: There are some minor spelling/grammar mistakes in the introduction section.

[Alterations in New Draft] A brief response to help clarify all the changes made to address Reviewer #3's concerns:

To address Reviewer #3's concerns we have updated/added the following new additions to our paper:

1) We have updated the introduction and background to help provide literature support for how reprogramming is used as a transfer learning technique and why it functions. Additionally, we've also updated the ablation section in an effort to show a UMAP visualization to clarify how reprogramming functions with respect to learned representations.
2) The reviewer raises a concern regarding the combination of reprogrammer with other techniques, or trying to apply reprogramming to just the text or image encoder. We've added a new discussion section that specifically seeks to address this concern for future readers. We also want to note here that reprogrammer is designed to intrinsically pair the image and text encoders and leveraging just reprogramming for image or text models is non-trivial and deviates outside the scope of our current paper.