Seeing Helps Reasoning in Language Models

Yulu Gan[1,†,*] Kaiya Ivy Zhao[1,†,*] Tomaso Poggio[1,2] Phillip Isola[1]

[1] CSAIL, MIT [2] CBMM, MIT † equal contribution

Abstract

Multimodal language models can process both images and texts, yet existing studies often find that naively incorporating visual information fails to improve, and can even degrade the performance of the language model. As a result, the language model backbone is usually kept fixed in multimodal setups. Nevertheless, vision and language are the two primary ways through which humans perceive and understand the world. A large language model (LLM) trained solely on text lacks direct grounding in the physical world, suggesting that, if integrated properly, visual input should enhance rather than harm its perceptual and representational capacities. However, how to integrate vision information so that it benefits LLMs remains an open challenge. In this paper, we propose Cross-Modal Alignment Regularization (CMAR), a method designed to improve LLMs by aligning their internal representations with those of vision models during training. Specifically, in addition to the standard next-token prediction objective, we introduce an alignment objective: the language model is trained to make its internal representations consistent with those of a pretrained vision model. This is achieved using an extra paired image–text dataset, where the text is fed to the language model and the image to the vision model to get language and vision representations. We use popular alignment measures to calculate the alignment score, and the model is encouraged to maximize this score, thereby bringing the internal representations of the language and vision models closer together. Experimental results demonstrate that CMAR consistently improves language models in both pre-training and fine-tuning settings for various model families, downstream tasks, and alignment measures.

1. Introduction

Perception, as Marr argued, is not just passively recording what we see but actively rebuilding the structure of the world [26]. Through hierarchical representations, vision transforms sensory signals into a structured understanding of reality. The purpose of vision, in Marr's words, is "knowledge of the world around us, not knowledge of the image itself." This insight goes beyond visual science; it defines the core of intelligence: to build world models that reflect what exists, where it is, and how it relates to everything else.

Language models, though trained only on text, are also world-modeling systems [12, 57]. They infer latent causal and relational structures from symbolic co-occurrences, reconstructing a semantic world from language alone. Yet this world is incomplete: text is rich in labels but poor in sensory content. it encodes relations but not the perceptual substrate that grounds them. For instance, a model may learn that "apples are red" or "the sun rises in the east," yet it has never seen redness nor experienced direction; it manipulates descriptions of perception without perception itself. Text teaches the model that "ice is cold" and "objects fall," but it never allows the model to feel coldness or see falling. The model thus constructs a symbolic physics of the world, coherent yet detached from perception. As a result, language models often reason fluently yet hallucinate freely. Words can describe the world, but they cannot perceive it. Between symbol and sensation lies the missing link of intelligence: perception. If vision discovers what is present and where, reasoning seeks to uncover why and what follows. Both rely on internal representations that capture the world's structure at different levels of abstraction. This complementarity suggests a natural question:

> *Can representations learned by vision models enhance those of language models?*

To answer this question, previous methods attempt to (1) *employ models from one modality to generate training data for another*, such as language models creating vision datasets [48] or programs synthesizing images [5]. Although effective, these methods rely on an indirect and costly "data bridge," resulting in limited knowledge transfer; (2) *align representations across modalities explicitly during training* [19, 40, 64, 65], by directly transferring knowledge in embedding space or jointly training Vision-Language Models (VLMs). Although more direct and effective, these methods either require multi-stage training procedures or are

*Correspondence to Yulu Gan yulu_gan@mit.edu and Kaiya Ivy Zhao kyzhao@mit.edu.

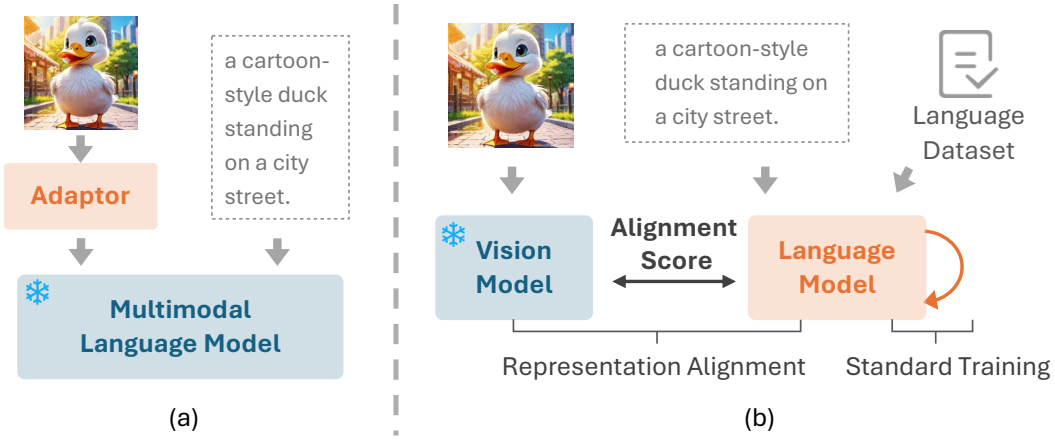

Figure 1. (a) **Motivation.** Previous works usually freeze the language model when incorporating the visual modality; otherwise, the additional modality often hurts its performance. However, our goal is for the vision modality to benefit the language model [2, 24, 32, 66]. (b) **Our Method**. We incorporate representation alignment with other modalities, such as vision models, during the training stage of the language model to improve its performance.

limited specifically to VLMs.

To address these limitations, we propose a Cross-Modal Alignment Regularization (CMAR) approach specifically designed to enhance language models by directly aligning their representations with vision models.

Exploring this question is important. Finding a way to enhance LLMs can improve the capabilities of current language models. Previous methods (e.g., Contrastive Representation Distillation [55], Cross Modal Distillation [11]) primarily conducted representation distillation in training vision models. To our knowledge, research exploring the opposite direction is scarce, mainly due to the counterintuitive idea of improving powerful LLMs using vision models.

Addressing this question is challenging. Models like GPT-4o, Deepseek-VL, and Flamingo [2, 24, 32, 66] show that integrating image inputs into LLMs, VLMs, or Multimodal LLMs (MLLMs) offers minimal improvement and sometimes even hurts their performance on language-only tasks. Thus, a common practice in MLLM or VLM training is to keep the LLM fixed to avoid performance decline from visual integration.

The Platonic Representation Hypothesis [18] observes a positive correlation between language-vision representation alignment in *off-the-shelf* models and language model performance on Hellaswag [62]. This correlation suggests the performance of language models is related to their alignment with vision models. Therefore, it is natural to wonder if integrating this alignment directly into the optimization process can help.

In CMAR, we directly align hidden representations of a language model with those of a fixed, pre-trained vision model during training. Our method introduces a regularization term to the objective function of the language model, encouraging alignment with specific vision model layers. Experimental results demonstrate that CMAR consistently improves performance in both pre-training and fine-tuning settings across tasks involving multi-task language understanding, causal reasoning, commonsense reasoning, and mathematical reasoning.

We empirically validate our approach in both fine-tuning and pre-training settings. In the pre-training scenario, CMAR achieves *4.33%* accuracy improvement on COPA (*causal reasoning*) and *1.58%* on LAMBADA (*language modeling*). When fine-tuning, CMAR also shows gains across six benchmarks, including *2.65%* on GSM8K (*mathematical reasoning*) and *1.35%* on Winogrande (*commonsense reasoning*).

In summary, the contributions of our method three folds:

- We introduce a cross-modal alignment regularization method that enables vision representations to benefit language models, overturning the belief that incorporating visual modalities harms language modeling performance.
- Our method is free from constraints in traditional distillation methods, such as shared architecture and feature dimensionality, by performing alignment through dimension-agnostic similarity measures (e.g., kernel- or correlation-based metrics), which operate on the structure of representations. This makes our approach broadly applicable.
- We validate our approach in both pre-training and fine-tuning settings across various model families, downstream tasks, and alignment measures. CMAR consistently improves language model performance, opening new avenues for language model training.

2. Related Work

In this section, we review existing approaches in multi-modal representation alignment (Section 2.1), unimodal alignment methods (Section 2.2), and model merging techniques (Section 2.3), highlighting how our approach addresses their

limitations and extends beyond current methodologies.

2.1. Multimodal Representation Alignment

Multi-modal alignment integrates cross-modality information to enhance model robustness. Prior methods rely on outputs from one modality to augment another. For example, [48] and [36] generate synthetic images via language models for training vision models; [5] creates controlled images through programs. In robotics, [17] utilizes language models for zero-shot task execution, and [23, 38, 42] employ language knowledge to guide robotic actions. While effective, such indirect approaches limit cross-modal representation alignment. In contrast, our method removes this limitation, directly aligning LLMs to the vision model in the representation space at specific layers to improve LLMs.

2.2. Unimodal Representation Alignment

Representation alignment within a single modality has received considerable attention in recent deep learning research. Examples include sharing task vectors across tasks [25], aligning diffusion models with advanced vision models [60], and enhancing tokenization with vision semantics [4]. Additionally, [45] fine-tunes language models using synthetic data for improved coherence, [52] shows alignment enables training of previously untrainable networks, and [37] aligns vision encoders with action decoders. ARDTs [9] further distills large language models into simpler decision-tree structures. Unlike the unimodal methods, our method is aimed at using alignment between vision and language modalities to improve LLMs.

2.3. Model Merging and Knowledge Distillation

Model merging methods [1, 27, 44, 58, 59] combine multiple pre-trained models to result in a model with improved performance, but require identical architectures and feature dimensions. Similarly, model distillation approaches, such as Knowledge Distillation [16] and Contrastive Representation Distillation [55], operate under similar restrictions. Cross Modal Distillation [11] expanded distillation to multiple visual modalities, yet cross-modality between vision and language remains unexplored. Practical applications of such techniques often require methods operating on the same architectures and dimensions, motivating our method to operate agnostic to architecture configuration of the models. Our research addresses these limitations by supporting distillation and merging without requiring identical architectures or dimensions across vision and language modalities.

3. Method: Cross-Modal Alignment Regularization for Language Model Training

Our goal is to enhance the language model by leveraging knowledge from a fixed pre-trained vision model. Traditional regularization methods, mostly designed for training vision models, operate within a single modality and ignore rich semantic structures from other modalities. In contrast, our proposed Cross-Modal Alignment Regularization (CMAR) directly encourages the language model to align its intermediate representations with those of the vision model, guiding it to learn more robust and semantically grounded features. As the training workflow illustrated in Figure 2, CMAR is applicable for both training and fine-tuning language models.

3.1. Language and Vision Representations

To implement CMAR, we first require an image-text dataset containing paired text-image samples. We process these two modalities using (i) a language model (LM) being trained, and (ii) a fixed pre-trained vision model that remains unchanged during training.

Language Model Representations. The language model takes an input and forwards it through all layers. By default, we extract its hidden representation from the layer-of-your-choice; see Section 5 for the ablation study on different layers). We define $H^l(\theta) \in \mathbb{R}^{B \times d_L}$ as the language model representation for a mini-batch of size B, where θ denotes the parameters of the LM and d_L is the dimensionality of the representation.

Vision Model Representations. For corresponding images, we use a *fixed, pre-trained* vision model to generate visual features. Various vision models (e.g., DINOv2 [33], MAE [14], or CLIP [43]) can be employed, please see Appendix for all vision models. We denote the vision model's output for the same mini-batch as $H^v(\varphi) \in \mathbb{R}^{B \times d_V}$, where φ represents the fixed parameters and d_V denotes its feature dimensionality. Note that matching dimensionalities ($d_L = d_V$) is *not* required, as many alignment metrics (e.g., CKA) are inherently invariant to dimensionality.

3.2. Cross-Modal Alignment

To align language and vision representations, we introduce an alignment function that measures the similarity between the two feature sets. Higher values indicate stronger alignment. In practice, one can choose from a range of similarity measures like kernel similarity and distance measures.

Alignment Measure. Formally, let $H^v(\varphi) \in \mathbb{R}^{B \times d_V}$ and $H^l(\theta) \in \mathbb{R}^{B \times d_L}$ be the vision and language features for a mini-batch of size B. We define a similarity function: $S\big(H^v(\varphi),\, H^l(\theta)\big) \in \mathbb{R}$, which outputs a scalar that is normalized into the interval $[0, 1]$, depending on the metric used. For example, one can adopt a dot-product based measure, use a radial basis function (RBF) kernel, or compute Pearson correlation coefficients. The key requirement is that higher scores reflect better alignment between the two modalities.

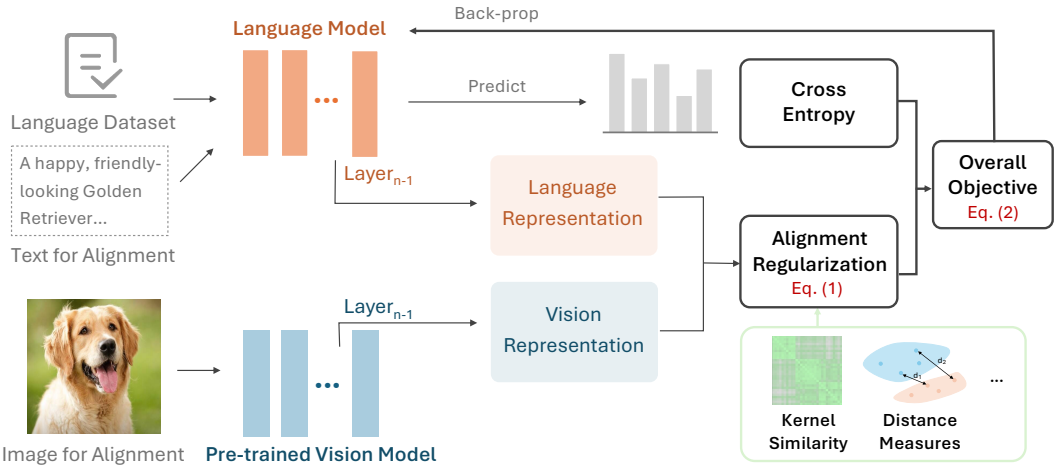

Figure 2. **The Workflow of the Training Process.** Text and image inputs are processed to generate language and vision representations respectively, which are used to calculate the alignment score using similarity measures (e.g. kernel similarity measures and direct distance measures). The overall objective augments the cross-entropy loss with our proposed alignment regularization term to update the language model using backpropagation. Note that the Pre-trained Vision Model is fixed.

3.3. Training Objective

We integrate this alignment score into the training loss as a regularization term. Intuitively, we would like to maximize the similarity $S\big(H^v(\varphi),\, H^l(\theta)\big)$. Thus, we define our alignment loss as:

$$L_{\text{align}}(\theta, \varphi) = \sum_{i=1}^{M} \Big[1 - S\big(H^v(\varphi),\, H^l(\theta)\big)\Big], \qquad (1)$$

where M is the number of mini-batches. By minimizing L_{align}, we maximize the similarity score S.

Combined Training Loss. We then add this alignment loss to the standard language modeling loss, usually the negative log-likelihood (NLL) or cross-entropy loss:

$$\theta^*, \varphi^* = \arg\min_{\theta,\varphi} \Big[L_{\text{NLL}}(\theta) \;+\; \lambda\, L_{\text{align}}(\theta, \varphi)\Big]. \qquad (2)$$

Note that φ is not updated since we keep the vision model completely fixed; in practice, we set $\nabla_\varphi = 0$. The scalar λ balances the importance of the language modeling objective versus cross-modal alignment. In our experiments, we determine λ via hyperparameter sweeping.

3.4. Simple Implementation

CMAR can be efficiently implemented without specialized operations (Algorithm 1). Computing alignment within mini-batches controls memory usage and simplifies integration into existing language model pipelines. We show in Section 4 that CMAR is applicable and effective for both pre-training and fine-tuning the language model.

4. Experiments on Pre-training and Fine-tuning Language Models with CMAR

To investigate whether alignment with vision models benefits language models trained from scratch, we first apply CMAR during pre-training and then evaluate the performance on downstream tasks.

4.1. Experiment Setup

4.1.1. Models and Datasets.

Models Used for Pre-training We use GPT-2 (124M) [39] and adapt the pipeline from NanoGPT [20]. GPT-2 is trained from scratch on the OpenWebText [10] dataset, which is a large-scale text corpus for text generation.

Models Used for Fine-tuning We experiment with several families of large language models, including Mistral, Phi-3, Qwen, DeepSeek R1-Distill models, and Llama-3 series. Unless otherwise specified, the fine-tuning pipeline follows hyperparameter settings similar to those used for GPT-2.

4.1.2. Baseline and Variants of CMAR.

We compare baseline and variants of CMAR as follows:

(1) **Baseline**. We use the language model training without regularization term as the baseline to compare our method with.

(2) **KL-Distillation** [16, 35]. In the absence of shared output space or soft labels, we approximate distillation by minimizing the KL divergence between the temperature-scaled similarity distributions. This acts as a soft alignment between the vision and language representations.

(3) **InfoNCE** [31]. InfoNCE is used as the objective function for training CLIP [40] that maximizes the similar-

Algorithm 1 Train Language Models with CMAR

```
# D_lan: Language dataset for training, D_img-text: image-text dataset, T: max iterations.
Initialize parameters θ
AlignmentScore_initial ← ComputeAlign(LanguageModel, D_img−text)                ▷ Initial alignment score
for t = 1 to T do
    (X, X[1:]) ~ D_lan;   L_NLL = CE(LanguageModel(X), X[1:])
    (X_img, Y_txt) ~ D_img−text
    H^l = HiddenStates(LanguageModel, Y_txt);
    H^v = HiddenStates(VisionModel, X_img)                                      ▷ Fixed, pre-trained vision model
    L_NLL += CE(LanguageModel(Y_txt), Y_txt[1:])                                ▷ Cross-entropy loss for text in image-text pairs
    L_align = 1 − AlignmentMeasure(H^v, H^l)                                    ▷ Proposed alignment measures
    Loss ← L_NLL + λ L_align
    Backpropagate Loss and update θ
end for
AlignmentScore_after ← ComputeAlign(LanguageModel, D_img−text)                  ▷ Final alignment score
```

CE: Cross-entropy loss; **AlignmentMeasure**: A chosen similarity measure (e.g., correlation, kernel-based similarity).

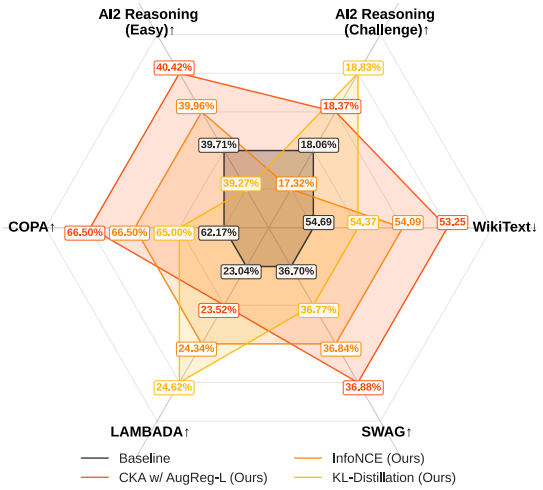

Figure 3. **Experimental Results for Pre-training GPT-2.** Results averaged over 5 runs indicate CMAR consistently improves performance on all six datasets.

ity between matching image-text pairs and minimizes the similarity for non-matching pairs.

(4) **Kernel Metrics** (**CKA**). Kernel-based similarity metrics compare the structural relationships between feature spaces by applying kernel functions before computing alignment. These metrics capture how closely the representation structures of different modalities correspond. We adopt Centered Kernel Alignment (CKA) [22] as a representative kernel metric in this paper.

4.1.3. Vision Models and Image-Text Dataset Used for Calculating the Alignment Score.

We align the language model with seven different types and sizes of vision models, including MAE [13], DinoV2 (four sizes) [33], CLIP [40], and AugReg [51]. We obtain image caption pairs with caption length of 10 words from the SA-1B [21] dataset alignment.

4.1.4. Evaluation Datasets.

Evaluation Datasets Used for Pre-training. We evaluate on six datasets spanning next-token prediction, causal reasoning, and commonsense reasoning tasks: AI2 Reasoning Challenge (Challenge) [6], AI2 Reasoning Challenge (Easy) [6], COPA [28], LAMBADA, [34] SWAG [61], and WikiText [29]. A summary is provided in Appendix.

Evaluation Datasets Used for Fine-tuning. Following CoCoMix [53] and Deepseek-R1 [8], we evaluate our model on a diverse set of tasks ranging from language modeling, commonsense reasoning, and knowledge-intensive QA to mathematical reasoning, covering widely used benchmark datasets and demonstrating its broad robustness.

4.2. Main Results

A common assumption in the community is that adding visual signals to language-model training often hurts pure-language performance. Our findings challenge this view: with a frozen vision encoder and no inference-time modalities, CMAR yields consistent improvements across diverse LLMs and tasks, showing that visual priors can offer gains.

Main Results for Pre-training Language Models. To assess the improvement in the language model resulting from being aligned with a vision model during pre-training, we apply CMAR to GPT-2 and evaluate on six standard benchmarks. As shown in Figure 3, all CMAR variants (KL-distillation , InfoNCE , and CKA) outperform the Baseline in average performance.

CMAR-CKA achieves the highest overall average accuracy (*+1.20%* over baseline), followed closely by CMAR-InfoNCE (*+1.05%*) and CMAR-KLD (*+0.96%*). CMAR-CKA leads on four out of five datasets. Specifically, CMAR-CKA achieves *4.33%* performance gain on COPA and *1.15%* gain on AI2 Reasoning (Easy). Meanwhile, other variants

Table 1. **Experimental Results under Fine-Tune Setting.** We evaluate CMAR across seven language models on six widely-used benchmarks. We compare CMAR using KL-Distillation, InfoNCE, and CKA as different alignment measures, and observe consistent improvements over the Baseline. CMAR provides larger gains on logic-heavy tasks.

Model	Method	Hellaswag ↑	Winogrande ↑	MathQA ↑	SWAG ↑	Comm.QA ↑	GSM8k ↑
Llama-3-8B	Baseline	59.15	73.64	38.79	57.74	69.45	41.55
	CMAR-KLD	59.20	**74.74** +1.10 ↑	38.36	57.81	69.45	**44.20** **+2.65** ↑
	CMAR-InfoNCE	59.32	73.80	38.63	57.73	69.29	43.59
	CMAR-CKA	**59.24** +0.09 ↑	74.50	**38.90** +0.11 ↑	**58.07** +0.33 ↑	**69.74** +0.29 ↑	43.30
DeepSeek-R1-Distill-Llama-8B	Baseline	55.23	67.80	33.57	54.32	61.18	62.10
	CMAR-KLD	55.18	**68.27** +0.47 ↑	**33.77** +0.20 ↑	54.30	**61.67** +0.49 ↑	63.31
	CMAR-InfoNCE	**55.23** +0.00 ↑	67.80	33.57	54.32	61.18	**64.44** +2.34 ↑
	CMAR-CKA	54.33	68.11	32.83	**55.06** +0.74 ↑	61.25	64.23
Qwen2.5-7B	Baseline	60.44	68.50	39.80	57.05	82.23	75.04
	CMAR-KLD	60.36	69.62	**39.93** +0.13 ↑	57.06	82.23	75.97
	CMAR-InfoNCE	60.44	69.73	39.80	57.05	82.23	76.19
	CMAR-CKA	**61.25** +0.81 ↑	**69.85** **+1.35** ↑	39.85	**57.23** +0.18 ↑	**83.00** +0.77 ↑	**76.24** +1.20 ↑
Mistral-7B-v0.2	Baseline	58.89	72.53	35.08	57.06	59.95	29.98
	CMAR-KLD	59.91	72.91	35.15	57.39	59.65	31.16
	CMAR-InfoNCE	59.56	**72.98** +0.45 ↑	35.46	**57.61** +0.55 ↑	59.80	31.24
	CMAR-CKA	**60.10** +1.21 ↑	72.89	**36.17** **+1.09** ↑	57.10	**60.10** +0.15 ↑	**31.50** +1.52 ↑
Phi-3-Medium -4k-Instruct	Baseline	64.01	75.53	44.56	60.78	75.10	79.58
	CMAR-KLD	64.59	75.63	**45.06** +0.50 ↑	60.95	74.45	81.47
	CMAR-InfoNCE	64.51	75.59	44.50	60.93	**74.90** -0.20 ↓	**81.88** +2.30 ↑
	CMAR-CKA	**65.17** +1.16 ↑	**75.69** +0.16 ↑	45.00	**61.59** +0.81 ↑	74.39	81.02
Llama-3.1-8B	Baseline	57.73	73.01	39.16	56.35	68.25	47.35
	CMAR-KLD	**58.92** **+1.19** ↑	**73.40** +0.39 ↑	**39.97** +0.81 ↑	**57.56** +1.21 ↑	69.21	**47.23** -0.12 ↓
	CMAR-InfoNCE	**58.92** **+1.19** ↑	72.93	39.80	57.52	69.29	47.08
	CMAR-CKA	58.61	**73.40** +0.39 ↑	**39.97** +0.81 ↑	57.16	**69.78** **+1.53** ↑	47.08
Llama-3.2-3B	Baseline	50.83	67.56	34.67	52.28	66.01	58.61
	CMAR-KLD	51.07	67.17	34.61	52.23	**66.34** +0.33 ↑	**59.89** +1.28 ↑
	CMAR-InfoNCE	**51.08** +0.25 ↑	67.25	34.97	52.23	66.01	59.14
	CMAR-CKA	50.43	**67.97** +0.41 ↑	**35.21** +0.54 ↑	**52.95** +0.67 ↑	66.09	58.73

KLD: KL-Distillation. **CKA**: Centered Kernel Alignment. **Comm.QA**: CommonsenseQA.

are competitive on some datasets. For example, CMAR-KLD performs best on LAMBADA (*+1.58%*) and AI2 Reasoning (Challenge) (*+0.77%*). This indicates that different alignment measures can capture complementary aspects of representational compatibility. More details are in Appendix.

Main Results for Fine-tuning Language Models. To measure the efficacy of CMAR to align a language model with a vision model during fine-tuning, we apply CMAR during the fine-tuning of seven language models and evaluate performance across six diverse downstream tasks.

As shown in Table 1, all CMAR variants (KL-distillation, InfoNCE, and CKA) outperform the Baseline across all model families and most tasks. On average, CMAR-CKA achieves the highest overall accuracy in four out of seven models, while CMAR-KLD and CMAR-InfoNCE remain competitive, occasionally outperforming on specific tasks (e.g., GSM8K and MathQA). Logic-intensive datasets such as GSM8K, MathQA, and CommonsenseQA benefit most from alignment—for instance, CMAR-KLD boosts GSM8K by *+2.65%* on LLaMA-3-8B, and CMAR-CKA improves MathQA by *+1.09%* on Mistral-7B-v0.2. Larger models like Qwen2.5-7B also exhibit strong gains, with CMAR-CKA improving Winogrande by *+1.35%* and GSM8K by *+1.20%* respectively. These results show that CMAR generalizes across scales and architectures, with different alignment measures providing complementary benefits.

5. Analysis

To better understand the properties of CMAR, we conduct a series of ablation studies as follows.

(a) If the Performance Gains are Truly from Vision Modality? To verify that CMAR's improvements arise from vision modality rather than generic regularization or auxiliary supervision, we conduct two controls. As shown in Figure 4(a), (1) We compare two CMAR variations with traditional *L1*, *L2*, and *KoLeo* [46] regularizers applied to the language model; none matches CMAR's gains. (2) We align the language model to an untrained/randomized vision encoder (denoted as *CMAR (RI)*), keeping all other settings identical to the full CMAR setting; this underperforms the *CMAR (Full)* method. These results indicate that CMAR's benefits derive from pretrained visual representations rather than from regularization alone.

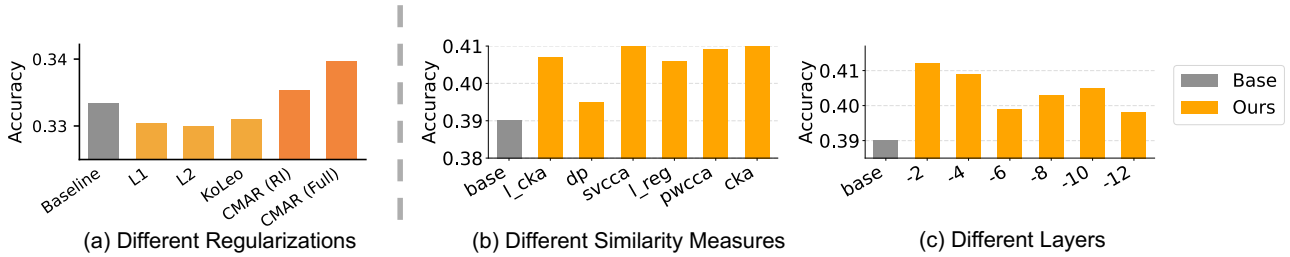

Figure 4. **Ablation Studies.** (a) **Comparison CMAR with Different Regularization Methods.** CMAR outperforms other regularization methods. (b) **The Effect of using Different Alignment Measures.** All alignment metrics work, among which CKA works the best. (c) **The Effect of Aligning with Different Layers.** Aligning the penultimate layer yields the best performance on average.

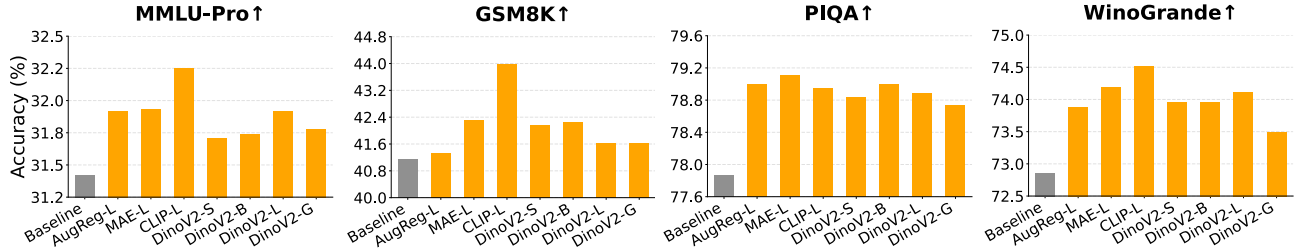

Figure 5. **The Effect of Different Vision Models.** CMAR (orange columns) consistently outperforms the baseline (gray columns), particularly in reasoning-intensive tasks such as GSM8K, PIQA, and MMLU-Pro.

(b) Our method is robust across different alignment metrics and across different model layers. As shown in Figure 4(b), we conduct experiments to compare the effect of different similarity metrics such as CKA [22], SVCCA [41], and PWCCA [30], and we observe that CKA yields the highest accuracy among the metrics tested. That said, other similarity metrics such as SVCCA also lead to notable gains, suggesting that a range of representation-similarity measures can be beneficial. Additionally, in Figure 4(c), we compare alignment at different layers in GPT-2. While aligning the penultimate layer yields the best performance on average, some tasks benefit from deeper or multi-layer alignment.

(c) Effect of Different Vision Models. Aligning LLaMA-3-8B with different vision models consistently improves performance on downstream tasks. Alignment with CLIP achieves the best results on six out of eight datasets, including the challenging multi-task language understanding benchmark MMLU-Pro [15]. As shown in Figure 5, all seven vision models used for CMAR-CKA outperform the baseline across most datasets. Interestingly, model strength (in terms of vision model scale or pre-training dataset size) does not always correlate with downstream performance. For example, DINOv2-Large and DINOv2-Giant do not always perform better than their smaller counterparts (DINOv2-Small/Base). This highlights that representation compatibility, rather than vision model capacity alone, may be a more important factor for effective alignment.

(d) Effect of Caption Length. We also discover that caption length plays a nuanced role in alignment quality. In Figure 7, increasing caption length from short (5 words) to moderate (10–20 words) improves performance. However, further extending captions to 200 words yields diminishing or slightly negative returns, possibly due to noise in longer textual descriptions that dilute relevant context.

5.1. Correlation of Alignment and Performance across Checkpoints.

To explore the relationship of alignment and performance, we train the models with CMAR-CKA and keep track of the CKA score as well as the performance across different checkpoints and different model sizes. Figure 6 illustrates the positive relationship between representation alignment and model performance across different training settings, datasets, and model families.

Pre-training Settings. In Figure 6(a), we observe a consistent and strong positive relationship between the alignment score and downstream performance. The alignment scores increase monotonically from 0.79 to 0.86, and correspond closely to improved task performance (y-axis), with Pearson correlation $r > 0.95$ across all tasks. These results not only reinforce our hypothesis that CMAR is an effective regularization term, but also reveal that the metrics used in CMAR can serve as reliable indicators for predicting performance.

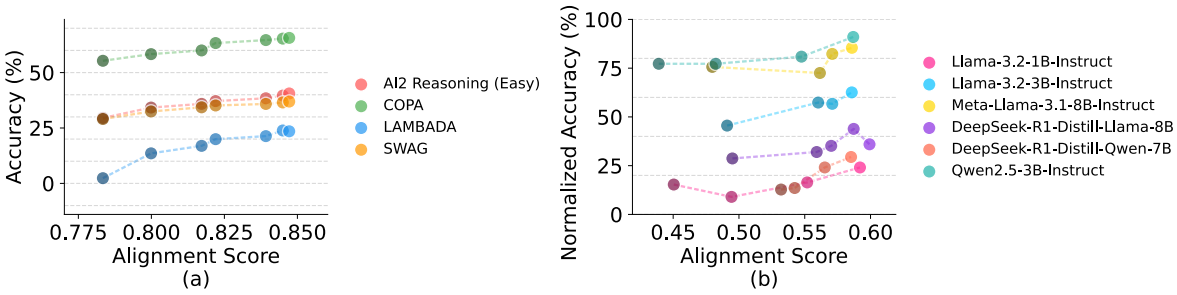

Figure 6. **Correlation of Alignment and Performance.** Darker shades indicate earlier checkpoints. (a) **Pre-training Setting:** Alignment score increases and correlates positively with performance across tasks. Colors denote tasks. (b) **Fine-tuning Setting:** Higher alignment scores correlate with better performance (normalized accuracy) across model checkpoints. Colors denote model families.

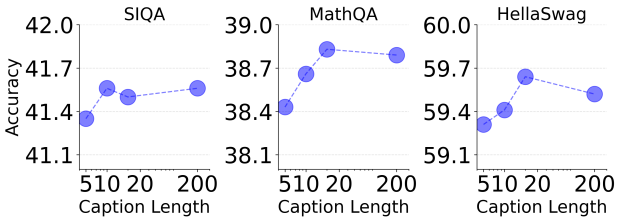

Figure 7. **Ablation Study on the Effect of Caption Length.** Moderately longer captions can enhance performance, though gains saturate or slightly drop at very long lengths.

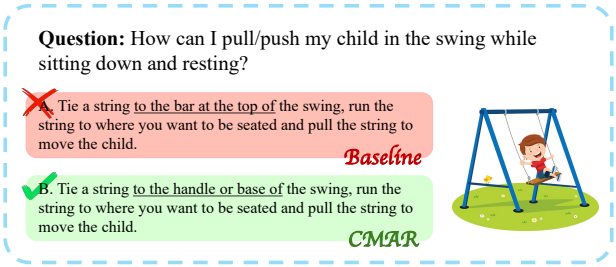

Figure 8. An example from PIQA in which CMAR benefits from better understanding of swing mechanism.

Fine-tuning Settings. As shown in Figure 6(b), we observe a clear, positive relationship between a model's alignment score (x-axis) and its normalized task accuracy (y-axis) across all seven different language models. Each curve traces a different language model, including instruct-turned and distilled models at different scales. Models with higher alignment scores are more likely to achieve better downstream performance: for example, Qwen2.5-3B-Instruct attains the highest alignment (≈*0.60*) and yields the top normalized accuracy (≈*90%*), while the distilled DeepSeek variants remain lower alignment (≈*0.47–0.55*) and correspondingly lower accuracy (≈*12–45%*). These results suggest that the alignment score is not only predictive of but also instrumental to model performance when training with CMAR.

5.2. Case Studies

We conduct case studies to investigate problems for which alignment with vision models helps. Across the examples, we observe recurring patterns: items that appear to benefit most from the vision-aligned model tend to evoke implicit spatial, physical, or diagrammatic structure, even when presented purely as text. These include geometry problems where relative areas must be visualized, multi-step chemistry questions that hinge on symmetry or stereochemical layout, and commonsense tasks whose solutions depend on the affordances of objects (e.g., where to apply force on a swing in Fig. 8). While we do not claim that these improvements constitute evidence of systematic visual reasoning, the consistency of these cases suggests that introducing vision-related information may shift the model's internal representations toward more grounded, world-aware interpretations of previously ambiguous text. In that sense, the gains may reflect not a domain-specific visual competence, but a broader tendency for vision-aligned features to stabilize physical intuitions. This observation remains tentative, yet it provides a useful interpretive lens for understanding where alignment with visual encoders might exert its qualitative effects. More examples can be found in the Appendix.

6. Conclusion

In this paper, we introduce CMAR that aligns the representations of a language model with those of a fixed, pre-trained vision model, distilling visual knowledge into language models to improve the performance. We observe consistent performance gains across different models and evaluation datasets in both pre-training and fine-tuning settings under various alignment measures. However, the effectiveness of CMAR can be affected by downstream task types. A deeper understanding of how alignment interacts with task characteristics can guide a targeted model training process. While this work focuses on aligning language models with fixed, pre-trained vision models, the formulation of CMAR is general and may be extended to other modalities, such as audio and robotics, for broader cross-modal knowledge transfer and distillation, opening up directions for the future research.

Acknowledgement

Research was sponsored by the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes, 2024. 3

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems, 35:23716–23736, 2022. 2

[3] Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms, 2019. 3

[4] Zechen Bai, Jianxiong Gao, Ziteng Gao, Pichao Wang, Zheng Zhang, Tong He, and Mike Zheng Shou. Factorized visual tokenization and generation, 2024. 3

[5] Manel Baradad, Richard Chen, Jonas Wulff, Tongzhou Wang, Rogerio Feris, Antonio Torralba, and Phillip Isola. Procedural image programs for representation learning. Advances in Neural Information Processing Systems, 35:6450–6462, 2022. 1, 3

[6] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. ArXiv, abs/1803.05457, 2018. 5, 3

[7] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021. 3

[8] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. 5

[9] Yulu Gan, Tomer Galanti, Tomaso Poggio, and Eran Malach. On the power of decision trees in auto-regressive language modeling. arXiv preprint arXiv:2409.19150, 2024. 3

[10] Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus, 2019. 4

[11] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2827–2836, 2016. 2, 3

[12] Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 8154–8173, 2023. 1

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 16000–16009, 2022. 5, 2

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 16000–16009, 2022. 3

[15] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. 7, 3

[16] Geoffrey Hinton. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015. 3, 4, 1

[17] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents, 2022. 3

[18] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. arXiv preprint arXiv:2405.07987, 2024. 2

[19] Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, et al. Dinov2 meets text: A unified framework for image-and pixel-level vision-language alignment. arXiv preprint arXiv:2412.16334, 2024. 1

[20] Andrej Karpathy. https://github.com/karpathy/nanoGPT, 2023. 4

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. 5

[22] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In International Conference on Machine Learning (ICML), pages 3519–3529, 2019. 5, 7, 3

[23] Jacky Liang, Wenlong Huang, F. Xia, Peng Xu, Karol Hausman, Brian Ichter, Peter R. Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. 2023 IEEE International Conference on Robotics and Automation (ICRA), pages 9493–9500, 2022. 3

[24] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. Deepseek-vl: towards real-world vision-language understanding. arXiv preprint arXiv:2403.05525, 2024. 2

[25] Grace Luo, Trevor Darrell, and Amir Bar. Task vectors are cross-modal. arXiv preprint arXiv:2410.22330, 2024. 3

[26] David Marr. Vision: A computational investigation into the human representation and processing of visual information. MIT press, 2010. 1

[27] Michael Matena and Colin Raffel. Merging models with fisher-weighted averaging, 2022. 3

[28] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer Sentinel Mixture Models, 2016. 5, 2, 3

[29] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. 5, 2, 3

[30] Ari S Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In Advances in Neural Information Processing Systems (NeurIPS), 2018. 7, 3

[31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018. 4

[32] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, et al. Gpt-4 technical report, 2024. 2

[33] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193, 2023. 3, 5, 2

[34] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset, 2016. 5, 3

[35] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In CVPR, 2019. 4

[36] Yi-Hao Peng, Faria Huq, Yue Jiang, Jason Wu, Amanda Li, Jeffrey P. Bigham, and Amy Pavel. Dreamstruct: Understanding slides and user interfaces via synthetic data generation. In European Conference on Computer Vision, 2024. 3

[37] Han Qi, Haocheng Yin, and Heng Yang. Control-oriented clustering of visual latent representation. ArXiv, abs/2410.05063, 2024. 3

[38] Jielin Qiu, Mengdi Xu, William Jongwon Han, Seungwhan Moon, and Ding Zhao. Embodied executable policy learning with language-based scene summarization. ArXiv, abs/2306.05696, 2023. 3

[39] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1:9, 2019. 4

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1, 4, 5, 2

[41] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In Advances in Neural Information Processing Systems (NeurIPS), 2017. 7, 3

[42] S. Sundar Raman, Vanya Cohen, Eric Rosen, Ifrah Idrees, David Paulius, and Stefanie Tellex. Cape: Corrective actions from precondition errors using large language models. 2024 IEEE International Conference on Robotics and Automation (ICRA), pages 14070–14077, 2022. 3

[43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 3

[44] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search, 2019. 3

[45] Xuan Ren, Biao Wu, and Lingqiao Liu. I learn better if you speak my language: Understanding the superior performance of fine-tuning large language models with llm-generated responses. In Conference on Empirical Methods in Natural Language Processing, 2024. 3

[46] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search, 2019. 6

[47] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019. 3

[48] Pratyusha Sharma, Tamar Rott Shaham, Manel Baradad, Stephanie Fu, Adrian Rodriguez-Munoz, Shivam Duggal, Phillip Isola, and Antonio Torralba. A vision check-up for language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14410–14419, 2024. 1, 3

[49] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge, 2018. 3

[50] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers, 2021. 2

[51] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers, 2022. 5

[52] Vighnesh Subramaniam, David Mayo, Colin Conwell, Tomaso Poggio, Boris Katz, Brian Cheung, and Andrei Barbu. Training the untrainable: Introducing inductive bias via representational alignment. arXiv preprint arXiv:2410.20035, 2024. 3

[53] Jihoon Tack, Jack Lanchantin, Jane Yu, Andrew Cohen, Ilia Kulikov, Janice Lan, Shibo Hao, Yuandong Tian, Jason Weston, and Xian Li. Llm pretraining with continuous concepts. arXiv preprint arXiv:2502.08524, 2025. 5

[54] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4149–4158, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. 3

[55] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. arXiv preprint arXiv:1910.10699, 2019. 2, 3

[56] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems, 2020. 2

[57] Lionel Wong, Gabriel Grand, Alexander K Lew, Noah D Goodman, Vikash K Mansinghka, Jacob Andreas, and

Joshua B Tenenbaum. From word models to world models: Translating from natural language to the probabilistic language of thought. arXiv preprint arXiv:2306.12672, 2023. 1

[58] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models, 2023. 3

[59] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch, 2024. 3

[60] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. arXiv preprint arXiv:2410.06940, 2024. 3

[61] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2018. 5, 3

[62] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? arXiv preprint arXiv:1905.07830, 2019. 2

[63] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019. 3

[64] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 18123–18133, 2022. 1

[65] Le Zhang, Qian Yang, and Aishwarya Agrawal. Assessing and learning alignment of unimodal vision and language models, 2024. 1

[66] Yi-Kai Zhang, Shiyin Lu, Yang Li, Yanqing Ma, Qingguo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, De-Chuan Zhan, and Han-Jia Ye. Wings: Learning multimodal llms without text-only forgetting. Advances in Neural Information Processing Systems, 37:31828–31853, 2024. 2