Seeing Helps Reasoning in Language Models

Supplementary Material

7. Mathematical Formulation of Our Method Variants

Definition 1 (Cross-Modal Representations)**.** *Let V be a vision feature space (e.g., the output of a fixed, pre-trained vision model) and L be a text feature space (e.g., hidden states from a language model). We denote x as an image and y as the corresponding text (e.g., a caption). Generally, we assume that the features are mean-centered.*

7.1. KL-Distillation

Traditional knowledge distillation transfers knowledge from a teacher model to a student by minimizing the KL divergence between their output distributions over a shared label space [16]. However, in our setting, the vision model (teacher) and the language model (student) do not share a common output space, and the teacher does not produce soft labels for the student's task.

To address this, we approximate the distillation process by aligning internal representational structures instead of outputs. Specifically, we project and normalize the vision representation f_v and language representation f_l with learned MLP layers $g_v(\cdot)$ and $g_l(\cdot)$. We then normalize these projected representations using the ℓ_2-norm to obtain unit-length vectors $f_v(\cdot)$ and $f_l(\cdot)$, as follows:

$$f_v = \frac{g_v(x)}{\|g_v(x)\|_2}, \quad f_l = \frac{g_l(y)}{\|g_l(y)\|_2}, \tag{3}$$

We compute pairwise similarity matrices for a batch with sample size B: $S_v = \frac{f_v f_v^\top}{\tau}, \quad S_l = \frac{f_l f_l^\top}{\tau}$,

where τ is the temperature scaling factor. Then we convert them to soft probability distributions $P_l = \text{softmax}(S_l)$, and $P_v = \log \text{softmax}(S_v)$.

The distillation loss minimizes KL divergence between the distributions:

$$\mathcal{L}_{\text{KL}} = \tau^2 \cdot \frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{B} P_{l,ij} \cdot (-P_{v,ij}) \,, \tag{4}$$

The τ^2 factor compensates for gradient scaling introduced by the temperature.

7.2. InfoNCE

To compute the InfoNCE loss, we first project language and vision features to a shared embedding space and apply ℓ_2 normalization, same as Equation 3.

Given B samples in a batch $\{(f_{l_i}, f_{v_i})\}_{i=1}^{B}$, we compute the similarity matrix $S_{ij} = \frac{f_{l_i}^\top f_{v_j}}{\tau}$, where τ is a temperature hyperparameter (default $= 0.07$). The InfoNCE loss is defined as the average of two cross-entropy losses (text-to-image and image-to-text):

$$\mathcal{L}_{\text{InfoNCE}} = \frac{1}{2B} \sum_{i=1}^{B} \left[-\log \frac{\exp(S_{ii})}{\sum_{j=1}^{B} \exp(S_{ij})} - \log \frac{\exp(S_{ii})}{\sum_{j=1}^{B} \exp(S_{ji})} \right] \tag{5}$$

7.3. Centered Kernel Alignment (CKA)

Kernel Computation. Let $H^v(\psi)$ and $H^l(\theta)$ be two sets of features, of dimensions $B \times d_V$ and $B \times d_L$ respectively, where B is the batch size, and d_V and d_L denote the feature dimensions for vision and language. We use a radial basis function (RBF) kernel to form the kernel matrices K^v and K^l. For sample i and sample j:

$$K^v_{ij} = k\big(H^v_i, H^v_j\big), \quad K^l_{ij} = k\big(H^l_i, H^l_j\big), \tag{6}$$

where H^v_i and H^l_i denote the i-th row of H^v and H^l, respectively. The Gaussian RBF kernel is particularly useful for capturing non-linear relationships.

CKA. After obtaining K^v and K^l, each matrix is centered by subtracting row and column means. Let $\tilde{K}^v$ and $\tilde{K}^l$ be the centered versions of K^v and K^l. The Hilbert-Schmidt Independence Criterion (HSIC) is then used to measure the similarity between these centered matrices:

$$\text{HSIC}\big(\tilde{K}^v, \tilde{K}^l\big) = \frac{1}{(B-1)^2} \,\text{trace}\Big(\tilde{K}^v \, \tilde{K}^l\Big), \tag{7}$$

The normalized CKA score is defined as:

$$\text{CKA}(K^v, K^l) = \frac{\text{HSIC}\big(\tilde{K}^v, \tilde{K}^l\big)}{\sqrt{\text{HSIC}\big(\tilde{K}^v, \tilde{K}^v\big) \,\text{HSIC}\big(\tilde{K}^l, \tilde{K}^l\big)}}, \tag{8}$$

Higher CKA values reflect stronger alignment between the fixed vision features $H^{\text{v}}(\psi)$ and the language model features $H^{\text{l}}(\theta)$.

8. Additional Experimental Setups

8.1. Implementation Details

Pre-training Setting. All experiments are conducted using PyTorch on two 80GB Nvidia H100 GPUs. For pre-training, each model is trained for 7000 iterations with a learning rate of 6e-4 and a batch size of 16. Each model is trained over approximately 16 hours, with computations distributed across

Table 2. **Vision Models used for Alignment.** We use seven vision encoders that differ in their pre-training objectives, architectures, and datasets.

Model	Size	Arch.	Pre-trained Dataset	Patch Size
MAE [13]	0.30B	ViT	IN-21K	16
DinoV2-Giant [33]	1.14B		LVD-142M	14
DinoV2-Large	0.30B		LVD-142M	14
DinoV2-Base	0.09B		LVD-142M	14
DinoV2-Small	0.02B		LVD-142M	14
CLIP [40]	0.30B		LAION-2B	16
AugReg [50]	0.33B		IN-21K	16

Table 3. **Performances across Model Sizes.** CMAR shows consistent gains over the baseline on ARC-Challenge and SWAG datasets.

Model	Method	ARC-Challenge ↑	SWAG ↑
Qwen2.5-3B-Instruct	Baseline	42.5%	53.0%
	CMAR	42.6% (+0.1%)	53.5% (**+0.5%**)
Qwen2.5-7B-Instruct	Baseline	46.6%	55.2%
	CMAR	47.4% (**+0.8%**)	55.7% (**+0.5%**)
Qwen2.5-14B-Instruct	Baseline	53.7%	60.7%
	CMAR	54.1% (+0.4%)	61.0% (+0.3%)

the GPUs to enhance training efficiency and scalability. The weight of alignment regularization, λ, is set to 0.01 after an extensive hyperparameter search. The alignment loss updates every 20 iterations, facilitating periodic integration into the training process. We employ CKA as the alignment metric in all experiments, except for those in the ablation study described in Sec. 5.

Fine-tuning Setting. All experiments are conducted using PyTorch on four 80GB Nvidia H100 GPUs. We utilize a single NVIDIA H100 for 100 iterations with a reduced learning rate of $1e-5$ and a batch size of 8.

8.2. Vision Model Used for Ablation Study

We list all seven teacher vision models in Table 2, including MAE and AugReg models pretrained on ImageNet-21K, as well as four variants of DINOv2—Giant, Large, Base, and Small—trained on the large-scale LVD-142M dataset. CLIP is also included, representing a contrastive vision–language pretrained model on LAION-2B. These models span a wide range of capacities (from 0.02B to 1.14B parameters), architectures, and training corpora, enabling us to examine how different visual priors for alignment influence downstream language model performance.

8.3. Evaluation Datasets

Datasets used for Pre-training. We used AI2 Reasoning Challenge (Challenge), AI2 Reasoning Challenge (Easy), COPA, LAMBADA, SWAG, and WikiText in our pre-training experiments.

AI2 Reasoning Challenge (ARC-Challenge) is a difficult subset of the AI2 Reasoning Challenge, containing science exam questions that requires multi-step reasoning beyond retrieval. It includes 1,172 training, 299 validation, and 1,172 test questions.

The AI2 Reasoning Challenge (ARC-Easy) is the easier subset of the AI2 Reasoning Challenge, consisting of grade-school science questions that are solvable with basic facts. It contains 2,251 training, 570 validation, and 2,376 test questions.

The *Choice of Plausible Alternatives (COPA)* [28] dataset, part of the SuperGLUE [56] benchmark, tests causal reasoning by asking participants to choose between two plausible outcomes based on a given premise, enhancing understanding of cause-effect relationships in text.

LAMBADA is a reading comprehension and long-context language modeling dataset where the goal is to predict the final word of a passage requiring understanding of long-range context. It includes $\approx 4.9K$ test cases, with 2M tokens of supporting text; training is optional and based on BookCorpus.

SWAG evaluates grounded commonsense inference via selecting plausible event continuations. It includes 113K training, 20K validation, and 20K test examples.

WikiText is a collection of high-quality language modeling corpora derived from verified Wikipedia articles. WikiText-103 contains 103M tokens in training, 218K tokens in validation, and 246K tokens in test.

Datasets used for finetuning. We use Hellaswag, Winogrande, MathQA, SWAG, Commensense QA and GSM8k for our fine-tuning experiments.

Hellaswag is a commonsense inference benchmark. It contains $\approx 409K$ training examples and 9.7K validation examples for choosing the most plausible continuation of a given context.

Winogrande is a large-scale commonsense coreference dataset with adversarially filtered pronoun-resolution questions. It includes $\approx 44K$ training examples, 1.2K validation, and 1.7K test instances.

MathQA is a math word-problem dataset requiring mapping text to solution programs and symbolic reasoning. It provides $\approx 29K$ training questions, $\approx 3K$ validation, and $\approx 3K$ test examples.

CommonsenseQA is a multiple-choice commonsense reasoning benchmark built from ConceptNet relations. It includes $\approx 9.7K$ training, 1.2K validation, and a held-out test set of $\approx 1.1K$ examples.

GSM8K is a set of grade-school math word problems that emphasize multi-step arithmetic reasoning. It includes 7.5K training and 1.3K test problems.

Wikitext [29] dataset comprises over 100 million tokens from Wikipedia's verified Good and Featured articles. Tai-

Table 4. **Summary of datasets used in evaluation.** Task types include language modeling (LM), multiple-choice (MC), commonsense reasoning, mathematics, and science QA.

Dataset	Task Type	Reference
WikiText	Next-Token Prediction (LM)	Merity et al. [29]
LAMBADA	Next-Token Prediction (LM)	Paperno et al. [34]
SWAG	Commonsense & Scenario Reasoning (MC)	Zellers et al. [61]
CommonsenseQA	Commonsense Reasoning (QA)	Talmor et al. [54]
HellaSwag	Commonsense & Scenario Reasoning (MC)	Zellers et al. [63]
WinoGrande	Coreference & Commonsense Reasoning (MC)	Sakaguchi et al. [47]
COPA	Causal Reasoning (Forced Choice)	Merity et al. [28]
ARC (AI2 Reasoning Challenge)	Science QA (Multiple Choice)	Clark et al. [6]
MMLU-Pro	Multitask Language Understanding	Hendrycks et al. [15]
MathQA	Mathematical Reasoning (QA)	Amini et al. [3]
GSM8K	Mathematical Reasoning (Step-by-Step QA)	Cobbe et al. [7]

lored for language modeling, it maintains the original article formatting, links, and structure, presenting a realistic challenge for language models.

Commonsense QA [54], derived from ConceptNet [49], challenges models to apply everyday commonsense knowledge to answer intuitive questions, pushing the limits of what AI can understand and respond to coherently.

Situations With Adversarial Generations (SWAG) [61] assesses models' ability to predict logical sentence endings in diverse scenarios. It features nearly 113,000 multiple-choice questions that test understanding and prediction of plausible outcomes.

8.4. Different Kernel Metrics Comparison

Below we describe the kernel similarity metrics we used.

CKA (RBF) [22] is a non-linear variant of centered kernel alignment that uses a radial basis function (RBF) kernel to capture higher-order feature similarities beyond linear correlations. **Linear CKA** [22] uses a linear kernel and computes the Hilbert-Schmidt Independence Criterion (HSIC), measuring relational invariances in representation spaces.

SVCCA [41] combines singular value decomposition (SVD) and canonical correlation analysis (CCA) to identify a shared subspace across models by comparing dominant directions of variation. **PWCCA** [30] builds on CCA by assigning importance weights to each canonical correlation based on explained variance, emphasizing more informative components.

Dot Product computes the mean elementwise dot product between feature vectors, offering a simple measure of alignment in the original space.

Linear Regression evaluates alignment by fitting a multi-output linear regressor from one feature space to another and measuring the average R^2 score, quantifying how well one representation predicts the other.

8.5. Results of Aligning using Different Methods.

Across the board in Table 5 of GPT-2 Pre-training, CMAR delivers consistent improvements over the baseline, with different alignment measures excelling on different task profiles. On ARC-Challenge, the hardest benchmark, CMAR-KLD lifts accuracy from 18.06% to 18.83% (+0.77%), while CMAR-CKA raises ARC-Easy from 39.71% to 40.42% (+0.71%). On conceptual reasoning tasks like COPA, CMAR-InfoNCE and CMAR-CKA both achieve 66.5%, outperforming the baseline by +4.33% points. LAMBADA benefits most from KLD, increasing by 1.58%, demonstrating that language-modeling coherence improves with representation alignment. CMAR-CKA also meaningfully reduces WikiText perplexity from 54.69 to 53.25. Taken together, these gains indicate that aligning language models to vision representations yields broad, multi-task improvements.

8.6. Consistency of Mode Performance Improvement over Different Model Size

As shown in Table 3, CMAR exhibits a clear monotonic improvement across all evaluated model sizes: models consistently yield absolute gains over their respective baselines. This pattern mirrors classic scaling-law behavior, indicating that the alignment signal introduced by CMAR is not size-limited. Consequently, extending CMAR to even larger Qwen or Llama-family models is a promising path with expected consistent model improvements.

8.7. Consistency of Model Performance Improvement over Number of Image-Text Pairs

In Figure 9, CMAR shows consistent improvements when more image-text pairs are used for alignment during fine-tuning. This pattern is consistent with data-scaling behavior: improvements follow a power-law-like trend in the small–to–mid data regime. The task-dependent saturation and small regressions at higher N argue for calibrated pair

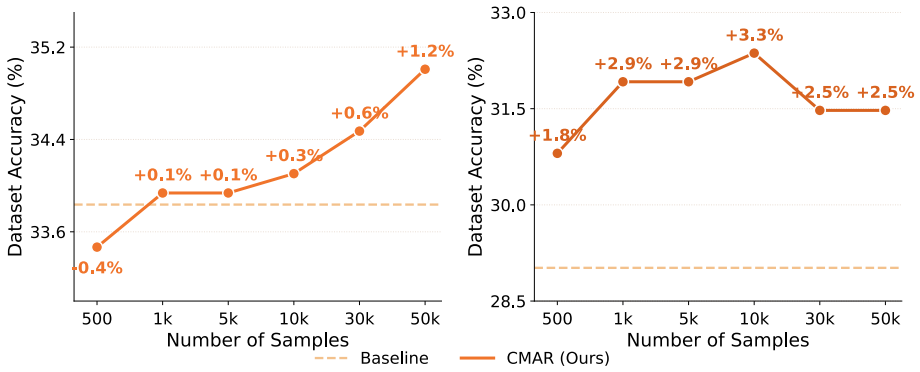

Figure 9. **Finetuned Qwen2.5-7B-Instruct Performances on MathQA and GPQA Main (zeroshot) vs. Number of Image-text Pairs.** Generally, a positive trend is observed when more image-text pairs are used during training.

Table 5. **Experimental Results.** CMAR improves performance across multiple reasoning and language understanding benchmarks using KL-Distillation, InfoNCE, or CKA as alignment measures. Results averaged over 5 runs, with WikiText evaluated on perplexity.

Method	ARC(Challenge)↑	ARC(Easy)↑	COPA↑	LAMBADA↑	SWAG↑	WikiText↓
Baseline	18.06	39.71	62.17	23.04	36.70	54.69
CMAR-KLD (Ours)	**18.83** +0.77↑	39.27	65.00	**24.62** +0.58↑	36.77	54.37
CMAR-InfoNCE (Ours)	17.32	39.96	**66.50** +4.33↑	24.34	36.84	54.09
CMAR-CKA (Ours)	18.37	**40.42** +0.71↑	**66.50** +4.33↑	23.52	**36.88** +0.18↑	**53.25**

collections where confounders are controlled. Therefore, we advocate for curated suites that stratify difficulty and annotate confidence/alignment strength.

8.8. Examples of Improved QA Pairs

As the examples below show, CMAR achieves gains in problems related to geometric relationships and spatial transformations. These improvements also extend to chemistry, physics, and multistep mechanistic questions where implicit diagrams or spatial constraints guide the correct answers. However, we hypothesize that the effect is not limited to visual tasks: by grounding language in more coherent spatial and physical priors, vision-aligned models appear to develop a more stable understanding of how the world works. In practice, this manifests as better discrimination between plausible and implausible actions, more consistent causal judgments, and a reduction in purely text-driven shortcuts, which suggests that vision acts not just as an auxiliary modality, but as a scaffold for more general world reasoning.

> *Question:* How do you make pants into shorts?
> A — Cut them vertically in the middle.
> B — Cut them at the knees.
> Correct: B Baseline: A CMAR: **B**.

> *Question:* A small white dog is standing on a table. A woman...
> A — holds out a wooden table and uses a large brush to cut down the woman's hair.
> B — is blow drying her hair.
> C — begins shaving the dog.
> D — is lifting weight up to her chest.
> Correct: C Baseline: B CMAR: **C**.

> *Question:* Points A, B, and C have xy–coordinates (2,0), (8,12), (14,0). Points X, Y, and Z are (6,0), (8,4), (10,0). What fraction c of area $\triangle ABC$ equals area $\triangle XYZ$?
> A — $1/9$ B — $1/8$ C — $1/6$
> D — $1/5$ E — $1/3$
> Correct: A Baseline: E CMAR: **A**.

> *Question:* Identify the number of ^{13}C NMR signals in the final product E from a multistep synthesis (propionaldehyde → A → B → C → D → E).
> A — 6 B — 11 C — 8 D — 3
> Correct: D Baseline: C CMAR: **D**.