# FINE: INFORMATION FLOW-GUIDED INTERVENTION WITH NOISE FOR MITIGATING TEXT DOMINANCE IN MULTIMODAL LARGE LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Multimodal LLMs, typically powered by pretrained LLMs, often suffer from undesirable text dominance, with the non-text modalities underutilized. Most of the existing approaches for modality balancing focus on late-fusion model architectures, with the balancing mechanism applied before the fusion. For the MLLMs adopting early-fusion architectures, information of multimodal inputs flows across modalities over the model layers, making the balancing non-trivial. To this end, we propose an information flow-guided learning framework which alleviates text dominance by injecting noises to the intermediate transformer layers of MLLMs, with reference to the cross-modality information flow. We measure text dominance using conditional mutual information and propose to approximate it using a newly proposed layer-conditional mutual information perturbed difference (LC-MIPD). We show that the approximation gap is characterized by the information flow from visual to textual and the visual information retained given noise injection, which provides a theoretical ground for controlling the noise intervention mechanism. We conduct extensive experiments on Qwen3-VL models across multiple multimodal benchmarks, and demonstrate improvement on overall performance with reduced hallucinations results.

## 1 INTRODUCTION

Multimodal large language models (MLLMs) (Liu et al., 2023; Lin et al., 2024; Bai et al., 2025) often consist of powerful pretrained language models and a lighter non-text encoder. Such a model can lower next-token loss by exploiting correlations in the text while ignoring the non-text input, a behavior known as *text dominance*. Across image, video, audio, time-series, and graph inputs (this paper focuses on images), MLLMs allocate far more attention to text tokens than to non-text tokens (Wu et al., 2025). When the textual context conflicts with the image, vision language models tend to follow the text, and their accuracy drops sharply (Deng et al., 2025). Over-reliance on language priors has also been identified as a main cause of object hallucination (Leng et al., 2024), where a model reports objects that are not in the image (Li et al., 2023). Text dominance is harmful whenever the correct answer depends on the image.

Several recent methods mitigate text dominance in MLLMs. MokA (Wei et al., 2025b), VPRL (Xu et al., 2026) and PostAlign (Wu et al., 2026) change the adapter design, the reasoning format, and the alignment step, respectively, and expect the model to use the image more as a result. None of them defines a quantity for how much the prediction actually depends on the non-text input, so visual reliance cannot be optimized directly. It is also unclear whether an accuracy gain comes from more use of the image or from other factors, such as better exploitation of language priors. Multimodal competition regularization (MCR) (Kontras et al., 2025) quantifies this dependence with the conditional mutual information (CMI) between the non-text input and the prediction given the text, approximated by the mutual information perturbed difference (MIPD) and then by the Jensen-Shannon divergence (JSD). However, MCR assumes that the modalities stay separate until a late fusion step. Early-fusion MLLMs mix visual and text tokens at every layer, so it is unclear where the corruption should be applied. MCR also does not analyze the gaps introduced by the two approximations, so it is unknown whether the JSD objective still tracks the CMI during training.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address these limitations, we propose FINE (information **F**low-guided **I**ntervention with **N**ois**E**), a novel framework that alleviates text dominance in early-fusion MLLM by strategically injecting Gaussian noise into intermediate MLLM layer. FINE introduces layer-conditional MIPD (LC-MIPD) to allow the noise to be injected in the fusion layer of MLLM, and mathematically prove that either the approximation gap between true CMI and LC-MIPD, or between LC-MIPD and JSD. The former gap is determined by the information flow from non-textual to textual feature and noise level, and the latter one is determined by the similarity either between the model output and an unknown distribution, or between the original output and corrupted output. These two decompositions reveal the noise injection position should be before the non-textual information substantial flowing into textual features and the noise level should be proportional to the feature magnitude of MLLM.

We train Qwen3-VL (Bai et al., 2025) with FINE and evaluate on MMBench (Liu et al., 2024), ChartQA (Masry et al., 2022), MMStar (Chen et al., 2024), MME (Fu et al., 2026), and POPE (Li et al., 2023). FINE improves over matched supervised fine-tuning (SFT) on all five benchmarks at every model scale, with the largest gains +2.5% on MMBench, +3.8% on ChartQA, +4.4% on MMStar, +53 on MME, and +2.0% on POPE. We also conduct sensitivity analysis to see how the performance of FINE varies given different noise intervention layers and noise level. To validate the theoretical analysis results, we carry out experiments using synthetic data to confirm the correctness of the derived approximation gaps. We exercise noise intervention to different layers of frozen Qwen3-VL-2B to implicitly detect the non-textual information flowing pattern. We also provide a control experiment by injecting noise to the text feature to show the performance gain is from the increasing the reliance of non-textual information instead of a simple noise regularization. Our contributions are summarized as follows:

- **A flow-guided intervention with noise framework.** We introduce **FINE** (**F**low-Guided **I**ntervention with **N**ois**e**), which formulates text dominance through CMI, introduces LC-MIPD as a surrogate for CMI, and leverages JSD as training objective to approximate the LC-MIPD.
- **Two gap decompositions.** The gap between CMI and LC-MIPD equals the non-text information that has flowed into text states before intervention plus the information that survives the noise, which we bound under Gaussian perturbation. The gap between LC-MIPD and the JSD equals decoder-mismatch and JSD-compression terms that grow with the noise level. Together they give a basis for jointly choosing intervention depth and noise level.
- **Theory-guided empirical validation.** We validate the proposed identities using controlled synthetic distributions, characterize layer-wise non-text information flowing and perturbation sensitivity in pretrained MLLMs, and evaluate FINE across Qwen3-VL variants. We further use matched intervention controls and counterfactual non-text inputs to assess whether performance improvements reflect genuine reliance on changing non-text evidence rather than generic hidden-state sensitivity.

## 2 Related Work

**Text Dominance in MLLMs** Several recent works have identified text dominance in MLLMs and proposed heuristic mitigations. MokA (Wei et al., 2025b) keeps modality-specific low-rank parameters to preserve unimodal information and adds cross-attention to strengthen cross-modal interaction. VPRL (Xu et al., 2026) instead encourages models to reason in a visual space on vision-centric tasks, biasing generation toward visual evidence, and PostAlign (Wu et al., 2026) explicitly promotes grounding in both visual and textual features as a corrective step. These methods act through adapter design, reasoning format, or post-hoc alignment. None has a criterion for how much non-text information the prediction uses, whereas FINE regularizes an information-theoretic surrogate of that quantity.

**Modality Competition and Information-Theoretic Balancing** Huang et al. (2022) prove that in late-fusion networks trained jointly by gradient descent, modality competition makes the encoder favor one dominant modality while the others are never fully learned. Several methods therefore rebalance the optimization signals. On-the-fly gradient modulation (Peng et al., 2022a) and its salience-guided extension (Peng et al., 2022b) adjust modality-specific gradients to speed up learning of weaker modalities, DGL (Wei et al., 2025a) separates fusion-related gradients from modality-

specific ones, and Gao et al. (2025) use CMI to estimate and reinforce underrepresented modalities. All of these assume separate per-modality encoders and gradients, so they do not transfer to early-fusion MLLMs, where a single shared backbone processes all modalities and per-modality gradients are entangled.

MCR (Kontras et al., 2025) is the closest to our work. It characterizes modality competition with CMI, uses MIPD as a tractable surrogate, and approximates MIPD with JSD for stable training. MCR corrupts the non-text modality only before fusion, which implicitly assumes that non-text and text information stay separate until then. This assumption fails in early-fusion MLLMs, where non-text information flows into the text representation across layers. MCR also does not analyze the gap between MIPD and the JSD objective it optimizes. We generalize MIPD to intermediate layers as LC-MIPD, and derive and empirically validate two gap decompositions: between CMI and LC-MIPD, and between LC-MIPD and the JSD training objective.

**Latent Representation Interventions and Multimodal Information Flow** Intervening on latent representations is a standard way to study how networks encode and transmit information. Causal tracing locates components that mediate model behavior (Meng et al., 2022), and activation patching and related methods corrupt or replace intermediate representations to identify the computation behind a prediction (Ghandeharioun et al., 2024). In vision-language models, Assouel et al. (2026) use activation interventions and probing to show that visual information moves into text tokens in middle-to-late layers. These works target causal localization or mechanistic interpretation, while we utilize intervention on latent representations as information-theoretic regularizer for mitigating text dominance in MLLMs.

# 3 Approach

## 3.1 Problem Formulation and MIPD Background

Given a dataset containing 3-tuples of non-textual (e.g., visual) input $X_1$, textual input $X_2$, and a ground-truth next token $Y$ sampled from $p_{\text{data}}(x_1, x_2, y)$,[1] we aim at learning an MLLM parameterized by $\theta$ to give $p_\theta(\hat{y} \mid x_1, x_2)$ for predicting the next token $\hat{Y}$, which can typically be learned by minimizing

$$\mathcal{L}_{\text{pred}}(\theta) = -\mathbb{E}_{p_{\text{data}}(x_1,x_2,y)}[\log p_\theta(y \mid x_1, x_2)]. \tag{1}$$

In the sequel, without loss of generality, we will simply refer the non-textual input as visual. Taking the information-theoretic perspective, the text dominance refers to the situation where the MLLM relies mostly on $X_2$ for predicting $\hat{Y}$, with the visual specific information in $X_1$ underutilized. As in (Kontras et al., 2025), we use conditional mutual information (CMI) to quantify text dominance:

$$\text{CMI} := I(X_1; \hat{Y} \mid X_2) = H(\hat{Y} \mid X_2) - H(\hat{Y} \mid X_1, X_2), \tag{2}$$

where $H()$ denotes entropy. CMI essentially measures the specific information the visual input $X_1$ contributing to the model's next-token prediction $\hat{Y}$ beyond the textual context$X_2$, and can be used for regularizing the model learning.

As direct estimation of CMI (2) is intractable, MCR (Kontras et al., 2025) first makes use of its lower bound *mutual information perturbed difference* (MIPD) as the surrogate:

$$\text{MIPD} := I(X_1; \hat{Y} \mid X_2) - I(\widetilde{X}_1; \hat{Y} \mid X_2) \leq \text{CMI}, \tag{3}$$

where $\widetilde{X}_1$ is a corrupted version of $X_1$. For training stability, it further approximates MIPD with the Jensen-Shannon divergence (JSD) between the output from normal inference (clean prediction) and that patched with latent representations of different samples (corrupted predictions).

$$\text{JSD}\left(p_\theta(\hat{Y} \mid X_1, X_2) \,||\, p_\theta(\hat{Y} \mid \tilde{X}_1, X_2)\right). \tag{4}$$

While the idea can be well applied to late-fusion model architectures, it cannot be directly applied to early-fusion MLLMs where the attention layers entangle visual and textual representations throughout the model, resulting in cross-modal information flow. To ease the subsequent description, we

[1] Uppercase letters denote random variables and lowercase letters their sample.

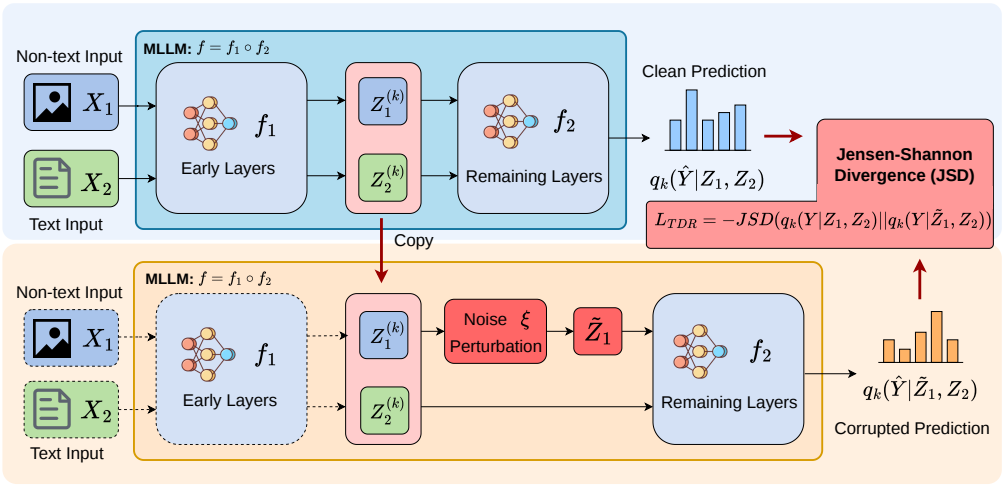


Figure 1: An overview of our proposed Layer-Conditioned MIPD for mitigating text dominance in MLLMs. The top panel illustrates the standard forward pass generating the original prediction distribution $q_k(\hat{Y}|Z_1, Z_2)$. The bottom panel illustrates the CMI estimation stream, where the non-text latent representation $Z_1$ is perturbed with noise to generate $q_k(\hat{Y}|\tilde{Z}_1, Z_2)$. The negative Jensen-Shannon Divergence ($L_{FINE}$) between the two distributions serves as a regularizer to encourage the model to rely more on non-textual representations.

define the portion of the $k^{th}$ layer's latent representation with positions corresponding to those of the visual input as $Z_1^{(k)}$ and the remaining portion corresponding to the textual input as $Z_2^{(k)}$. Instead of corrupting $X_1$, our idea is to corrupt $Z_1^{(k)}$ which is supposed to be corresponding to $X_1$ but also carries information from $X_2$ due to the underlying cross attention mechanism. In the following, we first propose *layer-conditioned mutual information perturbed difference* (LC-MIPD) which makes reference to the cross-modal information flow for the CMI estimation (Section 3.2). In addition, we add Gaussian noise to corrupt $Z_1^{(k)}$, instead of the patching trick used in Kontras et al. (2025), so that the degree of the corruption can be controlled. We then show that the gap of approximating CMI using LC-MIPD increases as the value of $k$ increases (getting deeper into the model). We later on call it the *gap* between CMI and LC-MIPD, and provide a theoretical study to analyze its relationship with the layer where the noise intervention is exercised as well as the noise level (Section 3.3). We conduct an additional theoretical analysis on the *gap* between LC-MIPD and the JSD (Section 3.4), which is not considered in MCR but important as the JSD is eventually used for the regularization.

### 3.2 Layer-Conditioned MIPD

With reference to the intervention layer $k$, we decompose an MLLM into two subnetworks $f_1$ and $f_2$ so that the prediction can be presented as:

$$\hat{Y} \sim q_k\big(\cdot \mid Z_1^{(k)}, Z_2^{(k)}\big), \qquad (Z_1^{(k)}, Z_2^{(k)}) = f_1(X_1, X_2), \tag{5}$$

where $f_1$ will have $k$ layers, $f_2$ will have $L - k$ layers given there are altogether $L$ layers in the MLLM, and $q_k(\cdot \mid Z_1^{(k)}, Z_2^{(k)})$ denotes the output distribution induced by running $f_2$ (together with the model's decoding head) on the layer-$k$ states. We then define the layer-conditioned conditional mutual information (LC-CMI) as:

$$\mathrm{CMI}_k := I(Z_1^{(k)}; \hat{Y} \mid Z_2^{(k)}, X_2), \tag{6}$$

which measures the information contributed by $Z_1^{(k)}$ (positionally aligned with the visual input) to the model's next-token prediction $\hat{Y}$ beyond the textual input $X_2$ and $Z_2^{(k)}$ (aligned with the textual one). To approximate $\mathrm{CMI}_k$, we use layer-conditioned mutual-information perturbed difference (LC-MIPD) as the surrogate defined as:

$$M_k := I(Z_1^{(k)}; \hat{Y} \mid Z_2^{(k)}, X_2) - I(\tilde{Z}_1^{(k)}; \hat{Y} \mid Z_2^{(k)}, X_2), \tag{7}$$

which measures the visual information retained in $Z_1^{(k)}$ given $Z_2^{(k)}$ and the textual input $X_2$ and with independent Gaussian noise added to $\tilde{Z}_1^{(k)}$:

$$\tilde{Z}_1^{(k)} = Z_1^{(k)} + \xi, \qquad \xi \sim \mathcal{N}(0, \sigma^2 I_m), \tag{8}$$

where $m$ is the flattened dimension of $Z_1^{(k)}$, i.e., the number of visual-token positions at layer $k$ times the per-token hidden size.

In addition, we further approximate LC-MIPD by the JSD between the clean and corrupted predictions as the text dominance regularizer. We denote the distribution of the outcome $\hat{Y}$ predicted by $f_2$ with and without the noise intervention respectively as:

$$P_k = q_k(\hat{Y} \mid Z_1^{(k)}, Z_2^{(k)}); \quad R_k = q_k(\hat{Y} \mid \tilde{Z}_1^{(k)}, Z_2^{(k)}). \tag{9}$$

The JSD between the clean and corrupted predictions is used as the text dominance regularizer, and the overall training objective of FINE thus becomes:

$$\mathcal{L}_{\text{FINE}}(\theta) = \mathcal{L}_{\text{pred}}(\theta) - \lambda\, JSD(P_k \parallel R_k), \tag{10}$$

where $\lambda > 0$ is a hyperparameter where a larger $\lambda$ implies favoring less the model if its prediction remains unchanged under visual corruption, thereby encouraging more attention to the visual information. Section 3.4 introduces the *scaled* objective $J_k := 2\,\mathbb{E}[\text{JSD}(P_k, R_k)]$, which is twice the raw JSD term above; Appendix C restates (10) in terms of $J_k$ and the experimental regularization weight $\beta$ used practical implementation.

### 3.3 How Intervention Position and noise level Determine the CMI–MIPD Gap

Before analyzing the gap between CMI and LC-MIPD, we assume the following hidden-state intervention MLLM:

**Assumption 3.1** (Hidden-state intervention MLLM)**.** *(i)* $(Z_1^{(k)}, Z_2^{(k)}) = f_1(X_1, X_2)$ *is deterministic; (ii) the clean output is generated by* $f_2$*, so* $\hat{Y} \perp X_1 \mid (Z_1^{(k)}, Z_2^{(k)}, X_2)$*; (iii) the Gaussian noise in Eq.* (8) *is independent of all clean variables; and (iv) the relevant entropies and divergences are finite, with common support for the decoder output distributions.*

We know the MLLM might have already flowed some visual information into $Z_2$ and no longer be contained in $Z_1$ at layer $k$ and the intervention will not affect these flowed information. To quantify how much information has been flowed into $Z_2$ and how the added Gaussian affect the remaining information in $Z_1$, we define Visual information flowing and corrupted visual information terms:

**Definition 3.2** (Visual information flowing and corrupted visual information terms)**.**

$$\ell_k := I(Z_2^{(k)}; \hat{Y} \mid X_2); \quad \rho_k := I(\tilde{Z}_1^{(k)}; \hat{Y} \mid Z_2^{(k)}, X_2). \tag{11}$$

The Visual information flowing term $\ell_k$ measures how much visual information is flowed from visual inputs $X_1$ to the text latent representations $Z_2$ at the $k$ noise injection position, which is still relevant to the final prediction $\hat{Y}$. The Corrupted visual information term $\rho_k$ measures how much visual information is retained in the noisy visual state $\tilde{Z}_1^{(k)}$ that is still relevant to the final prediction $\hat{Y}$ beyond the text information from the text latent representations $Z_2^{(k)}$ and the text input $X_2$.

**Theorem 3.3** (The gap between CMI and LC-MIPD)**.** *Under Assumption 3.1, for intervention layer* $k \in [0, L]$ *and* $\sigma > 0$*,*

$$\boxed{\text{CMI} - M_k = \ell_k + \rho_k}. \tag{12}$$

*Moreover,* $\ell_k \geq 0$ *and* $\rho_k \geq 0$*, and consequently* $0 \leq M_k \leq \text{CMI}$.

Theorem 3.3 provides an exact decomposition of the gap between the true conditional mutual information (CMI) and the layer-conditioned mutual-information perturbed difference ($M_k$). It demonstrates that $M_k$ underestimates CMI due to two non-negative terms: Visual information flowing $\ell_k$ and corrupted visual information $\rho_k$. During training, we have to minimize these two terms to make sure the optimization following the track of CMI. The visual information flowing term $\ell_k$ is determined by the model architecture and the depth $k$, and we expect this quantity to grow as $k$ increases, which is an expected qualitative trend rather than a claim proven for every $k$. The corrupted visual information term $\rho_k$ is determined by both the depth $k$ and the noise level $\sigma$, and we provide the upper bound of $\rho_k$ below.

**Theorem 3.4** (Corrupted visual information terms bound)**.** *Define* $E_k := \mathbb{E}\|Z_1^{(k)}\|^2$. *If* $E_k < \infty$, *then*

$$\boxed{\rho_k \le \frac{m}{2} \log\left(1 + \frac{E_k}{m\sigma^2}\right)}, \tag{13}$$

Theorem 3.4 shows that, for a fixed intervention layer $k$, increasing the noise level $\sigma$ can only reduce the corrupted visual information term $\rho_k$. Jointly, Theorems 3.3 and 3.4 imply that to keep $M_k$ a faithful surrogate for CMI, the intervention should be placed at a layer where $\ell_k$ is still small (i.e., before substantial visual-to-text flowing has occurred) and the noise level $\sigma$ should be set high enough, relative to the per-element magnitude of the visual state ($\sqrt{E_k/m}$), to keep $\rho_k$ small as well. Please find the full proof of Theorem 3.3 and Theorem 3.4 in Appendix A.

### 3.4 From LC-MIPD to the Computable JSD Objective

Recall that $M_k = H(\hat{Y} \mid Z_2^{(k)}, X_2) - H(\hat{Y} \mid \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2)$. The second entropy term is governed by the unknown *posterior predictive distribution*:

$$Q_k(\hat{y}) := p(\hat{y} \mid \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2) = \int q_k(\hat{y} \mid z_1^{(k)}, Z_2^{(k)})\, p(z_1^{(k)} \mid \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2)\, dz_1^{(k)}, \tag{14}$$

which averages clean predictions $q_k(\hat{y} \mid Z_1^{(k)}, Z_2^{(k)})$ over all possible clean visual sources given the observed perturbation $p(Z_1^{(k)} \mid \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2)$. Because $Q_k$ requires marginalizing over clean states, it is intractable in large MLLMs. In practice, we use the corrupted prediction $R_k$, obtained by passing the noisy latent representation directly through the decoder, in place of $Q_k$. The difference of $R_k$ and $Q_k$ introduces the gap between LC-MIPD and computable JSD objective, and we isolate this discrepancy into two non-negative terms.

**Definition 3.5** (Decoder mismatch and JSD compression terms)**.** *Let* $A_k := \frac{1}{2}(P_k + R_k)$ *denote the mixture of the clean and corrupted predictive distributions defined in* (9)*. We define*

$$C_k := \mathbb{E}\left[D_{\mathrm{KL}}(Q_k \parallel R_k)\right]; \quad S_k := 2\,\mathbb{E}\left[D_{\mathrm{KL}}(A_k \parallel R_k)\right]. \tag{15}$$

The decoder mismatch term $C_k$ measures how far the patched decoder output $R_k$ deviates from the true posterior predictive $Q_k$ that correctly marginalizes over the clean states consistent with the observed perturbation. The JSD compression term $S_k$ measures the divergence between the patched decoder $R_k$ and the mixture distribution $A_k$. Both $C_k$ and $S_k$ are non-negative by the non-negativity of KL divergence. In particular, when no noise is injected ($\sigma = 0$), the corrupted latent representation coincides with the clean one, so $R_k = P_k = Q_k = A_k$ almost surely and both gap terms vanish, and nonzero noise level can make the patched decoder $R_k$ diverge from $Q_k$ and $A_k$, causing $C_k$ and $S_k$ to grow.

**Theorem 3.6** (From LC-MIPD to Decoder-Divergence Objectives)**.** *Under Assumption 3.1, let* $J_k := 2\,\mathbb{E}[\mathrm{JSD}(P_k, R_k)]$ *denote the scaled JSD objective, and let* $C_k, S_k$ *be as in Definition 3.5. Then*

$$\boxed{J_k = M_k + C_k - S_k}. \tag{16}$$

*Consequently, the complete gap between the true conditional mutual information (*CMI*) and the computable training objective (*$J_k$*) is*

$$\mathrm{CMI} - J_k = \ell_k + \rho_k - C_k + S_k. \tag{17}$$

From Theorem 3.6, we know that the visual information flowing term $\ell_k$ is expected to increase with depth $k$ as visual information is flowed into the text modality. At the same time, the corrupted visual information term $\rho_k$ decreases with higher noise $\sigma$. Both the decoder mismatch $C_k$ and the JSD compression term $S_k$ expand as the noise level $\sigma$ increasing. Consequently, deep intervention or heavy noise does not guarantee a tighter surrogate bound, making a calibrated selection B of $(k, \sigma)$ necessary to balance flowing leakage against decoder divergence. Please find the full proof of Theorem 3.6 in Appendix A.4.

| Model | MMBench | ChartQA | MMStar | MME | POPE |
|---|---|---|---|---|---|
| GPT-5 nano | 64.6 | 50.4 | 55.0 | – | – |
| MiniCPM-V-9B | 84.2 | 85.4 | **67.1** | 2380 | **91.2** |
| Mulberry-7B | – | 83.9 | 61.3 | 2396 | – |
| Qwen2-VL-PostAlign-7B | 83.3 | – | – | 1750 | 89.7 |
| Perception-R1-7B | – | 81.6 | 64.5 | 2291 | – |
| DIVA-GRPO-7B | 82.4 | 85.1 | 62.4 | 2316 | 87.3 |
| Qwen3-VL-2B | 76.4 | 79.5 | 55.7 | 2027 | 86.1 |
| Qwen3-VL-2B (SFT) | 76.7 | 79.1 | 55.5 | 2032 | 87.5 |
| Qwen3-VL-2B (MCR) | 76.7 | 75.8 | 55.1 | 2029 | 88.3 |
| Qwen3-VL-2B (FINE) | 77.4 | 82.9 | 56.1 | 2044 | 88.4 |
| Qwen3-VL-4B | 83.3 | 83.6 | 62.5 | 2280 | 86.7 |
| Qwen3-VL-4B (SFT) | 83.3 | 83.6 | 62.1 | 2264 | 87.3 |
| Qwen3-VL-4B (FINE) | 83.8 | 84.0 | 65.7 | 2317 | 89.1 |
| Qwen3-VL-8B | 85.0 | 85.2 | 63.1 | 2362 | 89.1 |
| Qwen3-VL-8B (SFT) | 84.5 | 85.7 | 62.5 | 2374 | 88.4 |
| Qwen3-VL-8B (FINE) | **87.0** | **86.1** | 66.9 | **2403** | 90.4 |

Table 1: Results across various model scales. SFT: the same training without FINE, *MCR*: visual latent representations corrupted by cross-sample permutation, following (Kontras et al., 2025). *FINE*: trained with the proposed intervention. Bold: best per column.

# 4 EXPERIMENTS

We evaluated on MMBench (Liu et al., 2024) for general multimodal understanding, ChartQA (Masry et al., 2022) for chart reasoning, MMStar (Chen et al., 2024) for challenging multimodal reasoning, MME (Fu et al., 2026) for perception and cognition, and POPE (Li et al., 2023) for hallucination evaluation, using the lmms-eval harness (Li* et al., 2024) with greedy decoding. All training use LoRA (Hu et al., 2022) ($r = 64$, $\alpha = 128$) on attention and MLP projections, with the vision encoder and projector frozen, trained for one epoch on the LLaVA-Instruct-150K (Liu et al., 2023) (or subset of it) with AdamW (lr $2 \times 10^{-5}$, cosine schedule). Unless otherwise noted, FINE uses injection depth $k^* = \lfloor 0.5L \rfloor$, relative noise $\sigma_{\text{rel}} = 0.2$, and regularization weight $\beta = 1.0$. Appendices F–K provide detailed protocols.

## 4.1 MAIN RESULTS

We conduct experiments on the Qwen3-VL family (Bai et al., 2025) across three model scales (2B, 4B, and 8B parameters). For each scale, we compare three configurations: the original pretrained model, a matched supervised fine-tuning (SFT) baseline, and the corresponding FINE-trained model. For Qwen3-VL-2B, we additionally compare against an MCR-style baseline (Kontras et al., 2025) adapted to early-fusion architectures by corrupting the intermediate visual representation via cross-sample permutation. We further compare against strong existing MLLMs, including GPT-5 nano, MiniCPM-V (Yu et al., 2026), Mulberry (Yao et al., 2026), Perception-R1 (Xiao et al., 2026), DIVA-GRPO (Gao et al., 2026), and Qwen2-VL-PostAlign (Wu et al., 2026). Table 1 shows that FINE improves all five benchmarks over matched SFT at every model scale, with the largest gains on MMStar ($+0.6$ at 2B, $+3.6$ at 4B, $+4.4$ at 8B) and consistent hallucination reduction on POPE ($+0.9$ to $+2.0$ across scales). As what we expect, training Qwen3-VL-2B with MCR trick introduce performance degradation as permuting the visual features across samples equivalent to completely destroy the visual information, thereby increasing the gap between LC-MIPD and JSD according to theorem 3.6.

## 4.2 RULING OUT GENERIC PERTURBATION SENSITIVITY

The JSD regularizer rewards predictions that change under intervention, but sensitivity alone does not establish visual reliance: a model could become more perturbation-sensitive without using more visual evidence. To test this, we compare FINE with two matched controls and a counterfactual evaluation, all trained on a 40K subset of LLaVA-Instruct-150K (Liu et al., 2023). SFT omits the FINE regularizer. Text-Perturb and Visual-Perturb apply the same Gaussian intervention to a matched

| Method | MMBench | ChartQA | POPE | PC | Resolved | Introduced |
|---|---|---|---|---|---|---|
| SFT | $72.4 \pm 0.7$ | $71.8 \pm 1.1$ | $82.1 \pm 0.5$ | $61.3 \pm 1.4$ | – | – |
| Text-Perturb | $72.9 \pm 0.4$ | $70.9 \pm 0.7$ | $82.8 \pm 0.6$ | $62.4 \pm 0.7$ | $7.2 \pm 0.5$ | $8.1 \pm 0.3$ |
| Visual-Perturb | $74.9 \pm 0.5$ | $72.8 \pm 0.3$ | $84.4 \pm 1.0$ | $67.5 \pm 0.2$ | $13.8 \pm 0.7$ | $5.4 \pm 0.1$ |
| FINE | $77.4 \pm 0.3$ | $74.1 \pm 0.7$ | $87.2 \pm 0.9$ | $68.9 \pm 0.3$ | $14.7 \pm 0.7$ | $5.1 \pm 0.2$ |

Table 2: Perturbation of visual and textual latent representation on Qwen3-VL-2B across MMBench, ChartQA, and POPE. Text-Perturb and Visual-Perturb apply the same perturbation to a matched number of text- or visual-aligned tokens, while FINE perturbs the full visual latent representation.

| $k$ | CMI | $M_k$ | $J_k$ | $\ell_k$ | $\rho_k$ | $C_k$ | $S_k$ | $\lvert\Delta_1\rvert$ | $\lvert\Delta_2\rvert$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.361 | 0.192 | 0.276 | 0.000 | 0.169 | 0.910 | 0.826 | $2 \times 10^{-16}$ | $2 \times 10^{-4}$ |
| 1 | 0.361 | 0.034 | 0.063 | 0.066 | 0.262 | 0.208 | 0.179 | $2 \times 10^{-16}$ | $8 \times 10^{-7}$ |
| 2 | 0.361 | 0.010 | 0.102 | 0.287 | 0.065 | 0.367 | 0.275 | $2 \times 10^{-16}$ | $6 \times 10^{-5}$ |
| 3 | 0.361 | 0.011 | 0.156 | 0.287 | 0.063 | 0.521 | 0.376 | $2 \times 10^{-16}$ | $1 \times 10^{-4}$ |
| 4 | 0.361 | 0.007 | 0.069 | 0.338 | 0.016 | 0.133 | 0.071 | $2 \times 10^{-16}$ | $2 \times 10^{-5}$ |

Table 3: Exact validation of Theorems 3.3 and 3.6 on the synthetic distribution. Validated by sweeping the intervention depth $k \in \{0, 1, 2, 3, 4\}$ on a toy model. $\Delta_1 = \mathrm{CMI} - M_k - \ell_k - \rho_k$ and $\Delta_2 = M_k + C_k - S_k - J_k$ are the residuals of the two theorem identities.

number of sampled prompt-text or visual-token during training. FINE perturbs the complete visual latent representations. Full row-sampling and noise-matching details are in Appendix H.1. We evaluate on MMBench, ChartQA, and POPE, and on a new held-out set of counterfactual image pairs, sharing a question but with different correct answers depending on the image, constructed via LLM-assisted grouping and deterministic filtering (Appendix H.3). We report paired correctness (PC, fraction of pairs correct on both images) and, relative to SFT, the fraction of errors Resolved and the fraction of correct predictions with errors Introduced (Appendix H.4). FINE outperforming both Text-Perturb and Visual-Perturb shows the effect is specific to visual latent representation rather than generic representation perturbation. FINE also achieves the highest paired correctness with more resolved and fewer introduced errors, indicating the regularizer increases genuine reliance on changing visual evidence.

### 4.3 Validation of the Theorems on Synthetic Data

Because the terms $\ell_k$, $\rho_k$, $C_k$, and $S_k$ are intractable to compute exactly on real MLLMs, we first validate the two gap decompositions (Theorems 3.3 and 3.6) where every term can be calculated from a small transformer trained from scratch on a finite-support synthetic distribution (full setup in Appendix D). Using a the trained model, we sweep the intervention depth $k \in \{0, 1, 2, 3, 4\}$ and calculate these theoretical identities. Table 3 demonstrates both gaps decomposition hold with both residuals $|\Delta_1| = \mathrm{CMI} - M_k - \ell_k - \rho_k$ and $|\Delta_2| = M_k + C_k - S_k - J_k$ less than $2 \times 10^{-4}$, giving us confidence in using the same decomposition to interpret behavior on real MLLMs in other experiments.

### 4.4 Where and How Much to Intervene

Directly estimating $\ell_k$ requires the true data distribution and is intractable in practice. Following Assouel et al. (2026), we instead perturb the frozen model's visual latent representations at varying depths and inspect the resulting KL-divergence between clean and corrupted predictions: a sudden decrease in divergence indicates that corrupting the representation at that depth no longer affects the prediction much, signaling that visual information has already flowed into the text state. We study this pattern for frozen Qwen3-VL-2B on MMBench, ChartQA, and GQA, across injection depths $k \in \{4, 6, 10, 14, 18, 22, 26\}$ and noise levels $\sigma_{\mathrm{rel}} \in \{0.1, 0.2, 0.5, 1.0, 4.0\}$ (Appendix E). As shown in Figure 2, Qwen3-VL-2B begins actively flowing visual information around depth 18, consistent with Assouel et al. (2026)'s finding that MLLMs flow visual information into text representations mainly in their late layers. This motivates our default injection depth $k^* = 14$ for Qwen3-VL-2B, placed before substantial flow while balancing training overhead.

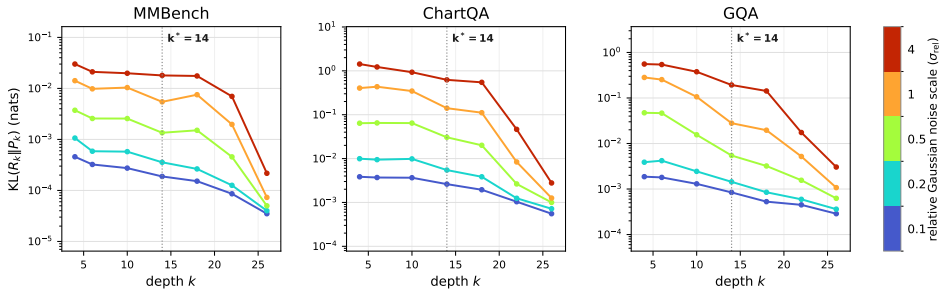


Figure 2: Visual latent representation sensitivity on frozen Qwen3-VL-2B among benchmarks: MMBench, ChartQA, and GQA at $k \in \{4, 6, 10, 14, 18, 22, 26\}$ and $\sigma_{\text{rel}} \in \{0.1, 0.2, 0.5, 1.0, 4.0\}$. Dotted lines $k^*$ mark the default noise injection depths for main experiments.

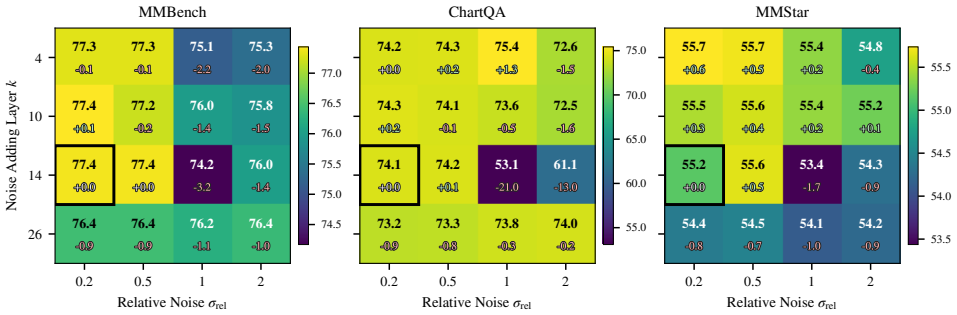


Figure 3: Joint depth–noise sensitivity across benchmarks MMBench, ChartQA, and MMStar. Each grid cell shows the absolute score along with the relative performance change (subscript) compared to the default settings (i.e., $k = 14$ & $\sigma_{rel} = 0.2$). Mild to moderate noise regularizations ($\sigma_{\text{rel}} \leq 0.5$) preserve performance, whereas stronger perturbations at deeper layers ($k \geq 14, \sigma_{\text{rel}} \geq 1.0$) lead to noticeable degradation.

We next test whether this depth–noise trade-off, derived from the theory and located above in a frozen model, also governs downstream task performance once the model is trained with FINE. We train Qwen3-VL-2B with FINE across $k \in \{4, 10, 14, 26\}$ and $\sigma_{\text{rel}} \in \{0.2, 0.5, 1, 2.0\}$ and evaluate on MMBench, ChartQA, and MMStar (Appendix G). Figure 3 shows that mild-to-moderate noise ($\sigma_{\text{rel}} \leq 0.5$) preserves accuracy at every depth tested, whereas combining deep intervention ($k \geq 14$) with strong noise ($\sigma_{\text{rel}} \geq 1$) causes marked degradation, most severely on ChartQA ($-21.0$ points at $k = 14$, $\sigma_{\text{rel}} = 1$). This matches the theoretical prediction that neither depth nor noise level alone controls the CMI–MIPD or MIPD–JSD gap, so the calibration rule of Section B is needed to avoid corrupting task-relevant visual information. Table 8 in Appendix additionally reports a sensitivity analysis over the regularization weight $\beta$ on Qwen3-VL-2B.

## 5 Conclusion

We introduced FINE to address text dominance in early-fusion MLLMs by injecting Gaussian noise to the intermediate layers of MLLMs, guided by the cross-modal information flow. Motivated by the use of conditional mutual information (CMI) for text dominance regularization, we proposed to approximate it using a newly proposed layer-conditional mutual information perturbed difference (LC-MIPD). We show that the approximation gap is characterized by the information flow from visual to textual and the visual information retained given noise injection, which provides a theoretical ground for controlling the noise intervention mechanism. In particular, injecting stronger noise to a deeper layer does not guarantee a tighter bound of the approximation gap. For performance evaluation, across Qwen3-VL-2B/4B/8B, FINE improves accuracy on five benchmark relative to matched SFT. Matched row-level controls and a counterfactual evaluation show these gains reflect genuine reliance on visual evidence rather than generic perturbation sensitivity. Synthetic experiments with exact computation confirm the theoretical analysis of the approximation gap within numerical tolerance. Finally, profiling the frozen model locates where visual information flows into text states, and a joint depth–noise sweep on trained models shows performance degrades where the theory predicts the surrogate gap should widen.

### AI USE STATEMENT

In this work, we used generative AI tools for paragraph polishing and programming. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by one of the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Rim Assouel, Declan Iain Campbell, and Taylor Whittington Webb. Visual symbolic mechanisms: Emergent symbol processing in vision language models. In *First Workshop on CogInterp: Interpreting Cognition in Deep Learning Models*, 2026. URL https://openreview.net/forum?id=eHEoujt4BF.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *ArXiv*, abs/1308.3432, 2013. URL https://api.semanticscholar.org/CorpusID:18406556.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024.

Ailin Deng, Tri Cao, Zhirui Chen, and Bryan Hooi. Words or vision: Do vision-language models have blind faith in text? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3867–3876, 2025.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=DgH9YCsqWm.

Haowen Gao, zhenyu zhang, Liang Pang, Fangda Guo, douhongjian, Guannan Lv, ShaoGuo Liu, Tingting Gao, Huawei Shen, and Xueqi Cheng. DIVA-GRPO: Enhancing multimodal reasoning through difficulty-adaptive variant advantage. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qKXYEg00eH.

Xiyuan Gao, Bing Cao, Pengfei Zhu, Nannan Wang, and Qinghua Hu. Asymmetric reinforcing against multi-modal representation bias. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i16.33841. URL https://doi.org/10.1609/aaai.v39i16.33841.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15466–15490. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ghandeharioun24a.html.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, volume 1, pp. 3, 2022.

Yu Huang, Junyang Lin, Chang Zhou, Hongxia Yang, and Longbo Huang. Modality competition: What makes joint training of multi-modal network fail in deep learning? (Provably). In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9226–9259. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/huang22e.html.

Konstantinos Kontras, Thomas Strypsteen, Christos Chatzichristos, Paul Pu Liang, Matthew B. Blaschko, and Maarten De Vos. Balancing multimodal training through game-theoretic regularization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f5784de266e45bd09333721b0a602b61-Abstract-Conference.html.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13872–13882, 2024.

Bo Li*, Peiyuan Zhang*, Kaichen Zhang*, Fanyi Pu*, Xinrun Du, Yuhao Dong, Haotian Liu, Yuanhan Zhang, Ge Zhang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Accelerating the development of large multimoal models, March 2024. URL https://github.com/EvolvingLMMs-Lab/lmms-eval.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 5971–5984, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, pp. 34892–34916, 2023.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. MMBench: Is your multi-modal model an all-around player? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 216–233. Springer, 2024.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177/.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=-h6WAS6eE4.

Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. Balanced multimodal learning via on-the-fly gradient modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022a.

Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. OGM-GE: Salience guided dynamic curriculum learning for modality-aware balanced multimodal learning. In *European Conference on Computer Vision*, 2022b.

Shicai Wei, Chunbo Luo, and Yang Luo. Boosting multimodal learning via disentangled gradient learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22879–22888, 2025a.

Yake Wei, Yu Miao, Dongzhan Zhou, and Di Hu. Moka: Multimodal low-rank adaptation for MLLMs. In *Advances in Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=oJ84bedrtM.

Huyu Wu, Meng Tang, Xinhan Zheng, and Haiyun Jiang. When language overrules: Revealing text dominance in multimodal large language models. *arXiv preprint arXiv:2508.10552*, 2025.

Yixuan Wu, Yang Zhang, Jian Wu, Philip Torr, and Jindong Gu. Postalign: Multimodal grounding as a corrective lens for MLLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zJnPyb2xrp.

Tong Xiao, Xin Xu, Zhenya Huang, Hongyu Gao, Quan Liu, Qi Liu, and Enhong Chen. Perception-r1: Advancing multimodal reasoning capabilities of MLLMs via visual perception reward. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=KttCXdjj4w.

Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. Visual planning: Let's think only with images. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wsnse46kRO.

Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, YuXin Song, Haocheng Feng, Li Shen, and Dacheng Tao. Mulberry: Empowering MLLM with o1-like reasoning and reflection via collective monte carlo tree search. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=lwOV2ACEK9.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Ranchi Zhao, Bokai Xu, Junbo Cui, Yingjing Xu, Liqing Ruan, Luoyuan Zhang, Hanyu Liu, Jingkun Tang, Hongyuan Liu, Qining Guo, Wenhao Hu, Bingxiang He, Jie Zhou, Jie Cai, Ji Qi, Zonghao Guo, Chi Chen, Guoyang Zeng, Yuxuan Li, Ganqu Cui, Ning Ding, Xu Han, Yuan Yao, Zhiyuan Liu, and Maosong Sun. MiniCPM-v 4.5: Cooking efficient MLLMs via architecture, data, and training recipe. In *Conference on Computer Vision and Pattern Recognition 2026*, 2026. URL https://openreview.net/forum?id=di0Twx0AnS.

# A Proofs of the Main Results

## A.1 Proof of Theorem 3.3

*Proof.* Fix $k$ and use the states $Z_1^{(k)}, Z_2^{(k)}, \tilde{Z}_1^{(k)}$ defined in the main text. We first establish the sufficient-state identity used in the main text. By Assumption 3.1(ii),

$$H(\hat{Y} \mid X_1, X_2, Z_1^{(k)}, Z_2^{(k)}) = H(\hat{Y} \mid Z_1^{(k)}, Z_2^{(k)}, X_2). \tag{18}$$

Moreover, because $(Z_1^{(k)}, Z_2^{(k)})$ is deterministic given $(X_1, X_2)$, $H(\hat{Y} \mid X_1, X_2, Z_1^{(k)}, Z_2^{(k)}) = H(\hat{Y} \mid X_1, X_2)$. Combining this fact with (18) gives

$$H(\hat{Y} \mid X_1, X_2) = H(\hat{Y} \mid Z_1^{(k)}, Z_2^{(k)}, X_2). \tag{19}$$

Therefore

$$\begin{aligned} I(X_1; \hat{Y} \mid X_2) &= H(\hat{Y} \mid X_2) - H(\hat{Y} \mid X_1, X_2) && (20)\\ &= H(\hat{Y} \mid X_2) - H(\hat{Y} \mid Z_1^{(k)}, Z_2^{(k)}, X_2) && (21)\\ &= I(Z_1^{(k)}, Z_2^{(k)}; \hat{Y} \mid X_2) && (22)\\ &= I(Z_2^{(k)}; \hat{Y} \mid X_2) + I(Z_1^{(k)}; \hat{Y} \mid Z_2^{(k)}, X_2), && (23) \end{aligned}$$

where the last line is the conditional mutual-information chain rule. By definition,

$$I(Z_1^{(k)}; \hat{Y} \mid Z_2^{(k)}, X_2) = M_k + I(\tilde{Z}_1^{(k)}; \hat{Y} \mid Z_2^{(k)}, X_2). \tag{24}$$

Substituting (24) into (23) yields

$$I(X_1; \hat{Y} \mid X_2) - M_k = \ell_k + \rho_k. \tag{25}$$

It remains to prove the sign claims. Mutual information is nonnegative, so $\ell_k \geq 0$ and $\rho_k \geq 0$. For $M_k$, Assumption 3.1(iii) implies

$$p(\hat{y}, \tilde{z}_1 \mid z_1, z_2, x_2) = p(\hat{y} \mid z_1, z_2) p(\tilde{z}_1 \mid z_1), \tag{26}$$

which is precisely the conditional Markov chain $\hat{Y} - Z_1^{(k)} - \tilde{Z}_1^{(k)}$ given $(Z_2^{(k)}, X_2)$. Conditional data processing then gives

$$I(\tilde{Z}_1^{(k)}; \hat{Y} \mid Z_2^{(k)}, X_2) \leq I(Z_1^{(k)}; \hat{Y} \mid Z_2^{(k)}, X_2), \tag{27}$$

and hence $M_k \geq 0$. Finally, the exact decomposition and nonnegative gap terms imply $M_k \leq I(X_1; \hat{Y} \mid X_2)$. □

## A.2 Proof of Theorem 3.4

*Proof.* Fix $k$. Independent noise gives the conditional Markov chain $\hat{Y} - Z_1^{(k)} - \tilde{Z}_1^{(k)}$ given $(Z_2^{(k)}, X_2)$. Conditional data processing gives

$$\rho_k = I(\tilde{Z}_1^{(k)}; \hat{Y} \mid Z_2^{(k)}, X_2) \leq I(\tilde{Z}_1^{(k)}; Z_1^{(k)} \mid Z_2^{(k)}, X_2). \tag{28}$$

Expanding the right-hand side and using $\tilde{Z}_1^{(k)} - Z_1^{(k)} = \xi$ gives

$$\begin{aligned} I(\tilde{Z}_1^{(k)}; Z_1^{(k)} \mid Z_2^{(k)}, X_2) &= h(\tilde{Z}_1^{(k)} \mid Z_2^{(k)}, X_2) - h(\tilde{Z}_1^{(k)} \mid Z_1^{(k)}, Z_2^{(k)}, X_2) && (29)\\ &= h(\tilde{Z}_1^{(k)} \mid Z_2^{(k)}, X_2) - h(\xi \mid Z_1^{(k)}, Z_2^{(k)}, X_2) && (30)\\ &= h(\tilde{Z}_1^{(k)} \mid Z_2^{(k)}, X_2) - h(\xi) && (31)\\ &\leq h(\tilde{Z}_1^{(k)}) - h(\xi). && (32) \end{aligned}$$

The third equality uses $\xi \perp (Z_1^{(k)}, Z_2^{(k)}, X_2)$, and the inequality is conditioning reduces entropy. Since $\mathbb{E}\|\tilde{Z}_1^{(k)}\|^2 = E_k + m\sigma^2$, we have $\operatorname{tr}\operatorname{Cov}(\tilde{Z}_1^{(k)}) \leq E_k + m\sigma^2$. Among distributions with

a fixed covariance, the Gaussian maximizes differential entropy; the arithmetic–geometric mean inequality further yields

$$h(\tilde{Z}_1^{(k)}) \le \frac{1}{2}\log\Big((2\pi e)^m \det \mathrm{Cov}(\tilde{Z}_1^{(k)})\Big) \tag{33}$$

$$\le \frac{m}{2}\log\left(2\pi e\left(\sigma^2 + \frac{E_k}{m}\right)\right), \tag{34}$$

$$h(\xi) = \frac{m}{2}\log(2\pi e\sigma^2). \tag{35}$$

Their difference is

$$\frac{m}{2}\log\left(1 + \frac{E_k}{m\sigma^2}\right), \tag{36}$$

which proves (13). The logarithm converges to zero as $\sigma \to \infty$, so only the visual information flowing term remains in the total-gap bound. For $0 < \sigma_1 < \sigma_2$, couple observations as $\tilde{Z}_1 = Z_1^{(k)} + \sigma_1\epsilon_1$ and $\tilde{Z}_2 = \tilde{Z}_1 + \sqrt{\sigma_2^2 - \sigma_1^2}\epsilon_2$, with independent standard Gaussian vectors independent of the clean variables. Conditional on $(Z_2^{(k)}, X_2)$, $\hat{Y} \to \tilde{Z}_1 \to \tilde{Z}_2$ is a Markov chain. Data processing gives $\rho_k(\sigma_2) \le \rho_k(\sigma_1)$, and $M_k = \mathrm{CMI} - \ell_k - \rho_k$ gives the claimed monotonicity of $M_k$. □

**Corollary A.1** (A sufficient fidelity certificate)**.** *For a tolerance $\tau \ge 0$, any admissible depth and noise level satisfying*

$$\ell_k + \frac{m}{2}\log\left(1 + \frac{E_k}{m\sigma^2}\right) \le \tau \tag{37}$$

*also satisfy* $\mathrm{CMI} - M_k \le \tau$.

It is a sufficient, not necessary, condition for latent MIPD fidelity. The power-based bound can be loose in large hidden spaces, and applying the condition requires knowledge of the visual information flowing term.

### A.3 Derivation of the Midpoint Identity

*Proof.* Let $A = (P + R)/2$. Using the common support assumption and expanding every KL pointwise,

$$D_{\mathrm{KL}}(P\|R) - 2\mathrm{JSD}(P,R) = D_{\mathrm{KL}}(P\|R) - D_{\mathrm{KL}}(P\|A) - D_{\mathrm{KL}}(R\|A) \tag{38}$$

$$= \sum_y P(y)\log\frac{A(y)}{R(y)} + \sum_y R(y)\log\frac{A(y)}{R(y)} \tag{39}$$

$$= \sum_y (P(y) + R(y))\log\frac{A(y)}{R(y)} \tag{40}$$

$$= 2D_{\mathrm{KL}}(A\|R). \tag{41}$$

The last equality uses $P + R = 2A$. Rearrangement proves the identity, and $D_{\mathrm{KL}}(A\|R) \ge 0$ proves the upper bound. For continuous outputs the same calculation holds with sums replaced by integrals. □

### A.4 Proof of Theorem 3.6

**Expectation laws.** Let $\mu_k$ be the joint law of $(Z_1^{(k)}, \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2)$ induced by the data input distribution, the deterministic prefix, and independent Gaussian noise. Let $\nu_k$ be its marginal on $(\tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2)$. Including the clean output gives

$$\pi_k(dz_1, d\tilde{z}_1, dz_2, dx_2, \hat{y}) = \mu_k(dz_1, d\tilde{z}_1, dz_2, dx_2)q_k(\hat{y} \mid z_1, z_2). \tag{42}$$

Thus $K_k, J_k, S_k$ average under $\mu_k$, $C_k$ averages under $\nu_k$, and the entropies and mutual informations use $\pi_k$. The posterior in $Q_k$ is $\mu_k(dz_1 \mid \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2)$. For token-level training, these laws are evaluated separately at each fixed textual context (including its history) before averaging over the chosen context distribution.

*Proof.* **Step 1: Explain what the noisy observation predicts.** Fix the depth $k$. Draw a clean example, compute its latent representations, and add independent noise to the visual state. The output $\hat{Y}$ is drawn from the *clean* decoder $P_k$ throughout this proof, including when we measure its information shared with the noisy state. We use the laws $\mu_k$, $\nu_k$, and $\pi_k$ defined in (42); all entropies and mutual informations below are computed under $\pi_k$.

If we know the clean visual state, the output distribution is $P_k$. If we know only its noisy version, we must average the clean predictions over the possible original states. The law of total probability gives

$$\begin{aligned} &p(\hat{Y} = y \mid \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2) \\ &\quad = \mathbb{E}_{Z_1^{(k)} \sim \mu_k(\cdot \mid \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2)} \left[ q_k(y \mid Z_1^{(k)}, Z_2^{(k)}) \,\middle|\, \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2 \right] \\ &\quad = Q_k(y). \end{aligned} \tag{43}$$

Independent noise justifies using the clean decoder inside this average: once the clean states are known, the noisy state supplies no further information about the clean output.

The posterior average $Q_k$ and the patched decoder $R_k$ need not agree. For example, if two possible clean states are equally likely given the noisy observation and assign an answer probabilities $0.9$ and $0.3$, then $Q_k$ assigns that answer probability $0.6$. Running the decoder directly on the noisy state need not give $R_k(y) = 0.6$.

**Step 2: Expand MIPD as an increase in uncertainty.** Conditional entropy measures the uncertainty remaining about an output after the conditioning variables are known. Expanding the two mutual informations in the definition of $M_k$ gives

$$\begin{aligned} M_k &= \left[ H(\hat{Y} \mid Z_2^{(k)}, X_2) - H(\hat{Y} \mid Z_1^{(k)}, Z_2^{(k)}, X_2) \right] \\ &\quad - \left[ H(\hat{Y} \mid Z_2^{(k)}, X_2) - H(\hat{Y} \mid \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2) \right] && (44) \\ &= H(\hat{Y} \mid \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2) - H(\hat{Y} \mid Z_1^{(k)}, Z_2^{(k)}, X_2). && (45) \end{aligned}$$

The common baseline $H(\hat{Y} \mid Z_2^{(k)}, X_2)$ cancels. What remains is the extra uncertainty from observing the noisy visual state instead of the clean one.

Because $P_k$ and $Q_k$ are precisely these two conditional output distributions, the same difference can be written as

$$\begin{aligned} \mathbb{E}_{\pi_k} \left[ \log \frac{P_k(\hat{Y})}{Q_k(\hat{Y})} \right] &= -H(\hat{Y} \mid Z_1^{(k)}, Z_2^{(k)}, X_2) \\ &\quad + H(\hat{Y} \mid \tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2) && (46) \\ &= M_k. && (47) \end{aligned}$$

Here we used the definition of conditional entropy as the average negative log probability of the output.

**Step 3: Connect the computable KL to MIPD.** The computable KL compares $P_k$ with $R_k$. Insert $Q_k$ into its log ratio to separate information loss from decoder mismatch:

$$\begin{aligned} K_k &= \mathbb{E}_{\mu_k}[D_{\mathrm{KL}}(P_k \| R_k)] && (48) \\ &= \mathbb{E}_{\pi_k} \left[ \log \frac{P_k(\hat{Y})}{R_k(\hat{Y})} \right] && (49) \\ &= \mathbb{E}_{\pi_k} \left[ \log \frac{P_k(\hat{Y})}{Q_k(\hat{Y})} \right] + \mathbb{E}_{\pi_k} \left[ \log \frac{Q_k(\hat{Y})}{R_k(\hat{Y})} \right]. && (50) \end{aligned}$$

The first term equals $M_k$ by Step 2. For the second, hold the noisy visual state, text state, and text input fixed. The clean output then follows $Q_k$, so averaging its log ratio gives

$$\sum_y Q_k(y) \log \frac{Q_k(y)}{R_k(y)} = D_{\mathrm{KL}}(Q_k \| R_k). \tag{51}$$

Averaging also over $(\tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2) \sim \nu_k$, the tower property gives

$$\mathbb{E}_{\pi_k}\left[\log \frac{Q_k(\hat{Y})}{R_k(\hat{Y})}\right] = \mathbb{E}_{\nu_k}\left[\mathbb{E}_{\hat{Y}\sim Q_k}\left[\log \frac{Q_k(\hat{Y})}{R_k(\hat{Y})}\right]\right] \tag{52}$$

$$= \mathbb{E}_{\nu_k}[D_{\mathrm{KL}}(Q_k \| R_k)] = C_k. \tag{53}$$

Here the inner expectation holds $(\tilde{Z}_1^{(k)}, Z_2^{(k)}, X_2)$ fixed. Therefore

$$\boxed{K_k = M_k + C_k.} \tag{54}$$

Thus the observed KL includes both the information lost through corruption and the patched decoder's mismatch with the posterior average. This equality requires averaging over the possible clean states; it is not a pointwise additive rule for KL.

**Step 4: Account for replacing KL by JSD.** Use the same midpoint $A_k = (P_k + R_k)/2$ as in the main text. Appendix A.3 shows, for every clean/noisy pair, that

$$2\mathrm{JSD}(P_k, R_k) = D_{\mathrm{KL}}(P_k \| R_k) - 2D_{\mathrm{KL}}(A_k \| R_k). \tag{55}$$

Taking expectations under $\mu_k$, the left side is $J_k = 2\mathbb{E}_{\mu_k}[\mathrm{JSD}(P_k, R_k)]$, while the subtracted term becomes $S_k$ by (55). Substituting Step 3,

$$J_k = K_k - S_k \tag{56}$$

$$= \boxed{M_k + C_k - S_k.} \tag{57}$$

These are exact identities under the theorem's assumptions. □

## B Noise-Scale Calibration

The corruption in (8) is specified by a single scalar $\sigma$, but the bound in (13) depends on it only through the per-element signal-to-noise ratio $E_k/(m\sigma^2)$. Because the per-element scale $E_k$ grows substantially with depth, an absolute $\sigma$ would place different layers into entirely different noise regimes, confounding depth with perturbation magnitude.

To maintain a consistent perturbation impact across layers, we scale the noise variance by the per-element RMS of the clean visual state:

$$s_k := \sqrt{E_k/m}, \qquad \sigma_k := \sigma_{\mathrm{rel}} \cdot s_k. \tag{58}$$

Here $s_k$ is the per-element RMS of the clean visual state at depth $k$, and the corruption becomes $\xi \sim \mathcal{N}(0, \sigma_k^2 I_m)$. Theorem 3.4 then simplifies to:

$$\rho_k \le \frac{m}{2} \log\left(1 + {\sigma_{\mathrm{rel}}}^{-2}\right), \tag{59}$$

making $\sigma_{\mathrm{rel}}$ a universal control parameter independent of depth: $\sigma_{\mathrm{rel}} = 1$ equates noise and signal variance, $\sigma_{\mathrm{rel}} \to 0$ recovers the clean pass, and $\sigma_{\mathrm{rel}} = 4$ serves as a stress test. Estimating $E_k$ once on a calibration split allows depth-noise trade-offs to be evaluated on a uniform axis, as detailed in Appendix I.

## C Relating the Loss Weight $\lambda$ to the Experimental Weight $\beta$

**Remark C.1** (Relating the loss weight $\lambda$ to the experimental weight $\beta$)**.** *Theorem 3.6 defines $J_k := 2\,\mathbb{E}[\mathrm{JSD}(P_k, R_k)]$, i.e., $J_k$ is exactly twice the raw JSD term appearing in* (10)*. Substituting this identity into* (10) *and matching coefficients of $\mathbb{E}[\mathrm{JSD}(P_k, R_k)]$ gives an equivalent, $J_k$-based form of the training objective,*

$$\mathcal{L}_{\mathrm{FINE}}(\theta) = \mathcal{L}_{\mathrm{pred}}(\theta) - \beta\, J_k, \qquad \textit{with} \qquad \lambda = 2\beta, \tag{60}$$

*where the identity $\lambda = 2\beta$ follows directly from equating $\lambda\,\mathrm{JSD}(P_k, R_k)$ in* (10) *with $\beta\, J_k = 2\beta\,\mathrm{JSD}(P_k, R_k)$ in* (60) *(see also the proof of Theorem 3.6 above, where $J_k$ is derived from the raw KL/JSD quantities). We use $\lambda$ when stating the objective abstractly in terms of the unscaled JSD (Section 3.2), and $\beta$ when reporting the regularization strength actually used in training, since $\beta$ is the coefficient on $J_k$, the quantity that is directly computed and backpropagated. The two are related exactly by $\lambda = 2\beta$; e.g., the default $\beta = 1.0$ used in Table 6 corresponds to $\lambda = 2.0$ in* (10)*.*

## D SYNTHETIC VALIDATION DETAILS

We trains the toy model from scratch and calculate the identities of theorem 3.3 and 3.6 on synthetic distribution. The paragraphs below record the data, model, training, and verification context.

**Data generation.** Discrete factors $v, t \in \{1, \dots, 10\}$; images $x_1 = \mathrm{onehot}(v)$ and text $x_2 = \mathrm{onehot}(t)$ are noiseless, so every latent representation at every depth takes at most 100 distinct values and all clean information-theoretic quantities are finite sums. The distribution is $y = (v + t) \bmod 10$. Training samples $(v, t)$ uniformly from the full support of 100 pairs (batch size 256). A single training seed is used; the 2,048 Monte Carlo draws used for noisy terms at each depth in Table 3 are reseeded per-depth.

**Toy model.** A softmax-attention transformer with $L = 4$ layers, $d = 64$, 4 heads, $d_{ff} = 128$, learned input embeddings, pre-normalized blocks, and a final layer normalization. A rank-one straight-through quantized bottleneck (Bengio et al., 2013) (five values, $-1$ to 1 in steps of $0.5$) is applied to both modality rows after each block. The text row additionally retains an input-text anchor, so it can solve the text-sufficient control while its quantized component can only carry a limited amount of flowed visual information. The classifier reads both final rows. This bottleneck is necessary rather than incidental: without it (a no-bottleneck variant), the text state becomes injective in $v$ after the first layer and $\ell_k, \rho_k$ both trivialize, which is why the capacity-limited design is used for the reported run.

**Training context.** The model is trained with AdamW (learning rate $10^{-3}$, weight decay 0, batch size 256, at most 6K steps) and early stopping on the exact full-support NLL, validated every 500 steps with patience 1500. The FINE regularizer is applied at $k = 0$, $\sigma_{\text{rel}} = 4$, $\beta = 1$, $n_\xi = 1$ with the patched branch retaining $Z_2^{(k)}$ and $X_2$ for conditioning and corrupting only $Z_1^{(k)}$. This training run is the single checkpoint used for every row of Table 1; the depth sweep $k \in \{0, \dots, 4\}$ is an offline evaluation on this one model, not five separately trained models.

**Exact computation.** For each $(v, t)$ pair, a forward pass gives the clean model distribution $p_\theta(\hat{y} \mid x_1, x_2)$. Marginalizing over $x_1$ gives $p(\hat{y} \mid x_2)$ and hence $H(\hat{Y} \mid X_2)$ by exact enumeration; CMI, $\ell_k$ (via $\mathrm{CMI}_k$), and the remaining clean entropies follow directly from their definitions by grouping the finite support according to the relevant conditioning states. For noisy quantities ($M_k$, $\rho_k$, $C_k$, $S_k$, $J_k$), we enumerate the possible clean source states to form the posterior mixture $Q_k$ and approximate the remaining expectation over the Gaussian noise $\xi$ with 2,048 common Monte Carlo draws, reused across depths where the underlying random stream permits paired comparison.

Table 4: synthetic configuration.

| Setting | Value |
|---|---|
| Model | 4 layers, $d = 64$, 4 heads, $d_{ff} = 128$, pre-normalized blocks, final layer norm |
| Bottleneck | rank-one straight-through quantizer, five level $\{-1, -0.5, 0, 0.5, 1\}$, both rows after each block; text row keeps an input-text anchor; classifier reads both rows |
| Data | $v, t \in \{1, \dots, 10\}$, 100 pairs sampled uniformly, batch size 256; main split $y = (v + t) \bmod 10$ |
| Training | AdamW, lr $10^{-3}$, weight decay 0, $\leq$6K steps, early stopping (patience 1500, exact full-support NLL every 500 steps), single seed, CPU |
| FINE | $k = 0$, $\sigma_{\text{rel}} = 4$, $\beta = 1$, $n_\xi = 1$, additive Gaussian noise on visual rows |
| Evaluation | exact enumeration over the 100 pairs; 2,048 common draws for noisy terms; depth sweep $\{0, 1, 2, 3, 4\}$; identity tolerance $10^{-4}$ nats |

## E FROZEN-MODEL PROFILE DETAILS

This part measures frozen models and updates no weights, so its protocol consists of sample construction, a fixed noise calibration, and a measurement rule; Table 5 collects the configuration.

Table 5: Frozen-model profiling configuration.

| Setting | Value |
|---|---|
| Models | frozen Qwen3-VL-2B ($L = 28$), bfloat16, FlashAttention-2 |
| Hardware | one RTX4090-24GB (Appendix K) |
| Sample | 96 examples per family MMBench, ChartQA, and GQA |
| Depth grid | $\{4, 6, 10, 14, 18, 22, 26\}$; reference $k^* = \lfloor 0.5L \rfloor = 14$ |
| Noise | $\sigma_{\text{rel}} \in \{0, 0.1, 0.2, 0.5, 1.0, 4.0\}$; frozen per-stratum scale $s_{k,g} = \sqrt{E_{k,g}/m_g}$ with $\sigma_{k,g} = \sigma_{\text{rel}} s_{k,g}$ fixed before profiling |
| Measurement | teacher forcing; full-vocabulary KL and reference-answer log-probability in float32 at active answer positions; 16 paired draws per nonzero cell; maximum sequence length 768; batch size 8 |
| Uncertainty | 10,000 bootstrap resamples over source-image groups within families; single model seed (0) |

## F Main Training Details

The main experiments trains one FINE adapter and one matched baseline per model scale and evaluates both on the suite of Appendix J. Table 6 collects the shared training and FINE configuration; the paragraphs below record the model, data, and evaluation context.

**Perturbation module.** Visual positions are identified by the image-patch token id 151655, and the patched branch is computed on the second half of a doubled batch with the early-layer activations reused, so only the tail $q_k$ is re-run (Section 3). The FINE objective is the scaled JSD in (10) with additive Gaussian noise (per-depth relative scale $\sigma_k = \sigma_{\text{rel}}\sqrt{E_k/m}$) and configurable $\beta$, $n_\xi$, and text-row control. The one-sided KL $K_k$ is an optional offline measurement and is not a training objective or controller.

**Models and training settings** The comparison covers Qwen3-VL-2B/4B/8B-Instruct (Bai et al., 2025), and the main experiment is compare the three scales. LoRA (Hu et al., 2022) is applied to the attention and MLP projections ($q, k, v, o$ and gate/up/down) with rank $r = 64$, $\alpha = 128$, and dropout $0.0$; the vision encoder and the projector are frozen, and all FINE-augmented models are trained with identical hyperparameters for fair comparison. We use a LLaVA-instruction-150K mixture (Liu et al., 2023) for the main experiments, containing about 150K examples. Images are preprocessed with each model's processor at dynamic resolution bounded by $28\times28\times576$ (max) and $28\times28\times16$ (min) pixels, matching the released framework defaults.

**Optimization.** LoRA runs use learning rate $2 \times 10^{-5}$ with cosine decay and 3% warmup; One epoch over this mixture is ≈2.3K optimizer steps at global batch 64, and the learning-rate schedule is stretched to the number of steps actually loaded. Every run uses maximum sequence length 1024, AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$ and weight decay 0.05, and seeds $\{0, 1, 2\}$.

**FINE configuration.** Table 6 lists the defaults. The FINE loss is the negative scaled expected JSD in (10), averaged over label positions with $n_\xi$ noise draws per example; the total loss is (60) with $\beta = 1.0$ by default. Gradients flow to both branches through the shared parameters; the patched branch's saved activations are retained as needed for backward and then released. Injection depth, relative noise, and $\beta$ are fixed within each run. No entropy threshold, KL threshold, or diagnostic-driven schedule is part of the training procedure.

**Evaluation harness.** All benchmarks are evaluated with lmms-eval (Li* et al., 2024) under greedy decoding, with at most 128 new tokens for multiple-choice and 512 for open-ended benchmarks, and with the official benchmark prompts; the suites and metrics are listed in Appendix J.

## G Joint Depth–Noise sensitivity Details

Table 7 summarizes the configuration.

Table 6: Main experiment configuration.

| Setting | Value |
|---|---|
| Models | Qwen3-VL-2B/4B/8B-Instruct (LoRA) |
| Training data | LLaVA-instruction-150K mixture, ≈150K examples, one epoch (≈2.3K steps at global batch 64) |
| LoRA | rank $r = 64$, $\alpha = 128$, dropout 0.0, applied to $q, k, v, o$ and gate/up/down; vision encoder and projector frozen |
| Learning rate | $2 \times 10^{-5}$ (LoRA) and $10^{-6}$ (full fine-tuning), cosine decay with 3% warmup |
| Batch and length | global batch 64, maximum sequence length 1024, one epoch |
| Optimizer | AdamW, $(\beta_1, \beta_2) = (0.9, 0.95)$, weight decay 0.05 |
| Seeds | $\{0, 1, 2\}$ |
| Injection depth $k$ | $k^* = \lfloor 0.5L \rfloor$ (14 for Qwen3-VL-2B; 18 for Qwen3-VL-4B/8B) |
| Relative noise $\sigma_{\text{rel}}$ | 0.2 |
| Regularization weight $\beta$ | 1.0 |
| Noise type | additive Gaussian on visual-token rows, $\tilde{Z}_1^{(k)} = Z_1^{(k)} + \sigma_k \epsilon$ with $\epsilon \sim \mathcal{N}(0, I_m)$ and $\sigma_k = \sigma_{\text{rel}} \sqrt{E_k / m}$ |
| Noise samples per step $n_\xi$ | 1 |
| Evaluation | lmms-eval (Li* et al., 2024), greedy decoding, at most 128 new tokens for multiple-choice and 512 for open-ended benchmarks, official prompts |

Table 7: joint depth–noise sensitivity configuration.

| Setting | Value |
|---|---|
| Model | Qwen3-VL-2B, LoRA as in Appendix F |
| Grid | $k \in \{4, 10, 14, 26\} \times \sigma_{\text{rel}} \in \{0.2, 0.5, 2.0\} \times seed \in \{0, 1, 2\}$ at $\beta = 1$, $n_\xi = 1$ |
| Training | 40K subset of LLaVA-style instruction mixture (Liu et al., 2023), 625 optimizer updates (one epoch at global batch 64), sequence length 1024. |
| Benchmarks | MMBench, ChartQA, and POPE. |

# H Ruling Out Generic Perturbation Sensitivity

We tests whether the effect of FINE is specifically associated with intervention on visual-aligned latent representations rather than generic sensitivity to perturbations of intermediate representations. We keep the training dataset, optimization procedure, intervention depth, noise calibration, and regularization coefficient fixed across all intervention conditions. We then evaluate the trained models on standard multimodal benchmarks and on a newly constructed held-out counterfactual image-pair set.

## H.1 Matched Intervention Controls

We train four configurations under the 40K subset of same LLaVA-Instruct-150K (Liu et al., 2023) instruction-tuning protocol: SFT, Text-Row, Visual-Row, and FINE. The SFT model is trained without the FINE regularizer and serves as the matched baseline. The other three configurations use the same prediction loss, intervention depth $k = 14$, relative noise level $\sigma_{\text{rel}} = 0.2$, regularization coefficient $\beta = 1$, optimizer, training schedule, and number of noise draws.

For each training example $i$, let $n_{v,i}$ and $n_{t,i}$ denote the numbers of eligible visual-token rows and prompt-text rows at the intervention layer. We define the matched row budget as

$$r_i = \min(n_{v,i}, n_{t,i}). \tag{61}$$

Special tokens, padding positions, and supervised response positions are excluded from both candidate sets. We uniformly sample $r_i$ distinct rows without replacement from each candidate set.

The **Visual-Row** condition adds Gaussian noise to the sampled visual rows, while the **Text-Row** condition adds the same type of noise to the sampled prompt-text rows. Therefore, the two controls have identical row budgets and per-coordinate noise scales but differ in intervention location. The **FINE** condition instead applies the intervention to the complete visual-state mask.

Table 8: Reference-setting weight sensitivity. Depth is 14 and relative noise is 0.2 for nonzero weights; $\beta = 0$ reuses SFT.

| $\beta$ | MMBench | ChartQA | MMStar |
|---|---|---|---|
| 0.3 | 76.4 | 73.8 | 54.7 |
| 1.0 | 77.4 | 74.1 | 55.2 |
| 2.0 | 77.6 | 73.6 | 55.7 |

Let $d$ denote the hidden dimension. For either matched partial-row control, the expected perturbation energy is

$$\mathbb{E}!\left[|\xi_i|2^2\right] = r_i d\sigma_g^2, \tag{62}$$

where the control noise scale is calibrated from the visual-state scale at the reference depth:

$$\sigma_g = 0.2\sqrt{\frac{E14,g}{m_g}}. \tag{63}$$

Here $E_{14,g}$ and $m_g$ denote the reference visual-state energy and number of eligible visual hidden-state elements for calibration stratum $g$.

The controls are matched to each other rather than to the full FINE configuration. Consequently, their total perturbation energy is generally smaller than that of FINE because FINE perturbs all eligible visual rows. This design isolates the *location* of the intervention while avoiding an additional difference in row count between the two controls.

All configurations use one noise draw per training example and the same regularization coefficient. If $r_i = 0$, the intervention term for that example is omitted and the occurrence of such examples is recorded.

It is important that the Text-Row condition is not interpreted as a "pure-text" representation. Because the backbone is early-fusion, intermediate prompt-text states can already contain information originating from the image. The purpose of the control is therefore narrower: it tests whether perturbing a matched number of rows has the same effect when the perturbation is applied to text-aligned rather than visual-aligned positions.

### H.2 Training and Evaluation Protocol

All four configurations are trained using the same Qwen3-VL-2B checkpoint, the LLaVA-Instruct-150K dataset, LoRA, optimizer, learning-rate schedule, number of epochs, batch size, and random-seed protocol. No counterfactual examples are included in training.

After training, every model is evaluated using the same evaluation pipeline on MMBench, ChartQA, and POPE. MMBench measures general multimodal understanding, ChartQA evaluates visual mathematical reasoning, and POPE provides an object-hallucination-oriented evaluation. The benchmark predictions are generated with identical decoding settings across all four models.

The three standard benchmark scores are reported together with the behavioral metrics described below. The benchmark examples used to construct the counterfactual evaluation set are separated from the examples used for the final counterfactual evaluation so that the counterfactual construction does not reuse the final evaluation instances.

### H.3 Construction of the Counterfactual Image-Pair Set

**Motivation.** Standard benchmark accuracy does not directly test whether a model follows changing visual evidence. For example, a model can obtain a correct answer using a strong textual prior even if its prediction changes little when the image changes. We therefore construct a separate held-out dataset of counterfactual image pairs.

Each counterfactual pair contains two images and a shared textual question such that the question is applicable to both images but the correct answers differ:

$$(x_i^{(a)}, q_i, y_i^{(a)}), \qquad (x_i^{(b)}, q_i, y_i^{(b)}), \qquad y_i^{(a)} \neq y_i^{(b)}. \tag{64}$$

The intended interpretation is that the textual context is held fixed while the visual evidence determines which answer is correct.

**Source examples.** We sample candidate examples from the evaluation portions of MMBench, ChartQA, and POPE. The source benchmark identity is retained as metadata, but the counterfactual set is evaluated as a separate behavioral dataset. Candidate examples are first normalized into a common representation containing an image identifier, question text, answer choices when present, ground-truth answer, and benchmark metadata.

**Semantic question grouping.** Questions from different images are not required to have exactly identical surface forms. We therefore use an external LLM API to identify groups of questions that express the same underlying information request. The LLM receives only the textual question and answer-schema information and is asked to return a canonicalized question and a group identifier. It is not asked to determine which visual answer is correct.

The API output is treated as a candidate-generation signal rather than as ground truth. We retain a pair only after deterministic checks verify that the two examples are compatible with a shared question template and that their benchmark-provided answers are different. When the compatibility decision is ambiguous, the pair is discarded or sent to manual verification.

**Pair filtering.** For each semantic question group, we consider pairs with different images. A candidate pair is retained only when all of the following conditions hold: (i) the two questions are semantically equivalent after canonicalization; (ii) both questions are valid for their corresponding images; (iii) the benchmark annotations provide different correct answers; (iv) the answer format is compatible between the two examples; and (v) neither example is used for model or hyperparameter selection.

For multiple-choice questions, the answer options are additionally checked to ensure that the two examples do not become trivially equivalent through identical option ordering. For numerical or free-form answers, benchmark normalization is applied before checking $y_i^{(a)} \neq y_i^{(b)}$.

**LLM-assisted construction pseudocode.** The following pseudocode describes the complete candidate-generation process. The API call can be implemented using any LLM provider supporting structured JSON output.

The actual API implementation should request structured output so that canonical questions and semantic groups can be parsed automatically. A representative prompt is:

> Given a multimodal evaluation question in text-only form, determine whether it expresses a reusable information request that could validly be asked about another image. Return a canonical question template, a short semantic-intent description, and an answer type. Do not infer the answer from any image and do not modify the benchmark ground-truth answer.

The API therefore assists with *question equivalence*, whereas the benchmark annotations and subsequent verification determine correctness. This separation is important because otherwise an LLM-generated label could leak an additional source of supervision into the behavioral evaluation.

### H.4 Paired Correctness

For each counterfactual pair $i$, let $(x_i^{(a)}, x_i^{(b)})$ denote the two images and let $(y_i^{(a)}, y_i^{(b)})$ denote their corresponding ground-truth answers. For a given model, define $c_i^{(a)}, c_i^{(b)} \in 0, 1$ as the correctness indicators of its two predictions.

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left[c_i^{(a)} = 1 \wedge c_i^{(b)} = 1\right]. \tag{65}$$

PC measures the fraction of counterfactual pairs for which the model answers both visual conditions correctly. It is intentionally stricter than per-image accuracy: a pair contributes to PC only when the model correctly tracks both visual states.

**Algorithm 1** LLM-assisted Counterfactual Image-Pair Construction

**Procedure** CONSTRUCTPAIRS($\mathcal{D}, \mathcal{A}$)
**Input:** Evaluation examples $\mathcal{D}$; LLM API $\mathcal{A}$
**Output:** Counterfactual image-pair set $\mathcal{C}$
Normalize each $e \in \mathcal{D}$ into $(id_e, q_e, y_e, a_e)$
Remove examples reserved for model selection or hyperparameter tuning
**for** each $e \in \mathcal{D}$ **do**
  Query $\mathcal{A}$ with $q_e$
  Obtain canonical question $\hat{q}_e$, semantic intent $\hat{z}_e$, and answer type $\hat{t}_e$
**end for**
Group examples according to $(\hat{z}_e, \hat{t}_e)$
$\mathcal{C} \leftarrow \emptyset$
**for** each group $G$ **do**
  **for** each pair $(e_i, e_j) \in G$ **do**
    **if** $id_{e_i} \neq id_{e_j}$ and $y_{e_i} \neq y_{e_j}$ **then**
      **if** COMPATIBLE($q_{e_i}, q_{e_j}$) **then**
        Construct candidate pair $(image_{e_i}, image_{e_j}, \hat{q}, y_{e_i}, y_{e_j})$
        **if** VALID($\cdot$) **then**
          $\mathcal{C} \leftarrow \mathcal{C} \cup \{(e_i, e_j)\}$
        **end if**
      **end if**
    **end if**
  **end for**
**end for**
Remove duplicate and ambiguous pairs from $\mathcal{C}$
Manually verify the remaining candidate pairs
**return** $\mathcal{C}$
**End Procedure**

### H.5 RESOLVED AND INTRODUCED ERRORS

To quantify behavioral changes relative to matched SFT, we compare the correctness of each evaluated prediction with the prediction of the matched SFT model on the same example.

$$\frac{\sum_i \mathbb{1}[c_i^{\mathrm{SFT}} = 0 \wedge c_i^{\mathrm{method}} = 1]}{\sum_i \mathbb{1}[c_i^{\mathrm{SFT}} = 0]}. \tag{66}$$

Resolved therefore measures the fraction of SFT errors that become correct under the evaluated intervention.

$$\frac{\sum_i \mathbb{1}[c_i^{\mathrm{SFT}} = 1 \wedge c_i^{\mathrm{method}} = 0]}{\sum_i \mathbb{1}[c_i^{\mathrm{SFT}} = 1]}. \tag{67}$$

Introduced measures the fraction of initially correct SFT predictions that become incorrect after intervention. We report the corresponding denominators for every benchmark. These statistics are descriptive and are not used for model selection or hyperparameter selection.

### H.6 CORRECTNESS VERIFICATION

All methods are evaluated on exactly the same held-out counterfactual examples. For automatically evaluable benchmark questions, correctness follows the benchmark-specific answer-matching protocol. When manual verification is required, three annotators independently assess the model responses without access to the intervention condition, and the final correctness label is determined by majority vote.

| Component | Configuration |
|---|---|
| Base model | Qwen3-VL-2B |
| Training data | LLaVA-Instruct-150K |
| Training method | LoRA |
| Baseline | SFT |
| Reference intervention | FINE, full visual-state mask |
| Reference setting | $(k, \sigma_{\mathrm{rel}}, \beta) = (14, 0.2, 1)$ |
| Noise draws | One |
| Visual-Row | $r_i$ sampled visual-token rows |
| Text-Row | $r_i$ sampled prompt-text rows |
| Row budget | $r_i = \min(n_{v,i}, n_{t,i})$ |
| Row sampling | Uniform without replacement |
| Excluded rows | Special, padding, supervised response positions |
| Control noise | Same $\sigma_g$ for Text-Row and Visual-Row |
| Training benchmarks | Not used for training beyond LLaVA-Instruct-150K |
| Evaluation | MMBench, ChartQA, POPE |
| Counterfactual data | Held-out pairs derived from evaluation examples |
| Counterfactual criterion | Same question, different correct answers |
| PC | Both members of a pair correct |
| Resolved | SFT incorrect $\rightarrow$ method correct |
| Introduced | SFT correct $\rightarrow$ method incorrect |
| Manual annotation | Three blinded annotators, majority vote |

Table 9: Configuration for the visual-reliance and behavioral-control experiments. All intervention models are trained under the same LLaVA-Instruct-150K instruction-tuning condition. Text-Row and Visual-Row are matched in row budget and per-coordinate noise scale, while FINE uses the full visual-state mask. Counterfactual examples are held out from training and model selection.

The annotation procedure is performed after the counterfactual pairs have been constructed and does not expose the model identity to the annotators. Consequently, the labels used for PC, Resolved, and Introduced are shared across all evaluated methods.

### H.7 Configuration

Table 9 summarizes the complete configuration.

## I Noise-Scale Calibration Details

The Gaussian corruption of (8) is fully specified only once $\sigma$ is fixed. The analysis treats $\sigma$ as constant within a fixed-dimension stratum, and Section B parameterises it as $\sigma_{k,g} = \sigma_{\mathrm{rel}} s_{k,g}$. This section specifies how $s_{k,g}$ is estimated, what the resulting artifact freezes, and what the procedure controls and what it leaves uncontrolled. Every experiments follows the same procedure, and each estimates its scales on a calibration split kept source-image-disjoint from that experiment's training, selection, diagnostic, and evaluation data.

**Estimator.** Take the initial checkpoint of the target model before any adaptation, in evaluation mode, and run one forward pass with gradients disabled over each calibration example. Capture the clean visual state $Z_1^{(k)}$ entering decoder block $k$ for every depth $k$ of the block's depth grid, and retain only the rows carrying the image-token id. For stratum $g$, let $S_g$ be the set of calibration examples whose image produces that grid, $n_g = |S_g|$, $v_g$ the number of visual tokens, $d$ the hidden size, and $m_g = v_g d$; for the models used here $v_g = t_g h_g w_g / 4$ follows from the $2 \times 2$ spatial merge of the vision tower, so that a $26 \times 32$ patch grid with $d = 2048$ gives $v_g = 208$ and $m_g = 425{,}984$. The per-stratum scale is the root-mean-square of the accumulated per-element power,

$$\widehat{E}_{k,g} = \frac{1}{n_g} \sum_{i \in S_g} \left\| Z_1^{(k)}(x_i) \right\|^2, \qquad \widehat{s}_{k,g} = \sqrt{\widehat{E}_{k,g} / m_g}, \tag{68}$$

where the power is accumulated in float32, and the estimator requires exactly one image per example so that the visual-token count is fixed within a stratum; training verifies that the token count of each example still matches the recorded $m_g$ and raises otherwise. The artifact stores, for every depth and stratum, the element count $m_g$, the number of contributing examples $n_g$, the summed squared power, and the derived $\widehat{E}_{k,g}$ and $\widehat{s}_{k,g}$, together with the model configuration hash, the calibration-

manifest version, and the protocol version. Every training and profiling run records the hash of the artifact it used, so runs sharing a grid cell share one noise law.

**Strata, coverage, and the pooled fallback.** Conditioning on the image-grid stratum keeps $m_g$ fixed within a stratum, which is required for the fixed-dimension statement of Theorem 3.4: pooling grids with different token counts would mix different $m$. It also respects the empirical fact that the visual-state scale depends on how many patches an image produces. Coverage is nevertheless partial, because the calibration split is held out and therefore small relative to the variety of geometries the training mixture produces. In the released experiments recipe the calibration manifest contains 1,000 images drawn with a fixed seed from a pool of 349,034 images, allocated proportionally across source families and source-image-disjoint from the 40K training subset; these images span 77 grid strata, with $n_g$ ranging from 1 to 141 and most strata contributing only a few examples. For a depth and grid with no calibrated stratum, the scale falls back to the pooled per-depth RMS,

$$\widehat{s}_k^{\mathrm{pool}} = \sqrt{\frac{\sum_g \sum_{i \in S_g} \left\| Z_1^{(k)}(x_i) \right\|^2}{\sum_g n_g m_g}}, \tag{69}$$

which is the noise law of Theorem 3.4 evaluated over the whole calibration split instead of a single stratum. Each unseen grid is logged once when the fallback is used. Two opposing effects are worth recording: because $m_g$ is large, the within-image average in (68) is precise even for $n_g = 1$, but the estimated target is an expectation over images, and images sharing a grid still differ in content, so strata with one or two examples carry cross-image sampling noise, whereas (69) averages over the entire split.

**What the frozen scale controls and what it does not.** The frozen scale fixes the units of the perturbation, so that $\sigma_{\mathrm{rel}}$ is a dimensionless, model- and resolution-independent control and a depth–noise grid can be compared on one axis. It is not a complete specification of the corruption. First, it matches only the second moment of the visual state: the implemented noise is isotropic with a single scalar per stratum, so per-element power is preserved while the covariance across visual coordinates is not. Second, it is estimated at the initial checkpoint, before adaptation; as training changes the scale of the visual states, the realized relative noise drifts away from the declared $\sigma_{\mathrm{rel}}$, which is why post-training activation power is recorded alongside the probe results (Appendix G). Third, the strata condition on grid geometry rather than content, so images of equal size share a scale regardless of visual complexity. Finally, the estimate must not be recomputed from the profiling or training batch: doing so would make $\sigma$ depend on the very samples whose divergence it scales, and the fixed-variance assumption of Theorem 3.4 would no longer describe the batch being optimised. Keeping the split disjoint by source image additionally ensures that no image used to set the scale is also an image the model is trained on.

## J Benchmark Suites, Splits, and Scoring

**Suites.** The main suite covers MMBench (Liu et al., 2024) for general multimodal understanding, ChartQA (Masry et al., 2022) for chart reasoning, MMStar (Chen et al., 2024) for challenging multimodal reasoning, MME (Fu et al., 2026) for perception and cognition, and POPE (Li et al., 2023) for hallucination. The ablation subsets reuse MMBench, ChartQA, and POPE. The frozen-model profiles use POPE for object presence, MMBench for perception, ChartQA for vision-necessary reasoning, and GQA for grounding, plus a text-sufficient control whose images are stripped.

**Metrics.** Accuracy is reported per benchmark, together with the aggregate MME score and the POPE F1. Paired correctness (PC) is the fraction of valid counterfactual pairs on which both images' answers are correct; $\Delta$Text is the text-only score difference from matched SFT, reported with a one-sided 95% lower bound against the prespecified noninferiority margin. All scores are produced by the harness of Appendix F.

## K Common Software, Hardware, and Artifacts

**Software.** The real-model configuration specifies PyTorch 2.6.0, Transformers (development version 4.57), PEFT 0.17.1, DeepSpeed 0.17.1, FlashAttention 2.7.4, and the lmms-eval harness (Li* et al., 2024), in bfloat16 with gradient checkpointing and sequence length 1024.

**Hardware.** Each training run uses one node with either 2×A100-80GB or 2×RTX4090-24GB, with ZeRO-2; per-device batch sizes are chosen per model and GPU (2B: 8 on A100/4 on RTX4090; 4B: 4/2; 8B: 4/2 with proportionally more gradient accumulation), always reaching the global batch of 64 at sequence length 1024.

**Artifacts and release.** The release includes code, data licenses, exact dependency revisions, versioned sample manifests, calibration records, per-draw Gaussian scores, control residuals, and bootstrap summaries.