PRC: An Efficient and Unified Framework for Evaluating Robustness and Controllability of Large Language Models on Few-shot Prompt Variations

Chunpeng Ma
Megagon Labs, Tokyo, Recruit Co. Ltd.
Marunouchi 1-9-2, Chiyoda, Tokyo, Japan
ma.chunpeng@megagon.ai

Abstract

Understanding the impact of prompt variations on large language model (LLM) outputs is critical. Despite extensive research, two challenges persist: conflated terminology stemming from loosely defined concepts, and inefficient evaluations reliant on output generation. We first formalize prompt categorization to clearly define prompt robustness and controllability. We then introduce the prompt vector (PV), a compact internal representation that reformulates these properties at the representation level. Building on PV, we propose PRC, an efficient and unified framework to systematically evaluate LLMs under few-shot prompt variations without requiring costly autoregressive decoding. Experiments across diverse instruction-tuned LLMs demonstrate that PRC tightly correlates with traditional output-based metrics while substantially reducing computational overhead. Overall, PRC offers a principled and highly efficient tool for diagnosing LLM behavior.

1 Introduction

While large language models (LLMs) have demonstrated impressive capabilities across diverse applications (Matsuda et al., 2025; Niwa et al., 2025), they remain highly sensitive to their input prompts (Giray, 2023; Wang et al., 2025). Understanding how prompt variations impact LLM outputs is therefore essential. Despite extensive research, systematic analysis is hindered by two major issues. First, the terminology is heavily fragmented; concepts such as sensitivity (Lu et al., 2024), robustness (Wu et al., 2025), controllability (Minder et al., 2025), and brittleness (Ceron et al., 2024) are often conflated without explicit boundaries, hindering both evaluation (Liu and Chu, 2026) and mitigation (Hejabi et al., 2025). Second, existing methods rely heavily on comparing generated outputs (Voronov et al., 2024; Mizrahi et al., 2024), making the evaluation computationally expensive due to autoregressive decoding.

To address the terminology gap, we first provide a conceptual clarification of these related notions. While prompts can vary in numerous ways, our framework specifically focuses on variations within in-context learning: the manipulation of few-shot examples. This controlled setting allows us to strictly quantify the exact information encoded in the prompt. We organize LLM behaviors into two cohesive and mathematically defined dimensions: *prompt robustness* and *controllability* (see §2).

Next, to address the efficiency bottleneck, we identify a compact representation of prompts within LLMs, termed the *prompt vector* (PV). Building upon PV, we propose **PRC** (**P**rompt **R**obustness and **C**ontrollability), an efficient and unified evaluation framework (see §3). Unlike existing output-based methods, PRC evaluates robustness and controllability directly at the representation level. By bypassing output generation, it significantly reduces computational overhead.

We evaluate several prominent instruction-tuned LLMs across diverse tasks (see §4) and demonstrate the feasibility of PV (see §5), revealing new insights into model-specific differences.

Our contributions are summarized as follows.

- We resolve fragmented terminology by providing clear, formalized definitions of prompt robustness and controllability.
- We introduce the PV, a compact and expressive internal representation of prompts.
- We propose PRC, an efficient, output-free framework that unifies the evaluation of robustness and controllability.
- We empirically uncover distinct behavioral insights regarding robustness and controllability across multiple state-of-the-art LLMs.

2 Clarification of Concepts

Existing literature often lacks explicit prompt categorizations and clear concept boundaries (Qiang

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 204–222
November 6-10, 2026 ©2026 Association for Computational Linguistics

et al., 2024; Sclar et al., 2024; Errica et al., 2025; Seleznyov et al., 2025). Chatterjee et al. (2024) were among the first to propose a clear definition of *intent-aligned* prompts.

Intent-aligned Prompts

[Extracted from Chatterjee et al. (2024)] Any two prompts x_1 and x_2 are said to be *intent-aligned* despite variations in their wording or template or inclusion of minor spelling errors, if they are designed to elicit **responses from a language model** based on the same underlying goal, intent or meaning.

While *intent-aligned* prompts are defined as those sharing an underlying goal despite superficial variations, their evaluation implicitly entangles input categorization with output behavior. To systematically analyze LLMs, we must first categorize prompts strictly based on their encoded information, enabling formal, decoupled definitions of robustness and controllability.

2.1 Categorization of Prompts

A prompt contains various forms of information (e.g., instructions, examples). While quantifying *information* across arbitrary zero-shot variations is highly subjective, in-context learning provides a controlled environment. Here, information is objectively operationalized via the inclusion or ordering of few-shot examples. Based on this, we categorize the relationship between two prompts p_1 and p_2.

Equivalent Prompts

Given two prompts p_1 and p_2, they are called *equivalent* if and only if they encode exactly the same information. We denote this relation by $p_1 \simeq p_2$.

Incremental Prompts

Given two prompts p_1 and p_2, we say that p_2 is *incremental* with respect to p_1 if and only if all information encoded in p_1 is contained in p_2. We denote this relation by $p_1 \subset p_2$ or, equivalently, $p_2 \supset p_1$.

In our settings, shuffling the order of identical examples yields equivalent prompts, while adding completely new in-context examples to a prompt makes it incremental. For example, consider a prompt p_1 containing few-shot examples A and B, and p_2 containing B and A. Since they present identical task information, $p_1 \simeq p_2$. If a new example C is added to p_1 to form p_3, then $p_1 \subset p_3$.

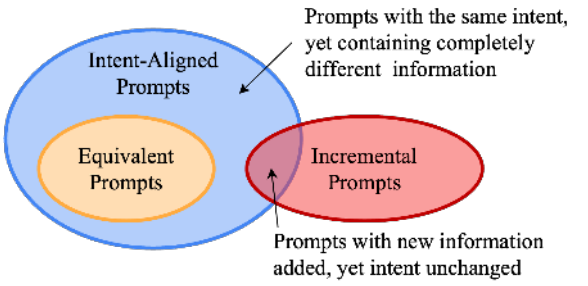


Figure 1: Relationship among equivalent prompts, incremental prompts, and intent-aligned prompts.

As illustrated in Figure 1, because incremental prompts strictly require new information, they form a disjoint set from equivalent prompts. Together, they represent the foundational axes needed to define behavioral invariance and responsiveness.

2.2 Robustness and Controllability

Intuitively, when we describe an LLM as *robust*, we expect its outputs *not* to change when presented with *the same* prompts. In contrast, when we describe an LLM as *controllable*, we expect its outputs *to* change in response to *different* prompts.

Let $\mathcal{M}$ be an LLM and $s_{\mathcal{M}}$ denote a specific internal or output state (e.g., hidden representations or probability distributions). We formulate two core properties using a state-mapping function $f(s_{\mathcal{M}}; p) \in \mathbb{R}^f$ and a divergence metric $\mathfrak{D}_f : \mathbb{R}^f \times \mathbb{R}^f \to [0, +\infty)$.

Local Prompt Robustness

For a set of equivalent prompts $\mathcal{P} = \{p_1, \ldots, p_P\}$ where $p_i \simeq p_j$, the *local prompt robustness* is defined as:

$$\mathbf{Rob}_{local}(\mathcal{M}, \mathcal{P}; s_{\mathcal{M}}) := \frac{1}{P(P-1)} \sum_{i=1}^{P} \sum_{j=1}^{P} \mathfrak{D}_f \left(f(s_{\mathcal{M}}; p_i), f(s_{\mathcal{M}}; p_j) \right),$$

The local prompt robustness reflects the average state variation on equivalent prompts; thus, a lower $\mathbf{Rob}_{local}$ signifies higher robustness. Note that we omit the input divergence denominator $\mathfrak{D}_e(e(p_i), e(p_j))$ here. Since equivalent prompts encode identical information, their theoretical input

distance is negligible, making such normalization mathematically ill-posed.

Local Controllability

For a sequence of strictly incremental prompts $\mathcal{P} = \{p_1, \ldots, p_P\}$ where $p_i \subset p_{i+1}$, the *local controllability* is defined as:

$$\mathbf{Con}_{local}(\mathcal{M}, \mathcal{P}; s_{\mathcal{M}}) := \frac{1}{P-1} \sum_{i=1}^{P-1} \frac{\mathfrak{D}_f\left(f(s_{\mathcal{M}}; p_i), f(s_{\mathcal{M}}; p_{i+1})\right)}{\mathfrak{D}_e\left(e(p_i), e(p_{i+1})\right)},$$

where $e(p) \in \mathbb{R}^e$ represents the prompt input, and $\mathfrak{D}_e$ measures input space divergence.

The local controllability captures the average state shift relative to informational changes in the prompt. An extremely small value of $\mathbf{Con}_{local}$ indicates that the LLM varies little even when the prompt changes substantially, whereas an extremely large value implies excessive sensitivity. A controllable model should avoid both extremes.

Given a dataset $\mathfrak{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_N\}$, we can define the global metrics $\mathbf{Rob}(\mathcal{M}, \mathfrak{P}; s_{\mathcal{M}})$ and $\mathbf{Con}(\mathcal{M}, \mathfrak{P}; s_{\mathcal{M}})$ simply by averaging the respective local scores across the N sets. Throughout this paper, all experiments and discussions are focused on global versions of robustness and controllability.

Connections between Robustness and Controllability. The definitions of robustness and controllability share several common characteristics:

- Dependence on a set of prompts $\mathcal{P}$.
- Flexibility in the choice of state $s_{\mathcal{M}}$.
- Use of divergences over distributed representations $\mathfrak{D}_f$ (and $\mathfrak{D}_e$).

These similarities highlight their intrinsic connection that has been overlooked in prior research.

3 PRC: Prompt Robustness and Controllability Evaluation

The state-mapping function $f(s_{\mathcal{M}}; p) \in \mathbb{R}^f$ and the divergence metric $\mathfrak{D}_f$ are the core components of our proposed definitions. In prior works, $s_{\mathcal{M}}$ typically represents the generated output, and $f(s_{\mathcal{M}}; p)$ corresponds to either task performance or probability distributions. This output-centric paradigm necessitates metrics such as performance deviation (Mizrahi et al., 2024; Wu et al., 2025), Kullback–Leibler divergence (Sun et al., 2024), or Jensen–Shannon divergence (Qiang et al., 2024). In contrast, we identify a compact, internal representation $f(s_{\mathcal{M}}; p)$ that completely bypasses output generation. We term this the *prompt vector* (PV), conceptually inspired by, yet fundamentally distinct from, function vectors (Todd et al., 2024).

3.1 Preliminary: Function Vector

According to Todd et al. (2024), within an autoregressive transformer, a *function vector* (FV) compactly represents a task t as the sum of mean outputs from a set of key attention heads $\mathcal{A}$:

$$\mathbf{FV}(t) := \sum_{(l,j)\in\mathcal{A}} \bar{a}^t_{lj}, \quad \text{where } \bar{a}^t_{lj} = \frac{1}{N} \sum_{\mathcal{P}^t_i \in \mathfrak{P}^t} a_{lj}(p^t_i). \tag{1}$$

Here, $a_{lj}(p^t_i)$ denotes the output of the j-th attention head in the l-th layer given prompt p^t_i, and $\mathcal{A}$ consists of heads exhibiting the highest Average Indirect Effect (AIE) across tasks under strict single-token output constraints. While FVs offer compact task-level representations, they are ill-suited for evaluating prompt robustness and controllability.

- **Format Constraints:** FVs strictly require single-word outputs and rigid prompt templates, failing to generalize to arbitrary prompt structures or multi-token generations.
- **Lack of Prompt Differentiation:** FVs treat all prompts within a task uniformly, whereas evaluating robustness and controllability necessitates structured sets of equivalent or incremental prompts.
- **Task-Level Abstraction:** FVs abstract across an entire dataset to form $\mathbf{FV}(t)$. In contrast, prompt evaluation requires a fine-grained, prompt-specific representation $f(s_{\mathcal{M}}; p)$ derived from individual inputs.

3.2 Prompt Vector

To address these limitations, we propose the *prompt vector* (PV). Given a dataset $\mathfrak{P} = \{\mathcal{P}_1, \cdots, \mathcal{P}_N\}$, where each $\mathcal{P}_i = \{p^1_i, \cdots, p^{P_i}_i\}$ represents a set of equivalent or incremental prompts, the PV of an individual prompt p is defined as a learned linear combination of its hidden layer representations:

$$\mathbf{PV}(p) := \sum_{l\in\mathcal{L}(\mathfrak{P})} \lambda_l \mathbf{h}_l(p), \tag{2}$$

where $\mathbf{h}_l(p)$ denotes the vector representation at the l-th layer of a transformer with L layers, with the embedding layer treated as the 0-th layer and

the output probability distribution treated as the $(L+1)$-th layer.

The layer index set $\mathcal{L}(\mathfrak{P})$, whose cardinality $|\mathcal{L}(\mathfrak{P})|$ is a hyper-parameter, consists of the layers with the largest *average divergence* (AD) computed on a small warm-up subset $\mathfrak{P}_W = \{\mathcal{P}_1, \cdots, \mathcal{P}_W\} \subset \mathfrak{P}$, with $W \ll N$:

$$\mathrm{AD}(l) = \frac{1}{W}\sum_{i=1}^{W}\frac{1}{P_i}\sum_{j=1}^{P_i}\mathfrak{D}_l\left(\mathbf{h}_l(p_i^j), \mathbf{h}_l(p_i^{\lceil \frac{P_i}{2} \rceil})\right), \tag{3}$$

where $\mathfrak{D}_l$ is the cosine distance when $l \leq L$, and is the Kullback–Leibler divergence when $l = L+1$.

The optimal coefficients $\lambda_l \geq 0$ are determined by minimizing the *average sensitivity difference* (ASD) over the warm-up set $\mathfrak{P}_W$:

$$\mathrm{ASD}(\mathfrak{P}_W) = \frac{1}{W}\sum_{i=1}^{W}\frac{1}{P_i}\sum_{j=1}^{P_i}\left|\psi_{\mathcal{M},\mathcal{P}_i} - \mathfrak{D}_{\mathbf{PV}}(p_i^j)\right|. \tag{4}$$

Here, $\psi_{\mathcal{M},\mathcal{P}_i}$ is the sensitivity function introduced by (Chatterjee et al., 2024). $\mathfrak{D}_{\mathbf{PV}}(p_i^j)$ denotes the average pairwise internal divergence between $\mathbf{PV}(p_i^j)$ and all other pairs of PVs in $\mathcal{P}_i$. This provides a mathematically sound, set-level supervision signal to learn layer contributions.

3.2.1 Remarks on Prompt Vector

Compared with FV, the PV is more suitable for evaluating robustness and controllability.

- **Freedom in prompt and task formats.** Unlike FV, the computation of PV imposes no constraints on the format of the prompt or the task, which enables broad applicability across diverse evaluation settings.
- **Discrimination among prompt categories.** Prompts in the dataset $\mathfrak{P}$ are explicitly organized into N sets of equivalent or incremental prompts, which aligns naturally with the definitions of robustness and controllability.
- **Fine-grained representation.** PV is defined as a function of an individual prompt p, capturing how the LLM $\mathcal{M}$ represents that prompt, rather than abstracting over an entire task.

Beyond the properties above, PV differs from FV in several additional aspects.

Hidden layers rather than attention heads. This design is motivated by the intuition that hidden layers primarily encode semantic representations, whereas attention heads mainly characterize information flow. Since semantic content varies across prompts, hidden-layer representations are better suited for constructing PV. This architectural design is directly corroborated by recent theoretical findings that LLMs map even structurally similar, meaning-preserving prompts to highly dispersed continuous representations (Liu and Chu, 2026).

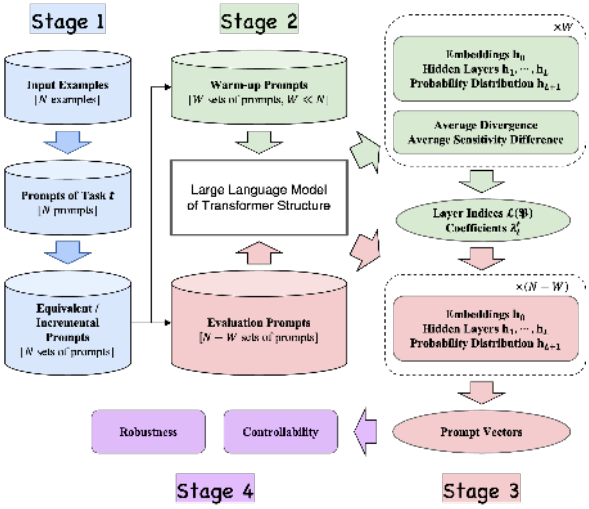


Figure 2: Framework of PRC, consisted of four stages for evaluation of robustness and controllability.

Linear combination rather than summation. Different layers encode information at varying levels of granularity, and this variation is model-dependent. Introducing learnable coefficients allows adjustment of contribution of each layer.

AD and ASD rather than AIE. AD is employed to identify layers exhibiting the largest representational variation across prompts, while ASD is used to ensure that PV remains correlated with output-based sensitivity. Both criteria make PV suitable for evaluating robustness and controllability.

Necessity of $\mathfrak{P}_W$. Because both the coefficients λ_l and the layer set $\mathcal{L}(\mathfrak{P})$ must be estimated, PV cannot be constructed entirely from scratch. Nevertheless, empirical results show that a sufficiently small W is adequate, enabling efficient evaluation.

3.3 Evaluation Framework

The PRC framework is illustrated in Figure 2.

Stage 1: Construction of $\mathfrak{P}$. Each example in the raw dataset of size N is first converted into a prompt suitable for input to an LLM. Given an initial prompt p_i, we then construct a corresponding prompt set $\mathcal{P}_i$ according to the evaluation objective.

For robustness evaluation, we initialize $\mathcal{P}_i := \{p_i\}$ and iteratively augment it by adding equivalent prompts, i.e., $\mathcal{P}_i := \mathcal{P}_i \bigcup \{p_i^j\}$, where $p_i^j \simeq p_i$, until $|\mathcal{P}_i| = P_i$. For controllability evaluation, we initialize $\mathcal{P}_i := \{p_i\}$ and iteratively expand the set by constructing two new prompts at each step: p_i^-, obtained by removing information from the first element of $\mathcal{P}_i$, and p_i^+, obtained by adding information to the last element of $\mathcal{P}_i$. The prompt set is then updated as $\mathcal{P}_i := \{p_i^-\} \bigcup \mathcal{P}_i \bigcup \{p_i^+\}$. This process is repeated until $|\mathcal{P}_i| = P_i$.

All prompt sets $\mathcal{P}_i$, $i \in [1, N]$ are collected to form $\mathfrak{P}$, which is further partitioned into two subsets: a warm-up set $\mathfrak{P}_W$, used in Stage 2, and an evaluation set $\mathfrak{P} \setminus \mathfrak{P}_W$, used in Stage 3.

Stage 2: Determination of $\mathcal{L}(\mathfrak{P})$ and λ_l. The computation of PV depends on both the selected layer set $\mathcal{L}(\mathfrak{P})$ and the coefficients λ_l. The layer set $\mathcal{L}(\mathfrak{P})$ is determined by computing $\mathrm{AD}(l)$ for all layers using Eq. (3) on $\mathfrak{P}_W$ and selecting the top $|\mathcal{L}(\mathfrak{P})|$ layers. The coefficients λ_l are obtained by minimizing the objective in Eq. (4), which we solve using L-BFGS-B algorithm (Byrd et al., 1995) with non-negativity constraints $\lambda_l \geq 0$.

Stage 3: Computation of Prompt Vectors. With $\mathcal{L}(\mathfrak{P})$ and the optimal coefficients λ_l determined in Stage 2, the prompt vector $\mathbf{PV}(p_i)$ is computed using Eq. (2) for each prompt $p_i \in \mathcal{P}_i \in \mathfrak{P} \setminus \mathfrak{P}_W$.

Stage 4: Evaluation of Robustness and Controllability. Using $\mathbf{PV}(p_i)$ as $f(s_{\mathcal{M}}; p_i)$, we compute the local evaluations $\mathbf{Rob}_{local}(\mathcal{M}, \mathcal{P}_i; s_{\mathcal{M}})$ or $\mathbf{Con}_{local}(\mathcal{M}, \mathcal{P}_i; s_{\mathcal{M}})$ according to the definitions in Section 2. Aggregating over all prompt sets yields the global robustness $\mathbf{Rob}(\mathcal{M}, \mathfrak{P}; s_{\mathcal{M}})$ and global controllability $\mathbf{Con}(\mathcal{M}, \mathfrak{P}; s_{\mathcal{M}})$.

Remark: Origin of Efficiency. By evaluating directly on internal representations, PRC bypasses both the final LM head matrix multiplication and sequential token generation. While per-instance computational savings over output distribution metrics (such as POSIX) are modest for strictly single-token prediction tasks, PRC provides massive speedups when scaled across large-scale prompt permutation suites or multi-token output settings. Importantly, PV extracts the static semantic integration of the prompt prior to decoding, completely decoupling model diagnostic throughput from context-dependent generation lengths.

Remark: Offline Calibration vs. Online Evaluation. Although PRC involves a four-stage architecture, we highlight that **Stages 1–3 constitute a fully automated, one-time offline setup** per dataset-model pair. Specifically, Stage 2 optimizes the layer coefficients λ_l over a tiny warm-up subset (10% of $\mathfrak{P}$). Once calibrated, these coefficients are fixed. In production or large-scale evaluation scenarios (Stage 4), PRC bypasses sequential token generation and vocabulary projection entirely, processing incoming prompt variations in parallel via single-pass forward propagation.

4 Experiments

4.1 Configurations

Datasets. We conduct experiments on two datasets. (1) The test split of the **MMLU** dataset (Hendrycks et al., 2021), containing approximately 15,000 multiple-choice questions spanning 57 domains. (2) Eight tasks from the **Big-Bench Hard** (BBH) benchmark (Suzgun et al., 2022): causal judgment (CJ.), disambiguation QA (DQ.), Dyck languages (DL.), formal fallacies syllogisms negation (FF.), hyperbaton (HB.), logical deduction five objects (LD.), reasoning about colored objects (RC.), and salient translation error detection (TE.).

LLMs. We evaluate the following instruction-tuned LLMs: gemma-3-4b-it (Team et al., 2025), Qwen2.5-7B-Instruct (Qwen et al., 2024), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023).

Prompts. We design prompts in a way that allows precise control over the information they contain. Each prompt consists of a brief task description, followed by three examples (without chain-of-thought reasoning), and a target input instance. Detailed prompt templates are provided in Appendix A.

Equivalent Prompts. Equivalent prompts are constructed by shuffling the order of the few-shot examples within a prompt. Each prompt set contains six equivalent prompts. We deliberately avoid commonly used perturbation strategies such as paraphrasing or rule-based modifications, as these operations may inadvertently alter the underlying information encoded in the prompt.

Incremental Prompts. Incremental prompts are created by progressively adding or removing few-shot examples from the prompt. Each prompt set

contains seven incremental prompts. While alternative strategies for constructing incremental prompts exist (e.g., incorporating domain knowledge or refining task descriptions) it is non-trivial and costly to ensure such modifications strictly satisfy the incremental property. Therefore, we adopt example-based modifications for reliability and clarity.

4.2 Robustness and Controllability of LLMs

We compare three representative approaches:

- **Task-performance-based evaluation.** Task performance is widely used to assess how LLM behavior changes under prompt variations (Lu et al., 2022; Voronov et al., 2024; Sclar et al., 2024). We report the mean and standard deviation of task accuracy.
- **Probability-distribution-based evaluation.** The output probability distribution of LLMs is another commonly adopted metric (Qiang et al., 2024; Sun et al., 2024). We compute POSIX (Chatterjee et al., 2024) to quantify distributional differences.
- **Prompt-vector-based evaluation.** This corresponds to PRC, our proposed evaluation method based on prompt vectors.

In prior work, the first two approaches are typically used to measure (often implicitly defined) sensitivity on intent-aligned prompts. In contrast, we apply these measures to equivalent and incremental prompts as approximations of robustness and controllability, respectively. Tables 1 and 2 report robustness and controllability results on the BBH tasks. Note that for the task-performance-based evaluation, we report the mean and standard deviation over five runs; however, for the POSIX and PRC metrics, we empirically observed that the results vary minimally across different runs, so we report the representative values without standard deviations. We draw the following observations.

Robustness and controllability cannot be reliably evaluated using task performance alone. Although the standard deviation of accuracy varies across tasks and models, it provides limited insight into the intrinsic properties of LLMs. This is because accuracy variance is influenced not only by the intrinsic understanding of the model itself, but also by the decoding strategy (e.g., temperature sampling) and surface-level formatting sensitivities during autoregressive generation. When output-based metrics and PRC disagree, PRC offers a uniquely valuable signal. PV isolates the model's foundational semantic encoding of the prompt before the decoding phase. Therefore, it provides a more deterministic and intrinsic measure of the model's true internal state shifts, independent of output generation artifacts.

Strong correlation between POSIX and PRC. Comparisons between POSIX and PRC reveal a strong correlation between the two metrics. Notably, except for the small warm-up set $\mathfrak{P}_W$, PRC does not require output generation, resulting in substantially improved evaluation efficiency.

Controllability varies substantially across LLMs. We observe pronounced differences in controllability across models, likely reflecting differences in training strategies. The relatively large value of $\mathbf{Con}(\texttt{gemma})$ may stem from distillation from larger teacher models, whereas the comparatively low value of $\mathbf{Con}(\texttt{Llama})$ may be attributed to the extensive instruction data used during fine-tuning.

5 Discussion: Feasibility of PV

The core idea of PRC is to use the prompt vector $\mathbf{PV}(p)$ as the $f(s_{\mathcal{M}}; p) \in \mathbb{R}^f$ for evaluation. This design is feasible only if three assumptions hold. We demonstrate that all assumptions are satisfied.

- **Suitability for robustness evaluation.** Given equivalent prompts $p_1 \simeq p_2$, the divergence $\mathfrak{D}_f\left(\mathbf{PV}(p_1), \mathbf{PV}(p_2)\right)$ should correlate with robustness metrics computed from the corresponding outputs $\mathbf{y}(p_1)$ and $\mathbf{y}(p_2)$.
- **Suitability for controllability evaluation.** Given incremental prompts $p_1 \subseteq p_2$, the normalized divergence $\mathfrak{D}_f\left(\mathbf{PV}(p_1), \mathbf{PV}(p_2)\right) / \mathfrak{D}_e\left(e(p_1), e(p_2)\right)$ should correlate with the difference in information encoded in p_1 and p_2.
- **Efficiency of evaluation.** Stage 2 is the most time-consuming component of PRC; therefore, the warm-up set size $|\mathfrak{P}_W|$ must be sufficiently small to ensure efficiency.

5.1 Suitability for Robustness Evaluation

Given a dataset $\mathfrak{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_N\}$ in which each $\mathcal{P}_i$ consists of equivalent prompts, the sensitivity function $\psi_{\mathcal{M},\mathcal{P}_i}$ provides a suitable robustness metric based on model outputs. We examine the correlation between $\psi_{\mathcal{M},\mathcal{P}_i}$ and $\mathbf{Rob}_{local}(\mathcal{M}, \mathcal{P}_i; \mathbf{PV})$,

Task	CJ.	DQ.	DL.	FF.	HB.	LD.	RC.	TE.
Evaluation based on task performance: mean and standard deviation of task accuracy.								
gemma	0.497 ±0.028	0.445 ±0.021	0.566 ±0.014	0.511 ±0.026	0.558 ±0.016	0.453 ±0.015	0.569 ±0.012	0.520 ±0.015
Qwen	0.564 ±0.015	0.565 ±0.008	0.595 ±0.021	0.592 ±0.010	0.625 ±0.026	0.441 ±0.008	0.676 ±0.014	0.521 ±0.012
Llama	0.482 ±0.019	0.445 ±0.024	0.513 ±0.017	0.507 ±0.022	0.751 ±0.027	0.478 ±0.018	0.605 ±0.019	0.592 ±0.013
Mistral	0.569 ±0.023	0.612 ±0.014	0.523 ±0.017	0.529 ±0.012	0.625 ±0.021	0.439 ±0.022	0.663 ±0.036	0.555 ±0.012
Evaluation based on probability distribution: POSIX.								
gemma	3.18	5.99	6.59	7.19	1.51	0.47	0.59	0.74
Qwen	1.99	2.43	1.16	1.39	2.41	2.16	2.05	2.47
Llama	0.048	0.11	0.085	0.043	0.076	0.065	0.084	0.16
Mistral	3.43	1.19	0.92	2.57	1.79	3.31	3.97	3.12
PRC (Ours): Evaluation based on prompt vectors.								
gemma	1.90	5.16	6.48	8.11	1.52	0.38	0.42	0.44
Qwen	1.96	2.39	0.97	1.12	2.26	2.27	1.97	2.56
Llama	0.041	0.034	0.088	0.038	0.079	0.064	0.012	0.069
Mistral	3.39	0.47	0.38	2.29	0.29	3.34	4.18	3.00

Table 1: Evaluation of robustness on BBH dataset. CJ., DQ., DL., FF., HB., LD., RC., TE. are short for causal judgment, disambiguation QA, Dyck languages, formal fallacies syllogisms negation, hyperbaton, logical deduction five objects, reasoning about colored objects, salient translation error detection, respectively. `gemma`, `Qwen`, `Llama`, `Mistral` are abbreviations of LLMs investigated.

Task	CJ.	DQ.	DL.	FF.	HB.	LD.	RC.	TE.
Evaluation based on task performance: mean and standard deviation of task accuracy.								
gemma	0.488 ±0.016	0.482 ±0.083	0.556 ±0.016	0.507 ±0.011	0.526 ±0.023	0.456 ±0.067	0.541 ±0.064	0.411 ±0.053
Qwen	0.519 ±0.018	0.567 ±0.072	0.512 ±0.021	0.557 ±0.038	0.632 ±0.059	0.446 ±0.037	0.597 ±0.028	0.476 ±0.058
Llama	0.448 ±0.044	0.498 ±0.107	0.518 ±0.011	0.503 ±0.044	0.643 ±0.099	0.477 ±0.044	0.506 ±0.074	0.496 ±0.038
Mistral	0.539 ±0.033	0.505 ±0.057	0.531 ±0.009	0.509 ±0.026	0.578 ±0.066	0.494 ±0.039	0.551 ±0.118	0.446 ±0.052
Evaluation based on probability distribution: POSIX.								
gemma	12.94	11.87	7.24	9.45	12.33	12.23	13.68	14.78
Qwen	3.73	3.29	1.11	0.37	3.33	2.73	3.80	3.24
Llama	0.16	0.21	0.12	0.094	0.55	0.17	0.17	0.17
Mistral	5.33	4.49	1.07	4.65	6.00	3.99	5.04	5.30
PRC (Ours): Evaluation based on prompt vectors.								
gemma	1167.41	1818.50	464.51	1199.25	759.17	1235.86	2592.19	6951.98
Qwen	211.44	232.40	47.81	2.99	95.81	195.28	505.63	785.02
Llama	10.32	5.39	8.81	4.51	10.87	5.41	0.75	18.41
Mistral	192.11	230.91	83.39	152.06	162.11	141.99	471.17	1019.80

Table 2: Evaluation of controllability on BBH dataset.

as shown in the first row of Figure 3.[1] We can see that $\psi_{\mathcal{M},\mathcal{P}_i}$ exhibits a strong correlation with $\mathbf{Rob}_{local}(\mathcal{M}, \mathcal{P}_i; \mathbf{PV})$, with consistently high R^2 values, indicating that PV is well suited for robustness evaluation.

5.2 Suitability for Controllability Evaluation

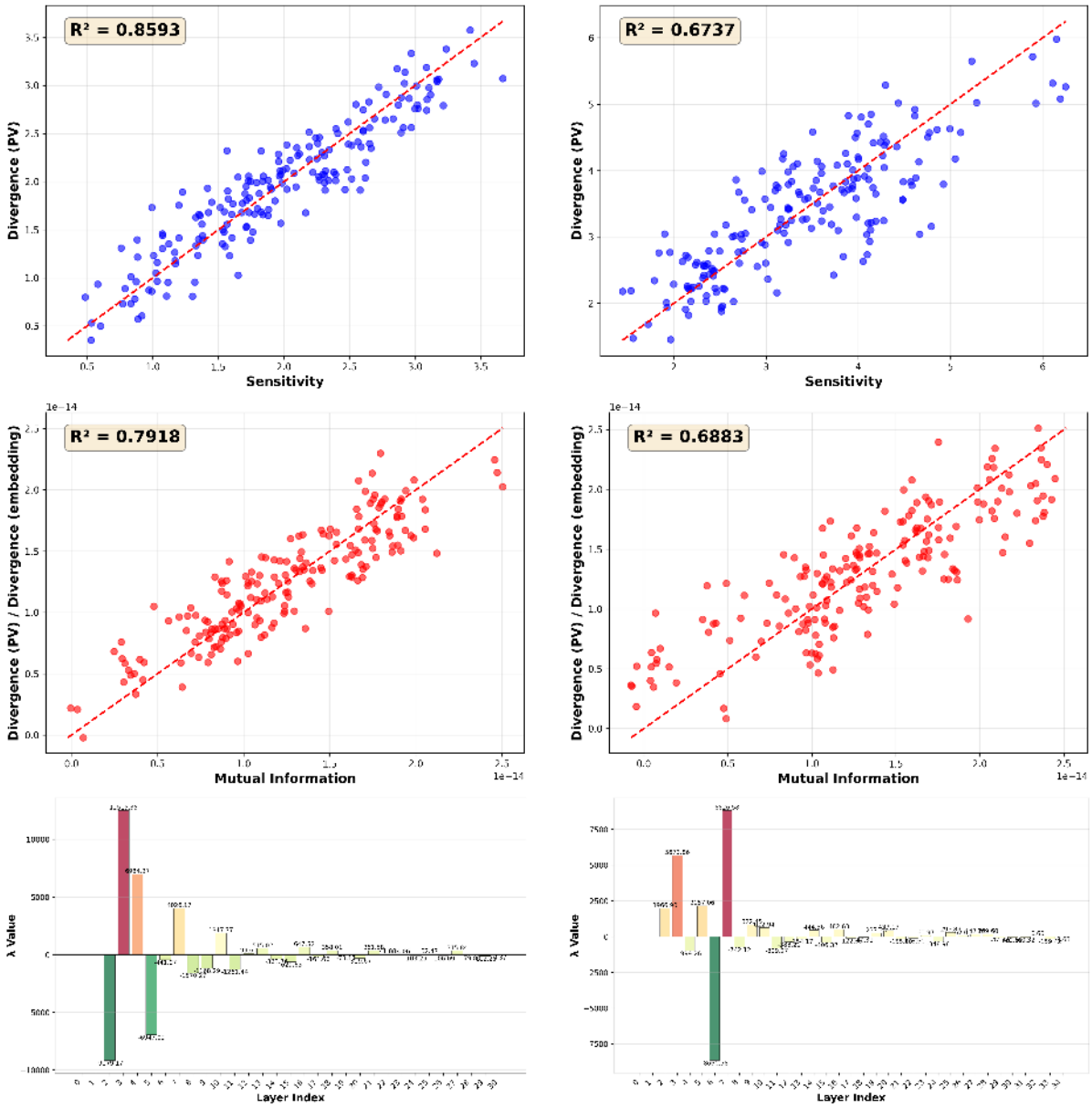
Given a dataset $\mathfrak{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_N\}$ where each $\mathcal{P}_i = \{p_i^1, \ldots, p_i^{P_i}\}$ consists of incremental prompts, we adopt average mutual information as a proxy for the difference in encoded information: $I(\mathcal{P}_i) = \frac{1}{P_i}\sum_{j=1}^{P_i-1}\Big(H(p_i^j) + H(p_i^{j+1}) - H(p_i^j, p_i^{j+1})\Big)$, where $H(p)$ denotes the entropy of prompt p. We analyze the correlation between $I(\mathcal{P}_i)$ and $\mathbf{Con}_{local}(\mathcal{M}, \mathcal{P}_i; \mathbf{PV})$, as shown in the second row of Figure 3. Again, we observe strong correlations with high R^2 values, supporting the suitability of PV for controllability evaluation.

5.3 Efficiency of Evaluation

The computational efficiency of PRC encompasses both the one-time offline calibration (Stage 2) and the end-to-end evaluation throughput (Stage 4).

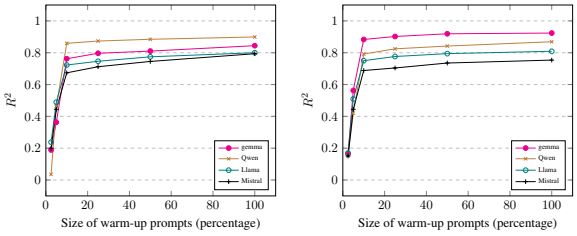
Offline Warm-up Overhead. Stage 2 optimizes layer coefficients λ_l over a warm-up subset $\mathfrak{P}_W$. Efficient optimization requires minimizing: (1) the selected layer count $|\mathcal{L}(\mathfrak{P})|$, and (2) the warm-up size $|\mathfrak{P}_W|$. As shown in the third row of Figure 3, optimizing across all layers reveals that only a small subset of λ_l exhibits large magnitudes. Thus,

[1] Refer Appendix C for more results.



(a) Qwen (b) Mistral

Figure 3: Verification the feasibility of PV on the causal judgment task of BBH dataset. First row: correlation between $\psi_{\mathcal{M},\mathcal{P}_i}$ and $\mathbf{Rob}_{local}(\mathcal{M}, \mathcal{P}_i; \mathbf{PV})$. Second row: correlation between $I(\mathcal{P}_i)$ and $\mathbf{Con}_{local}(\mathcal{M}, \mathcal{P}_i; \mathbf{PV})$. Third row: λ_l where $l \in [0, L+1]$.



(a) $R^2(\psi_{\mathcal{M},\mathcal{P}_i}, \mathbf{Rob}_{local})$ (b) $R^2(I(\mathcal{P}_i), \mathbf{Con}_{local})$

Figure 4: Correlations with different values of $|\mathfrak{P}_W|$, represented as the percentage of $|\mathfrak{P}|$.

setting $|\mathcal{L}(\mathfrak{P})| = \lceil\sqrt{L+2}\rceil$ is sufficient. Furthermore, as shown in Figure 4, marginal R^2 gains diminish once $|\mathfrak{P}_W| > 0.1|\mathfrak{P}|$. Accordingly, we set $|\mathfrak{P}_W| = \lceil 0.1|\mathfrak{P}|\rceil$, incurring minimal overhead.

Empirical Wall-Clock Speedup. To ground real-world efficiency, we measured end-to-end execution time across the entire BBH suite (Table 3). PRC completes the benchmark in merely **7.2 minutes**. Compared against POSIX ($\sim$ 2.3 hours), the strictly orthogonal baseline for output distribution sensitivity, PRC achieves a 19.16$\times$ **speedup**. This

Method	Need Output?	Total Time	Speedup
Task Accuracy	Yes	∼11.4 hours	1.0×
POSIX	Yes	∼2.3 hours	5.0×
PRC (Ours)	**No**	**0.12 hours**	95.0×

Table 3: End-to-end wall-clock execution time for evaluating the entire BBH benchmark suite.

speedup stems from completely bypassing the vocabulary projection matrix multiplication required by output-based metrics across all variations.

5.4 Discussion on Metric Deviations

While PRC aligns closely with POSIX across most model-task settings, notable metric deviations occasionally emerge (e.g., Llama exhibiting $R^2 = 0.5424$ on MMLU). A vital question arises: *What does it qualitatively mean when internal prompt vectors drift while the final output remains stable?*

Literature Context: Information Funneling. This phenomenon is grounded in the representation engineering literature on *information funneling* and layer-wise *residual stream saturation* (Clark et al., 2019; Voita et al., 2019). Intermediate transformer layers are highly sensitive to superficial token order and contextual formatting. However, as representations propagate into deeper layers, the residual stream often converges onto a low-dimensional, semantically grounded task manifold, stabilizing prior to the vocabulary projection layer.

Dual Mechanistic Hypotheses. Rather than indicating a framework failure, deviations between PRC and output metrics reveal two plausible mechanistic interpretations:

- **Hypothesized Latent Volatility:** Under heavy instruction alignment, squashing top-layers may mask underlying internal instability, rendering the model's output stability precarious under temperature shifts or context extension.
- **Robust Semantic Funneling:** Alternatively, intermediate layer drift represents benign context-tracking features that later layers successfully aggregate into a robust, invariant task representation.

In either scenario, PRC offers unique diagnostic value by capturing fine-grained **representational volatility** prior to output decoding, providing a nuanced view of internal model behavior that output-matching inherently obscures.

Interpreting Cross-Model R^2 Variations: Coupling vs. Dampening. The variation in correlation strength across models is not an evaluation flaw, but rather an insightful diagnostic property that quantifies a model's *Representational Dampening Capacity*:

- **High R^2 (Direct Coupling):** Demonstrates a tight, direct pipeline where intermediate prompt vector shifts propagate linearly into downstream logit distributions.
- **Lower R^2 (Dampening Capacity):** Reveals structural decoupling, where extensive post-training alignment functions as a non-linear cushioning filter, actively absorbing internal semantic shifts to maintain output stability.

Without an internal probe like PRC, evaluators cannot determine whether observed output consistency stems from genuine representational invariance or output-layer dampening.

6 Conclusions

In this work, we address the fragmented terminologies surrouding LLM prompt sensitivity by cohesively formalizing prompt robustness and controllability. We introduce the prompt vector, a compact representation derived from internal model states, enabling the unified PRC framework to assess LLM behaviors on prompt variations efficiently, without relying on costly autoregressive output generation. Extensive experiments across multiple datasets and LLMs demonstrate that PRC strongly aligns with traditional output-based metrics while offering deterministic, deep-structural insights and substantial efficiency gains, showing clear differences across LLMs. Our framework provides a principled diagnostic lens for analyzing LLMs' internal behaviors, complementing existing output-based evaluations.

Limitations

Despite its advantages, this work has several limitations that suggest directions for future research.

First, the prompt vector relies on access to internal model representations, which limits the applicability of PRC to open-weight or white-box LLMs. Extending the framework to black-box settings, for example by approximating PV using observable signals, remains an open challenge.

Second, although PV eliminates the need for output generation during evaluation, its construction requires a warm-up set $\mathfrak{P}_W$ to determine layer

selection and the learning of layer coefficients λ_l. While we empirically show that $\mathfrak{P}_W$ can be small, this step still introduces additional hyperparameter configurations, design choices, and computational overhead. In particular, tuning the layer coefficients adds complexity that could potentially limit the scalability and seamless reproducibility of the framework across entirely new model architectures and diverse evaluation settings.

Third, our construction of equivalent and incremental prompts focuses on controlled manipulations such as example reordering and example addition or removal. More complex prompt variations – such as semantic paraphrasing, stylistic changes, or incorporation of external knowledge – are not explored, and may exhibit different robustness–controllability characteristics.

Fourth, our empirical analysis is limited to a subset of instruction-tuned LLMs and benchmark tasks. While the observed trends are consistent across models and datasets, broader evaluation across more architectures, languages, and application domains would further strengthen the generality of our conclusions.

Fifth, the prompt vector (PV) is extracted from the hidden representation at the final token position of the input prompt. While this effectively captures the model's static semantic integration of few-shot context prior to generation, a single-position vector cannot trace dynamic, multi-step autoregressive reasoning processes (e.g., explicit Chain-of-Thought tokens). Evaluating representation-level robustness across multi-step reasoning chains remains an important direction for future extension.

Finally, a potential trade-off exists regarding the construction of prompt sets and the framework's efficiency. While PRC significantly improves evaluation efficiency by eliminating the bottleneck of LLM output generation, defining whether two completely distinct prompts encode the "same information" (e.g., highly ambiguous semantic paraphrasing in zero-shot settings) might require human verification. Relying on a human-curated set of prompts would incur substantial manual effort, potentially offsetting the computational efficiency gained. However, in practical applications—including our experimental setup—this bottleneck is effectively mitigated by utilizing automated, programmatic perturbations (e.g., shuffling in-context examples, or systematically adding/removing examples). Much like automated typo or template perturbations in prior output-based evaluations (Chatterjee et al., 2024), these heuristic rules ensure strict information preservation or incrementality without any manual cost. Thus, the human-in-the-loop requirement is confined only to complex, unstructured semantic variations, leaving PRC highly efficient for scalable, automated benchmarking.

References

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208.

Tanise Ceron, Neele Falk, Ana Barić, Dmitry Nikolaev, and Sebastian Padó. 2024. Beyond prompt brittleness: Evaluating the reliability and consistency of political worldviews in llms. *Transactions of the Association for Computational Linguistics*, 12:1378–1400.

Anwoy Chatterjee, H S V N S Kowndinya Renduchintala, Sumit Bhatia, and Tanmoy Chakraborty. 2024. POSIX: A prompt sensitivity index for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14550–14565, Miami, Florida, USA. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Federico Errica, Davide Sanvito, Giuseppe Siracusano, and Roberto Bifulco. 2025. What did I do wrong? quantifying LLMs' sensitivity and consistency to prompt engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1543–1558, Albuquerque, New Mexico. Association for Computational Linguistics.

Louie Giray. 2023. Prompt engineering with chatgpt: a guide for academic writers. *Annals of biomedical engineering*, 51(12):2629–2633.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 3 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Parsa Hejabi, Elnaz Rahmati, Alireza S. Ziabari, and Morteza Dehghani. 2025. Flip-flop consistency: Unsupervised training for robustness to prompt perturbations in llms. *Computing Research Repository*, arXiv:2510.14242.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Yang Liu and Chenhui Chu. 2026. Understanding the prompt sensitivity. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 44362–44388, San Diego, California, United States. Association for Computational Linguistics.

Sheng Lu, Hendrik Schuff, and Iryna Gurevych. 2024. How are prompts different in terms of sensitivity? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5833–5856, Mexico City, Mexico. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Hiroshi Matsuda, Chunpeng Ma, and Masayuki Asahara. 2025. Step-by-step instructions and a simple tabular output format improve the dependency parsing accuracy of LLMs. In *Proceedings of the 18th International Conference on Parsing Technologies (IWPT, SyntaxFest 2025)*, pages 11–19, Ljubljana, Slovenia. Association for Computational Linguistics.

Julian Minder, Kevin Du, Niklas Stoehr, Giovanni Monea, Chris Wendler, Robert West, and Ryan Cotterell. 2025. Controllable context sensitivity and the knob behind it. In *The Thirteenth International Conference on Learning Representations*.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Ayana Niwa, Masahiro Kaneko, and Kentaro Inui. 2025. Rectifying belief space via unlearning to harness LLMs' reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25060–25075, Vienna, Austria. Association for Computational Linguistics.

Yao Qiang, Subhrangshu Nandi, Ninareh Mehrabi, Greg Ver Steeg, Anoop Kumar, Anna Rumshisky, and Aram Galstyan. 2024. Prompt perturbation consistency learning for robust language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1357–1370, St. Julian's, Malta. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

Mikhail Seleznyov, Mikhail Chaichuk, Gleb Ershov, Alexander Panchenko, Elena Tutubalina, and Oleg Somov. 2025. When punctuation matters: A large-scale comparison of prompt robustness methods for LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20370–20385, Suzhou, China. Association for Computational Linguistics.

Jiuding Sun, Chantal Shaib, and Byron C Wallace. 2024. Evaluating the zero-shot robustness of instruction-tuned language models. In *The Twelfth International Conference on Learning Representations*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 16 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6287–6310, Bangkok, Thailand. Association for Computational Linguistics.

Jiaqi Wang, Enze Shi, Sigang Yu, Zihao Wu, Huawen Hu, Chong Ma, Haixing Dai, Qiushi Yang, Yanqing Kang, Jinru Wu, and 1 others. 2025. Prompt engineering for healthcare: Methodologies and applications. *Meta-Radiology*, page 100190.

Zhaofeng Wu, Michihiro Yasunaga, Andrew Cohen, Yoon Kim, Asli Celikyilmaz, and Marjan Ghazvininejad. 2025. reWordBench: Benchmarking and improving the robustness of reward models with transformed inputs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3383–3409, Suzhou, China. Association for Computational Linguistics.

```
{task_description}

# Examples
{few_shot_examples}

# Question
{input_instance}
```

Table 4: Prompt template followed by all tasks.

A Prompts for Each Task

As introduced in §4.1, our goal is not to achieve state-of-the-art task performance, but rather to evaluate robustness and controllability under equivalent and incremental prompt variations. Accordingly, we design prompts in a way that allows precise control over the information they contain.

Each prompt consists of a brief task description, followed by three examples (without chain-of-thought reasoning), and a target input instance. For all tasks, the prompt templates follow the format shown in Table 4.

The `{task_description}` varies across tasks, which is shown in Table 5.

For BBH dataset, the `{few_shot_examples}` is a concatenation of examples with the format of `Q: {question}\nA: {answer}\n`, and the `{input_instance}` follows the format of `Q: {question}\nA:` . For MMLU dataset, the `{few_shot_examples}` is a concatenation of examples with the format of `Question: {question}\nAnswer: {answer}\n`, and the `{input_instance}` follows the format of `Question: {question}\nAnswer:` .

Few-shot examples are randomly selected from the dataset, and are ensured to be different from the input instance. For the MMLU dataset, all few-shot examples are from the same domain as the input instance.

B Robustness and Controllability of LLMs on MMLU dataset

Besides BBH dataset, we also evaluate robustness and controllability on MMLU dataset, and compare three representative approaches. Results are shown in Table 6. We can see that the observations on BBH dataset (reported in §4.2) remain hold on MMLU dataset. Robustness and controllability cannot be reliably evaluated using task performance alone. There is a strong correlation between POSIX and PRC. Controllability varies substantially across LLMs.

Task	{task_description}
causal judgment	Answer questions about causal attribution.
disambiguation QA	Clarify the meaning of sentences with ambiguous pronouns.
Dyck languages	Correctly close a Dyck-n word.
formal fallacies syllogisms negation	Distinguish deductively valid arguments from formal fallacies.
hyperbaton	Order adjectives correctly in English sentences.
logical deduction five objects	A logical deduction task which requires deducing the order of a sequence of objects.
reasoning about colored objects	Answer extremely simple questions about the colors of objects on a surface.
salient translation error detection	Detect the type of error in an English translation of a German source sentence.
MMLU	You are a helpful assistant that can answer multiple-choice questions.

Table 5: The {task_description} of all tasks.

	Robustness			Controllability		
Measure	Accuracy	POSIX	PRC	Accuracy	POSIX	PRC
gemma	0.453 ±0.016	5.26	5.76	0.456 ±0.066	12.11	1797.97
Qwen	0.521 ±0.025	2.33	2.26	0.509 ±0.054	2.09	134.74
Llama	0.507 ±0.021	0.064	0.045	0.518 ±0.106	0.11	6.82
Mistral	0.523 ±0.012	1.45	0.87	0.532 ±0.076	4.60	267.32

Table 6: Evaluation of robustness and controllability on MMLU dataset. gemma, Qwen, Llama, Mistral are abbreviations of LLMs investigated.

C Detailed Results for Discussion of Feasibility of Prompt Vector

This section report detailed results for discussing the feasibility of prompt vectors.

Figure 8 shows the correlation between $\psi_{\mathcal{M},\mathcal{P}_i}$ and $\mathbf{Rob}_{local}(\mathcal{M}, \mathcal{P}_i; \mathbf{PV})$ on all eight tasks of BBH dataset, and Figure 5 shows the result on MMLU dataset. All these results show strong correlation with high R^2 values, indicating that PV is well suited for robustness evaluation.

Figure 9 shows the correlation between $I(\mathcal{P}_i)$ and $\mathbf{Con}_{local}(\mathcal{M}, \mathcal{P}_i; \mathbf{PV})$ on all eight tasks of BBH dataset, and Figure 6 shows the result on MMLU dataset. All these results show strong correlation with high R^2 values, indicating that PV is well suited for controllability evaluation.

Figure 10 shows the magnitudes of optimal coefficients λ_l on all eight tasks of BBH dataset, and Figure 7 shows the result on MMLU dataset. In most cases, only a small subset of λ_l exhibits large absolute values, suggesting that most layers contribute minimally, indicating the efficiency of evaluation.

D Discussions on Potential Risks

This paper proposed a framework for evaluating robustness and controllability of LLMs on prompt variations.

The focus is the issue of evaluation, rather than text generation. Therefore, our research has no potential risks.

E Discussions on Dataset

We have reported statistics of datasets in Section 4.1. All datasets are public benchmark datasets that are widely used in NLP research. They have been confirmed that no personally identifying information or offensive content are contained.

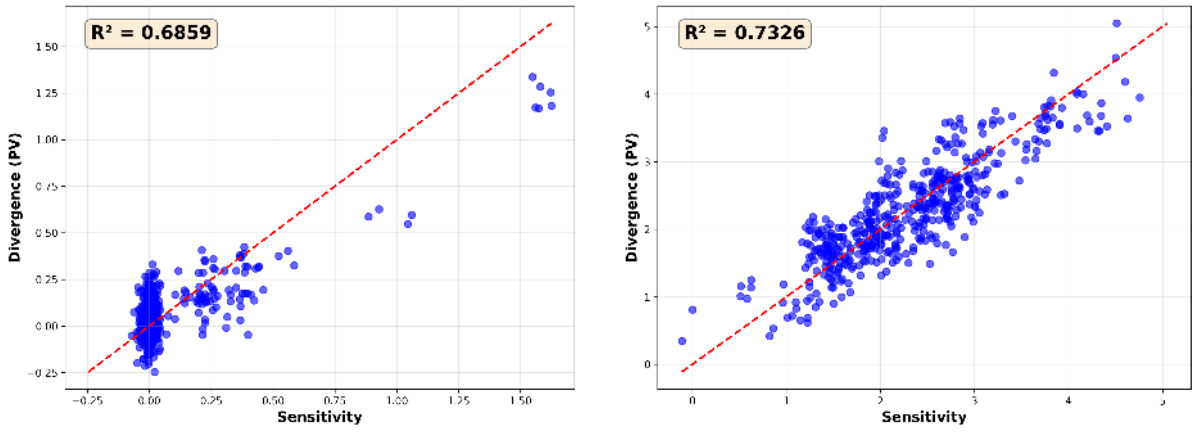
F Discussions on Computational Experiments

We investigated four public LLMs that are accessible using the Python library `transformers`. All experiments are conducted on an instance of Google Cloud Platform with a GPU of Nvidia A100 (40G).

For using LLMs, all hyper-parameters are chosen as the default values. For evaluation using PRC, two hyper-parameters need to be tuned: $|\mathcal{L}(\mathfrak{P})|$ and $|\mathfrak{P}_W|$. The values used for experiments are reported in Section 5.3.

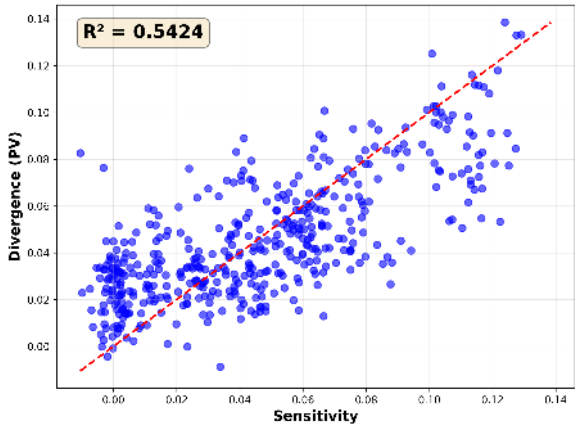
The main results are reported in Table 1 and Table 2. For the evaluations based on task performances, we evaluated the means and standard deviations over five runs. For POSIX and PRC, we found that the results vary little over different runs, so we only reported representative values.

Wall-Clock Time Evaluation Protocol. The empirical wall-clock times reported in Table 3 were recorded on a single instance equipped with four NVIDIA L4 GPU (96GB VRAM). Execution times

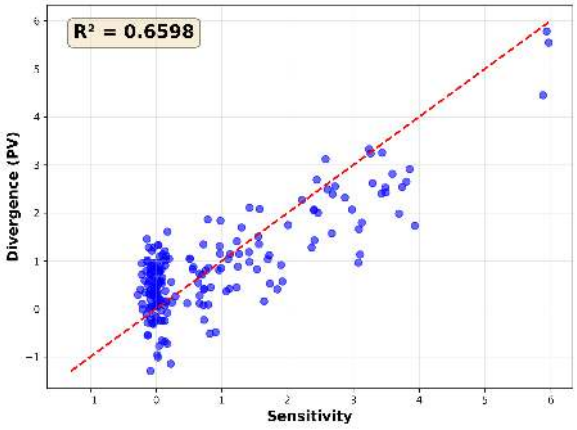


(a) gemma

(b) Qwen

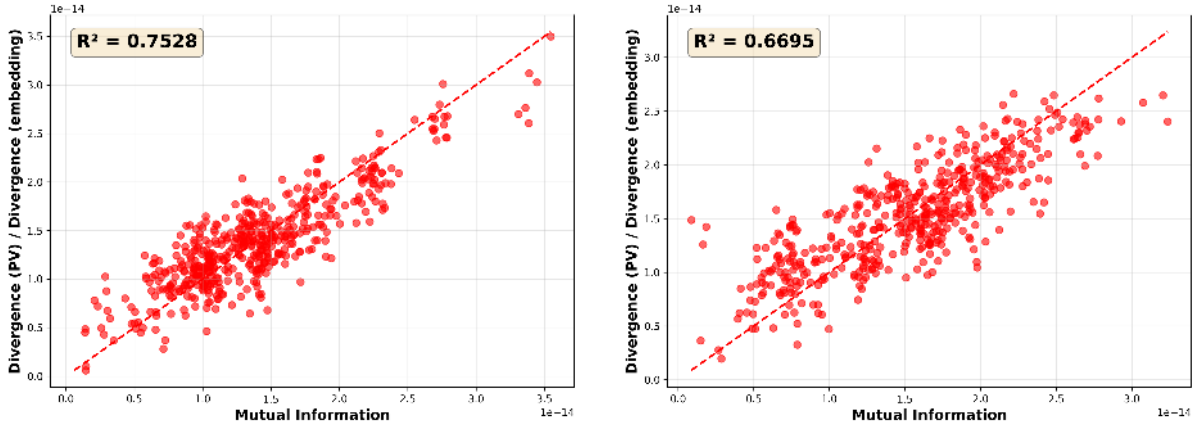


(c) Llama



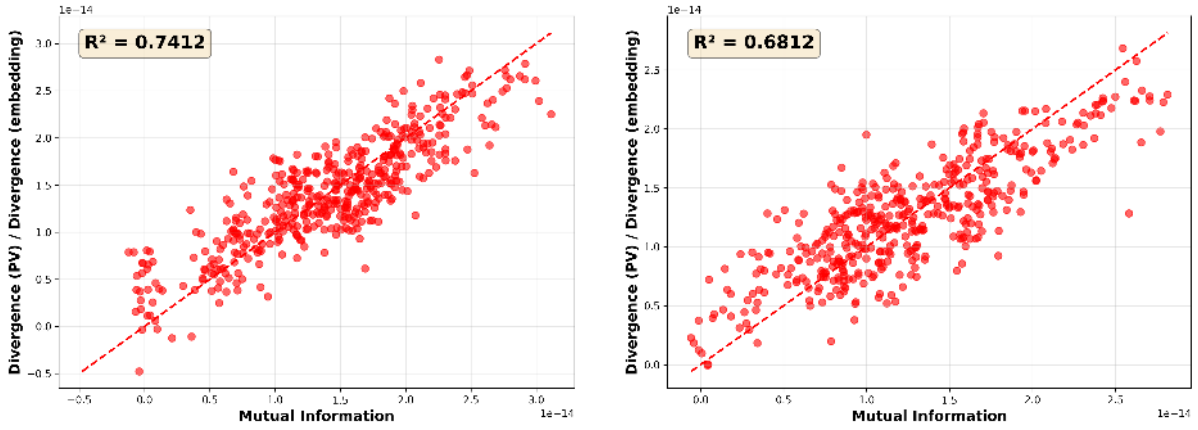
(d) Mistral

Figure 5: Verification suitability of PV for robustness evaluation, evaluated by correlation between $\psi_{\mathcal{M},\mathcal{P}_i}$ and $\mathbf{Rob}_{local}(\mathcal{M}, \mathcal{P}_i; \mathbf{PV})$. Dataset: MMLU.



(a) gemma

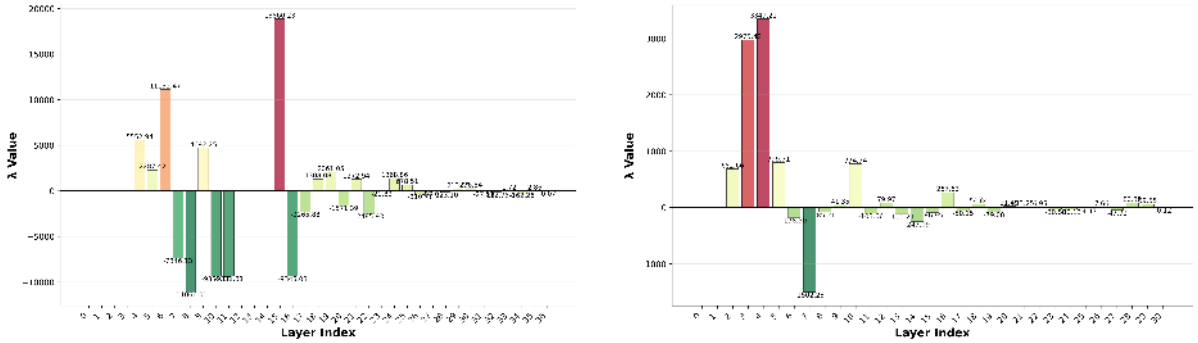
(b) Qwen



(c) Llama

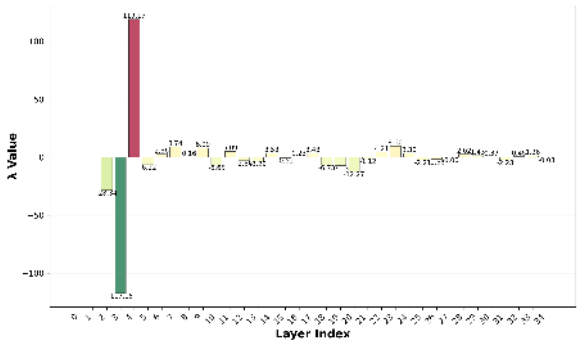
(d) Mistral

Figure 6: Verification suitability of PV for robustness evaluation, evaluated by correlation between $I(\mathcal{P}_i)$ and $\mathbf{Con}_{local}(\mathcal{M}, \mathcal{P}_i; \mathbf{PV})$. Dataset: MMLU.

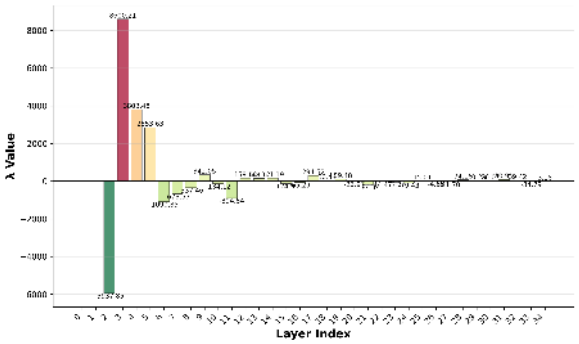


(a) gemma

(b) Qwen



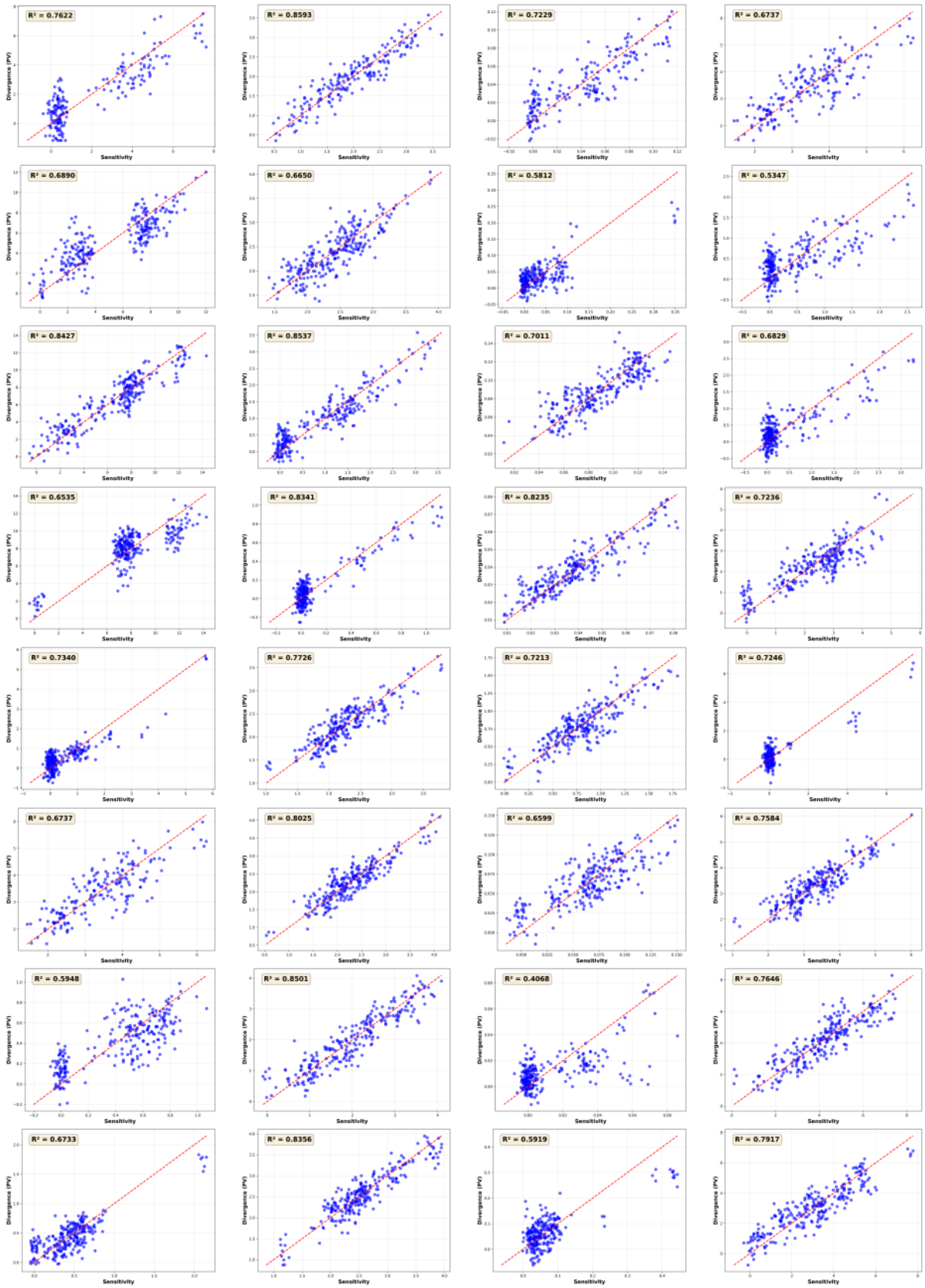
(c) Llama



(d) Mistral

Figure 7: Verification evaluation efficiency of PV. Values of optimal λ_l are reported, where $l \in [0, L+1]$. Dataset: MMLU.

reflect the complete end-to-end processing pipeline, including data loading, forward propagation across all 8 BBH tasks and 4 model families, and metric computation. For PRC, the reported time incorporates both the one-time Stage 2 coefficient optimization over $\mathfrak{P}_W$ (10% subset) and Stage 3–4 representation extractions over the remaining dataset.



(a) gemma (b) Qwen (c) Llama (d) Mistral

Figure 8: Verification suitability of PV for robustness evaluation, evaluated by correlation between $\psi_{\mathcal{M},\mathcal{P}_i}$ and $\mathbf{Rob}_{local}(\mathcal{M}, \mathcal{P}_i; \mathbf{PV})$. Dataset: BBH. Tasks (up to down): causal judgment, disambiguation QA, Dyck languages, formal fallacies syllogisms negation, hyperbaton, logical deduction five objects, reasoning about colored objects, and salient translation error detection.



(a) gemma (b) Qwen (c) Llama (d) Mistral

Figure 9: Verification suitability of PV for robustness evaluation, evaluated by correlation between $I(\mathcal{P}_i)$ and $\mathbf{Con}_{local}(\mathcal{M}, \mathcal{P}_i; \mathbf{PV})$. Dataset: BBH. Tasks (up to down): causal judgment, disambiguation QA, Dyck languages, formal fallacies syllogisms negation, hyperbaton, logical deduction five objects, reasoning about colored objects, and salient translation error detection.



(a) gemma (b) Qwen (c) Llama (d) Mistral

Figure 10: Verification evaluation efficiency of PV. Values of optimal λ_l are reported, where $l \in [0, L+1]$. Dataset: BBH. Tasks (up to down): causal judgment, disambiguation QA, Dyck languages, formal fallacies syllogisms negation, hyperbaton, logical deduction five objects, reasoning about colored objects, and salient translation error detection.