FROM RAWTOKENS TO PHYSSUMMARY: PROBING TEXT INTERFACES FOR INVERSE 1D PDE PARAMETER ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text is the de facto glue layer between numerical data and large language models (LLMs), yet interface design can silently determine whether an LLM-based scientific pipeline succeeds or fails. We present a controlled benchmark isolating text-interface effects on LLM-based inverse parameter estimation for 1D heat and advection equations, comparing RAWTOKENS (quantized grid serialization) and PHYSSUMMARY (physics-informed descriptors) across zero-shot learning (ZSL), in-context learning (ICL), and supervised fine-tuning (SFT) regimes with four LLMs. Rather than replacing conventional solvers, we derive practical design rules: on heat, closed-API models benefit from PHYSSUMMARY under prompting, while open-weight models and SFT are more robust with RAWTOKENS. Evaluating *in-range rate* alongside root mean square error (RMSE) exposes three failure modes—prompt-visible constant copying, SFT collapse on compressed features, and non-monotonic shot scaling—invisible to accuracy metrics alone. SFT on RAWTOKENS narrows the gap to float-grid baselines despite operating on quantized text. These findings provide interface-aware design guidance for any scientific workflow routing numerical data through an LLM.

1 INTRODUCTION AND RELATED WORK

Large language models (LLMs) are increasingly applied beyond natural language to scientific tasks that involve continuous numerical data—time series forecasting (Song et al., 2024; Gruver et al., 2023) and in-context numerical regression (Garg et al., 2023; Serrano et al., 2025). A natural next frontier is *inverse* partial differential equation (PDE) problems, where the goal is to infer latent physical parameters (e.g., diffusivity, wave speed) from observed spatio-temporal fields. Traditionally, inverse PDE estimation relies on specialized numerical solvers, physics-informed neural networks (PINNs) (Raissi et al., 2019), or neural operators (FNOs, DeepONets, NIOs) (Li et al., 2023; Lu et al., 2021; Molinaro et al., 2023) that are task-specific and require retraining for each new PDE. PDE foundation models (McCabe et al., 2024; Hao et al., 2024; Herde et al., 2024; Shen et al., 2024), equation-conditioned solvers (Zhou et al., 2025b; Bhatnagar et al., 2025), and text-conditioned solvers (Zhou et al., 2025a; Lorsung & Farimani, 2025) show that Transformer pretraining can enable cross-PDE generalization, while LLMs have also been applied to PDE discovery (Luo et al., 2025); yet all these efforts focus on *forward* prediction or symbolic identification. Whether LLMs can perform *inverse* estimation—and how the choice of textual input representation shapes their accuracy and reliability—remains unexplored.

In any LLM-based scientific workflow, text is the de facto glue layer: numerical data from simulations, sensors, or databases must be serialized into tokens before the model can process it. This serialization is not neutral. How numbers are tokenized can fundamentally affect model behavior (Golkar et al., 2024), and different text interfaces expose different information, introduce different quantization artifacts, and interact differently with a model's pretraining distribution. The resulting *representation gap* creates a concrete risk: predictions that parse correctly but are physically implausible can go undetected by standard accuracy metrics, producing silent failures in downstream pipelines. Interface design is therefore a reliability-critical concern—not merely an engineering convenience—for any system that routes numerical data through an LLM. Understanding these failure modes in a controlled setting is a prerequisite for deploying LLMs in more complex scientific

pipelines, where interface-related errors would be confounded with modeling errors and far harder to diagnose.

Scope and positioning. We do not propose LLMs as replacements for conventional PDE solvers; supervised baselines trained on continuous floats substantially outperform all LLM configurations in our study. Instead, this work is a *controlled empirical study* that isolates text-interface effects on LLM-based inverse estimation: we vary the interface while holding the underlying data, models, and evaluation fixed. The value lies in the failure-mode taxonomy and practical design rules that emerge, which are directly applicable to any LLM-based scientific pipeline. The failure modes catalogued here—particularly silent validity failures invisible to RMSE—would be far harder to diagnose in multi-parameter or coupled-PDE settings, where interface errors confound modeling errors; characterising them in a controlled setting is therefore a prerequisite for scaling up.

We compare two textual interfaces derived from the same underlying field data: (i) RAWTOKENS, which serializes quantized spatio-temporal grids, and (ii) PHYSSUMMARY, which provides compact, physics-informed descriptors. Our contributions are:

1. **A controlled benchmark that isolates text-interface effects.** By varying only the interface while fixing data, models, and evaluation, we provide a reproducible testbed for studying how text serialization shapes LLM-based inverse estimation across two 1D PDEs, four models, and three learning regimes (ZSL, ICL, SFT; §3.4–3.8).
2. **Reliability analysis and failure-mode taxonomy beyond RMSE.** We identify three concrete failure modes—prompt-visible constant copying, SFT collapse on compressed features, and non-monotonic shot scaling—that are invisible to accuracy metrics alone but exposed by evaluating in-range rate alongside RMSE (§3.6). These failures are not predictable from standard metrics: one configuration achieves RMSE≈ 0.12 yet 0% in-range (every prediction physically invalid), while another maintains 96% in-range yet collapses to near-constant outputs.
3. **Representation-gap quantification.** SFT on RAWTOKENS narrows the gap to float-grid MLP baselines (heat RMSE 0.011 vs. 0.002) despite operating on quantized text—closer than one might expect given the lossy tokenization, yet confirming a persistent information bottleneck that bounds what current text interfaces can recover (§3.7).

2 METHODS

Our goal is to understand how text interfaces affect LLM performance on inverse PDE estimation. We study inverse heat diffusion from simulated 1D trajectories $u(t, x)$ on a fixed spatio-temporal grid, where the diffusion coefficient κ is the unknown parameter. We cast inference as structured generation: given a textual encoding of the trajectory, the model outputs a JSON object (e.g., `{"kappa": 0.05}`). We compare two interfaces for the same observations: (i) RAWTOKENS, which serializes an affinely quantized grid, and (ii) PHYSSUMMARY, which provides physics-inspired descriptors (energy decay, gradient proxies, and spectral cues). We evaluate zero-shot (ZSL), in-context learning (ICL), and supervised fine-tuning (SFT), reporting root mean square error (RMSE) on κ together with JSON parsing rate and in-range rate. Finally, we apply the same interface comparison to a 1D advection task to test whether trends persist across PDE types. The method overview is shown in Figure 1.

Study design overview. Table 1 summarises the axes of the study. Heat diffusion serves as the primary analysis (main text); advection is a replication task, and oracle/full shot-scaling results appear in the Appendix.

RAWTOKENS. We adopt the affine quantization and token-serialization scheme of Bao et al. (2025), but use global range statistics to ensure a consistent physical scale across trajectories. Specifically, we compute $u_{\min}$ and $u_{\max}$ once from the dataset, and map each floating-point value $u_{t,x}^{(i)}$ to an integer $Q_{t,x}^{(i)}$ in a fixed set $\mathcal{Z} = [I_{\min}, I_{\max}]$. We then serialize the quantized grid row-wise over time into a text sequence. Each integer is rendered with a fixed-width "v-scheme" token (e.g., `v0491`) to avoid token-boundary ambiguity and encourage the tokenizer to treat each value as an atomic unit. For the details, see Appendix A.2.

Table 1: Study design matrix. Primary analysis is in the main text; secondary results are in the Appendix.

Axis	Levels
PDEs	1D heat (primary), 1D advection (replication)
Interfaces	RAWTOKENS, PHYSSUMMARY
Regimes	ZSL (k=0), ICL ($k \in \{2, 4, 8\}$), SFT (open-weight only)
Models	Qwen3-4B, Qwen3-30B (open); GPT-4o-mini, Gemini-2.5-Flash (closed)
Metrics	RMSE (accuracy), In-range rate (reliability)
Appendix-only: oracle interface, full shot-scaling tables, parse-rate breakdown	

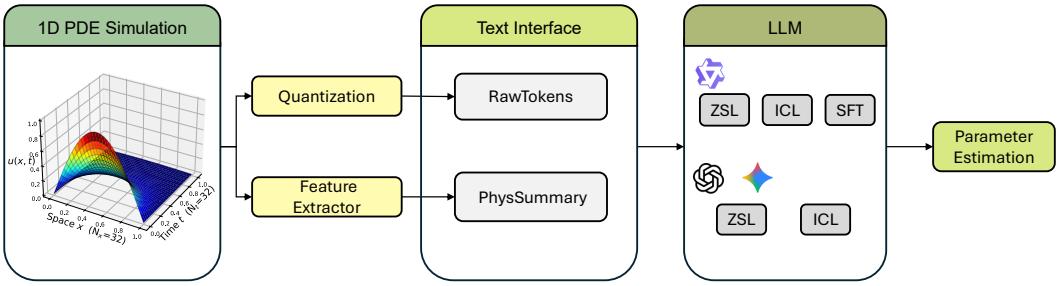


Figure 1: Method overview: (1) simulate 1D PDE trajectories, (2) form either RAWTOKENS prompts (quantized grid serialization) or PHYSSUMMARY prompts (physics-informed descriptors), and (3) evaluate LLMs to predict the unknown PDE parameter (e.g., diffusivity κ).

PHYSSUMMARY. As a compact alternative to RAWTOKENS, we compute deterministic, physics-motivated descriptors from each trajectory and render them as a structured prompt. This representation is task-conditioned: for dissipative dynamics (1D heat) we summarize decay and frequency attenuation, while for transport dynamics (1D advection) we summarize translation and conservation diagnostics. Concretely, the PHYSSUMMARY includes:

- **Heat (dissipation)**: energy decay over time, gradient-based dissipation proxies, and spectral attenuation of low-order spatial modes.
- **Advection (transport)**: mass/energy stability diagnostics and a coarse shift estimate based on normalized periodic cross-correlation between consecutive time slices.

Full feature definitions and equations are in Appendix A.3.1 (heat) and Appendix A.3.3 (advection). Throughout the main paper, PHYSSUMMARY refers exclusively to the non-oracle variant; oracle-style features that leak the target parameter are evaluated as a diagnostic upper bound in the Appendix only (Appendix A.3.2, A.3.4).

Interface tradeoff. By design, the two interfaces differ in token budget ($\sim$1,000 tokens for a serialized grid vs. a handful of numeric fields), information bottleneck (near-lossless grid vs. lossy summary statistics), and exposure to prompt-visible numerical constants that models may copy rather than interpret. Characterising how this tradeoff interacts with model family and learning regime is the central goal of this study; the representation-gap baselines in §3.7 measure the bottleneck's magnitude directly.

LLMs and learning regimes. We evaluate open-weight instruction-tuned LLMs QWEN3-4B-Instruct[1] and QWEN3-30B-Instruct[2] (Yang et al., 2025), and closed-source API models GPT-4O-MINI (OpenAI et al., 2024) and GEMINI-2.5-FLASH (Comanici et al., 2025). All models are evaluated in ZSL and ICL settings. For ICL, we vary the number of demonstrations $k \in \{2, 4, 8\}$ to assess sensitivity to shot count; demonstrations are selected to cover the parameter range. For open-weight models, we additionally report SFT via Low-Rank Adaptation (LoRA) on task-specific training data

[1] huggingface.co/Qwen/Qwen3-4B-Instruct-2507
[2] huggingface.co/Qwen/Qwen3-30B-A3B-Instruct-2507

(Hu et al., 2021). We use identical prompt templates across models and deterministic decoding, instructing the model to output only the JSON object.

Non-LLM baselines. To contextualize LLM performance, we include ridge regression and a small multilayer perceptron (MLP) trained directly on the original floating-point trajectory, which avoids tokenization/quantization effects and serves as a conventional inverse-modeling reference.

Implementation details and hyperparameters are in Appendix A.5.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Tasks and data. We evaluate on two 1D inverse PDE tasks. *Heat diffusion* (primary): recover diffusivity $\kappa \in [0.01, 0.2]$ from $u(t, x)$ on a 32×32 grid ($N_t = 32,\ N_x = 32$). *Advection* (replication): recover velocity $c \in [0.1, 0.8]$ from the same grid geometry with periodic boundary conditions. For each task we generate 10,000 noise-free synthetic trajectories with randomised initial conditions (Eq. 1); 9,000 for training and 1,000 for testing. Full dataset details are in Appendix A.

Models, interfaces, and regimes. We use the interfaces, models, and learning regimes defined in §2 (Table 1), with identical prompt templates and deterministic decoding throughout. Closed-API model outputs may vary over time due to provider updates; we report results as best-effort snapshots (see Appendix A.5 for model identifiers and access dates). An oracle interface variant is evaluated as a diagnostic upper bound in the Appendix only (Tables 10, 12).

3.2 EVALUATION METRICS

We report two primary metrics. *RMSE* (root mean square error) is computed over the N_{parsed} predictions that the model returned as valid JSON, measuring regression accuracy on the inferred parameter. Unparseable outputs are excluded from RMSE but penalised by the second metric: *In-range rate* is the fraction of *all* N_{total} test samples whose prediction both parses successfully and falls within the valid parameter range; this metric jointly penalises parse failures and out-of-range outputs, serving as a holistic reliability indicator. A model with low RMSE but low in-range rate is unreliable despite appearing accurate. Parsing rate and the in-range rate are reported in the Appendix (formal definitions in Appendix A.4).

3.3 SUMMARY OF KEY FINDINGS

Three overarching findings emerge from the experiments that follow. First, interface choice interacts strongly with model family and learning regime, producing predictable but PDE-specific patterns (§3.4–3.8). Second, evaluating in-range rate alongside RMSE reveals silent failures that accuracy metrics alone cannot detect (§3.6). Third, we identify three concrete failure modes—prompt-visible constant copying, SFT collapse on compressed features, and non-monotonic shot scaling—each tied to specific interface–regime combinations (§3.6, §3.5).

3.4 MAIN HEAT RESULTS

Table 2 summarises heat diffusion results for ZSL, ICL (k=2, chosen for overall stability), and SFT where available.

Key patterns in Table 2.

- **Interface sensitivity is model-class dependent.** Closed-API models show moderate RMSE improvements on PHYSSUMMARY (GPT-4o-mini ZSL: 0.079 vs. 0.110; Gemini at k=4: 0.054 vs. 0.088 in Table 3), while open-weight models exhibit the opposite: Qwen3-30B PHYSSUMMARY in-range drops to 34% at ZSL and 39% at k=2, compared to ≥99% on RAWTOKENS throughout.
- **SFT amplifies interface effects rather than smoothing them.** Under SFT, RAWTOKENS achieves the best RMSE for both open models (Qwen3-4B: 0.011; Qwen3-30B: 0.040), whereas

Table 2: Inverse heat diffusion (κ prediction). RMSE (computed on parsed predictions) and in-range rate (fraction of all test samples with valid, in-range output) across interfaces and regimes. **Bold**: best LLM per text interface. SFT: LoRA fine-tuning (open models only). Supervised baselines are compared in §3.7.

Model	Regime	RawTokens		PhysSummary	
		RMSE ↓	InRange %↑	RMSE ↓	InRange %↑
Qwen3-4B	ZSL	0.1167	0.0	0.1183	0.0
	ICL k=2	0.0853	98.3	**0.0712**	100.0
	SFT	**0.0112**	100.0	1.7825[†]	96.1
Qwen3-30B	ZSL	0.1551	100.0	0.1183	34.2
	ICL k=2	0.0842	99.9	0.6625[‡]	39.4
	SFT	0.0404	100.0	0.3629[§]	85.4
GPT-4o-mini	ZSL	0.1099	99.4	0.0793	90.6
	ICL k=2	0.0903	99.4	0.0812	94.5
Gemini-2.5-Flash	ZSL	0.1061	51.8	0.0930	95.1
	ICL k=2	0.0931	79.4	0.0756	86.1

[†]SFT on PhysSummary collapsed; see §3.6.
[‡]PhysSummary instability at k=2 (see §3.5).
[§]PhysSummary SFT degraded (85% in-range); see §3.6.

Table 3: Shot-scaling for 1D heat (κ). RMSE vs. number of ICL demonstrations k. Superscripts flag low in-range rates: [a] 0%, [b] <50%, [c] <95%. **Bold**: best per model (across interfaces and regimes). SFT: LoRA (open models only). Float-grid baselines: MLP = 0.0022, CNN = 0.0035.

Model	Interface	k=0	k=2	k=4	k=8	SFT
Qwen3-4B	Raw	0.117[a]	0.085	0.124	0.145	**0.011**
	Phys	0.118[a]	0.071	2.664[c]	0.099	1.783
Qwen3-30B	Raw	0.155	0.084	0.092[c]	0.155	**0.040**
	Phys	0.118[b]	0.663[b]	0.303[c]	0.302[c]	0.363[c]
GPT-4o-mini	Raw	0.110	0.090	0.120	0.128	—
	Phys	**0.079**[c]	0.081[c]	0.082	0.137	—
Gemini-2.5-Flash	Raw	0.106[c]	0.093[c]	0.088	0.122	—
	Phys	0.093	0.076[c]	**0.054**	0.096	—

PhysSummary SFT either degrades (Qwen3-30B: 0.363, 85% in-range) or collapses entirely (Qwen3-4B: 1.783, near-constant outputs despite 96% in-range; §3.6).

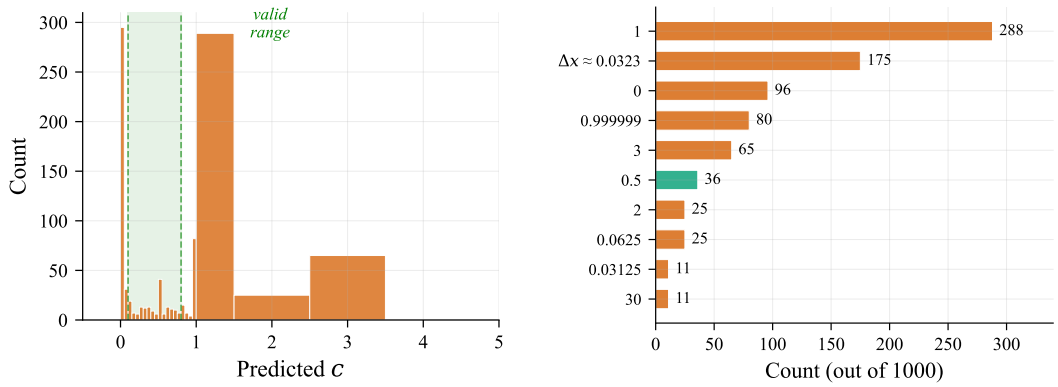
- **A representation bottleneck persists even at best.** Qwen3-4B SFT on RawTokens (RMSE = 0.011) approaches the float-grid MLP baseline (0.002) but does not close the gap, confirming that text serialization introduces information loss even for fine-tuned models (§3.7).

3.5 Shot-Scaling Analysis

Table 3 reports RMSE as a function of k for all four models on both interfaces. Several patterns emerge.

Non-monotonic shot scaling. For RawTokens, most models improve from k=0 to k=2 but degrade at k=8, suggesting that additional demonstrations dilute rather than reinforce the useful signal in long tokenized contexts. Gemini-2.5-Flash is the exception: its RawTokens RMSE improves monotonically from k=0 to k=4 before rising at k=8 (Table 3).

PhysSummary instability for open models. Both open models exhibit erratic shot-scaling on PhysSummary: Qwen3-4B collapses at k=4 before recovering at k=8, and Qwen3-30B shows a similarly unstable trajectory (Table 3). These outliers are driven by prompt-visible constant copying: the model intermittently reproduces salient numerical tokens from the PhysSummary prompt



(a) Prediction distribution

(b) Top predicted values

Figure 2: GPT-4o-mini ZSL on advection PHYSSUMMARY: prompt-visible constant copying. Green band = valid $[0.1, 0.8]$. The model copies prompt-visible constants ($\Delta x \approx 0.032$, round numbers). Only 19% in range; k=2 ICL partially overrides constant copying (in-range rises from 19% to 63%), but does not fully resolve it.

rather than producing a calibrated regression estimate. By contrast, closed models (GPT-4o-mini, Gemini-2.5-Flash) show stable PHYSSUMMARY behavior across all k.

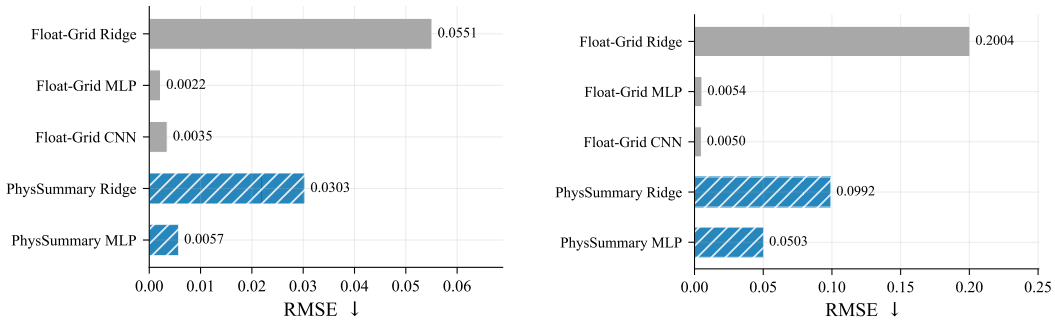
Closed models benefit from PHYSSUMMARY on heat. Gemini-2.5-Flash achieves the best prompting-only RMSE (0.054 on PHYSSUMMARY at k=4), a 39% improvement over its best RAWTOKENS result. GPT-4o-mini also favours PHYSSUMMARY in ZSL (0.079 vs. 0.110), though the gap narrows under ICL. Full k-sweep results for all models are in Appendix Table 9.

3.6 RELIABILITY AND FAILURE MODES

In-range rate highlights hidden failures. While most model–interface configurations achieve high parsing rates (≥99%), Qwen3-30B on PHYSSUMMARY is a notable exception, with parsing rates as low as 34.5% at ZSL (see Appendix Table 9). Beyond parsing, the in-range rate reveals significant additional differences in output validity. Qwen3-4B ZSL produces 0% in-range predictions on both interfaces for heat, despite reasonable RMSE values (≈0.12), because the model systematically predicts values near but outside $[\kappa_{\min}, \kappa_{\max}]$. Similarly, Gemini-2.5-Flash ZSL on RAWTOKENS yields only 52% in-range, indicating that about half its predictions exceed the valid bounds. ICL rapidly improves reliability: at k=2, Qwen3-4B jumps to 98–100% in-range.

Constant-copying failure: GPT-4o-mini on advection PHYSSUMMARY ZSL. The most striking failure mode occurs when GPT-4o-mini is asked to predict the advection speed c from PHYSSUMMARY features without demonstrations. We refer to this behavior as *prompt-visible constant copying*: the model outputs a numerical value that appears verbatim in the prompt (e.g., grid spacing Δx) rather than producing a calibrated regression estimate. This is a descriptive label for the observed output pattern, not a claim about human-like cognitive anchoring bias. Figure 2 shows that 81% of predictions cluster at a handful of prompt-visible or round-number values: c=1.0 (29%), c=Δx≈0.032 (18%), c=0 (10%), c≈1 (8%), and c=3 (7%). The value $\Delta x = 1/(N_x - 1) \approx 0.032$ appears in the prompt as a grid-spacing parameter, confirming that the model latches onto salient numerical tokens rather than producing a calibrated estimate. Because grid spacing, physical units, and normalization constants appear routinely in scientific prompts, this failure mode is a practical deployment risk, not an artifact of our specific prompt design. Only 19% of predictions fall within the valid range $[0.1, 0.8]$. This failure is substantially mitigated with as few as k=2 demonstrations (in-range rises from 19% to 63%), suggesting that ICL can partially override constant copying by calibrating the output scale, though reliability remains well below 100%.

SFT collapse on PHYSSUMMARY (heat). Qwen3-4B SFT on the PHYSSUMMARY interface for heat produces RMSE = 1.78 (vs. 0.011 on RAWTOKENS), a complete regression failure despite 96%



(a) Heat (κ) (b) Advection (c)

Figure 3: Baseline RMSE by input representation. Grey: float-grid; blue hatched: PHYSSUMMARY. Ridge benefits from physics features; MLP/CNN perform best on the full grid.

of predictions falling in range. Inspection reveals the model learns to output near-constant values close to the training-set mean. Notably, a conventional MLP trained on the *same* PHYSSUMMARY features achieves RMSE = 0.006 (§3.7), so the features themselves are sufficiently informative for supervised regression. The collapse is therefore specific to LoRA fine-tuning dynamics on the LLM, possibly reflecting shortcut learning or mean-regression behavior amplified by the low-dimensional input. This is a cautionary finding: SFT does not universally resolve interface limitations and can amplify failure modes that do not arise under prompting or conventional supervised learning.

3.7 BASELINES AND REPRESENTATION GAP

To contextualise LLM performance, we compare against conventional regression models trained on three input representations: (a) the full 32×32 float grid (1024 features), (b) the PHYSSUMMARY feature vector (22 features for heat, 7 for advection), and (c) oracle features (Appendix only).

Representation gap. Figure 3 reveals a clear representation hierarchy. For heat, the MLP on the full float grid achieves RMSE = 0.002, while the same architecture on PHYSSUMMARY features reaches 0.006—a $2.6\times$ gap that quantifies the information lost by the physics summary. Notably, Ridge regression *benefits* from PHYSSUMMARY features (0.030 vs. 0.055 on float grid), showing that the hand-crafted features provide a more informative basis for linear models. For advection, the representation gap is larger: MLP on the float grid (0.005) vs. PHYSSUMMARY features (0.050), a $9.3\times$ ratio, confirming that the advection PHYSSUMMARY features (mass deviation, lag bin) capture less of the relevant signal than the heat features. This larger bottleneck is consistent with the more severe PHYSSUMMARY instability observed under prompting on advection (§3.8).

LLMs vs. baselines. The best prompting-based LLM for heat (Gemini-2.5-Flash PHYSSUMMARY, k=4: RMSE = 0.054) is comparable to a Ridge regressor on PHYSSUMMARY features (0.030) but far from the float-grid MLP (0.002). SFT narrows this gap substantially: Qwen3-4B SFT on RAWTOKENS (RMSE = 0.011) outperforms Ridge on PHYSSUMMARY and approaches the float-grid MLP. This is notable given that the SFT model processes quantized text while the MLP operates on continuous floats.

3.8 ADVECTION: REPLICATION AND STRESS TEST

To assess whether the design rules from heat generalise, we replicate the full evaluation pipeline on 1D advection (predicting $c \in [0.1, 0.8]$), which involves fundamentally different dynamics (transport vs. dissipation) and a substantially weaker PHYSSUMMARY feature set (7 features vs. 22 for heat).

The key finding is that **heat-specific patterns do not transfer automatically**. Advection is harder across the board: the best prompting RMSE is roughly $3\times$ worse than on heat. Most strikingly, PHYSSUMMARY ZSL triggers prompt-visible constant-copying failures for *all* models on advection—including closed-API models that handle PHYSSUMMARY reliably on heat (Appendix Table 11). SFT, however, remains functional: Qwen3-4B achieves comparable RMSE on both

interfaces with 100% in-range rate, and Qwen3-30B maintains ≥99% in-range on both (though PHYSSUMMARY RMSE is 1.7× higher than RAWTOKENS)—a marked contrast with heat, where PHYSSUMMARY SFT collapses entirely. This underscores that interface choices must be validated per PDE type; a single set of design rules is insufficient. Full advection results appear in Appendix Tables 11 and 12.

3.9 NOISE ROBUSTNESS (SFT)

We evaluate SFT models on test sets with injected observation noise ($\sigma \in \{10^{-3}, 10^{-2}, 10^{-1}\}$) to test robustness beyond the noise-free training distribution. For heat, Qwen3-4B SFT on RAWTOKENS degrades gracefully from RMSE = 0.011 (clean) to ≈0.10 at σ=0.1, maintaining 100% in-range throughout. Note that Qwen3-4B PHYSSUMMARY SFT on heat already collapsed on clean data (RMSE = 1.78; §3.6), so noise comparisons for that configuration are not informative. For advection, where PHYSSUMMARY SFT succeeds on clean data (Qwen3-4B: 0.053 RAWTOKENS, 0.056 PHYSSUMMARY), both interfaces degrade comparably under noise. Figure 4 shows the full noise robustness curves for both tasks, including the Oracle (L2) diagnostic upper bound.

Practical design guidance. Synthesising across both PDEs, models, and regimes, we distil five rules of thumb for practitioners building LLM-based inverse-estimation pipelines:

1. **Default to RAWTOKENS for SFT.** Across both PDE types, RAWTOKENS SFT is the most reliable configuration—it avoids the SFT collapse observed with PHYSSUMMARY on heat and matches or exceeds it on advection.
2. **Use PHYSSUMMARY with closed-API models under prompting (on dissipative PDEs).** GPT-4o-mini and Gemini-2.5-Flash consistently benefit from physics-informed descriptors on heat, but this advantage does not transfer to advection.
3. **Prefer RAWTOKENS for open-weight models.** Open models show higher reliability and more stable shot-scaling on RAWTOKENS in both prompting and SFT regimes.
4. **Always report in-range rate alongside RMSE.** RMSE alone can hide configurations where the majority of outputs are physically invalid. In-range rate is a minimal reliability check.
5. **Validate interface choices per PDE type.** The patterns observed on heat do not automatically transfer to advection; per-task evaluation is necessary before deployment.

4 CONCLUSION AND DISCUSSION

Our controlled benchmark yields three actionable conclusions for LLM-based inverse PDE estimation. *First*, RAWTOKENS is the safer default for fine-tuning across PDE types, while PHYSSUMMARY benefits closed-API models under prompting on dissipative PDEs—but this advantage does not transfer to advection, making per-task interface validation essential. *Second*, reliability metrics beyond RMSE are indispensable: in-range rate reveals configurations where the majority of outputs are physically invalid despite low RMSE, and exposes three concrete failure modes—prompt-visible constant copying, SFT collapse on compressed features, and non-monotonic shot scaling—that accuracy metrics alone cannot detect. *Third*, SFT on RAWTOKENS narrows the gap to supervised float-grid baselines despite operating on quantized text, but a representation bottleneck persists, confirming that text serialization introduces information loss even for fine-tuned models. These findings provide interface-aware design guidance for any scientific pipeline that routes numerical data through an LLM, and the benchmark itself offers a reproducible testbed for future work on text-interface effects in scientific inverse problems.

Future work. Our study uses deterministic decoding by design to isolate interface effects from sampling variability, but this precludes uncertainty quantification. The benchmark is compatible with non-zero-temperature sampling for empirical prediction intervals, logit-based confidence extraction for open-weight models, and conformal or risk-coverage analysis for distribution-free guarantees; studying how interface choice interacts with these methods is a natural next step. A second direction is hybrid workflows: LLM estimates could serve as coarse initial guesses gated by in-range validity checks, with a conventional inverse solver refining accepted estimates to full precision—making the reliability diagnostics studied here operational components of a deployment pipeline.

Limitations. By design, our study is restricted to simple 1D single-parameter PDEs (heat, advection) with noise-free training data—a deliberate choice that isolates text-interface effects from confounding factors, though results may not transfer directly to higher-dimensional, multi-parameter, or real-world noisy settings (a limited SFT noise-robustness analysis is included in §3.9). Both open-weight models belong to the Qwen3 family; testing additional families would strengthen generality. Deterministic decoding precludes uncertainty quantification, and the PHYSSUMMARY features are hand-designed per PDE type; automating feature selection is an important direction. More broadly, text interfaces admit many design axes we do not explore—quantization resolution, feature subsets, prompt template wording, and output format variations—each of which could shift relative performance; our two-interface comparison should therefore be seen as a lower bound on the design space that merits investigation. Finally, the in-range rate uses a hard boundary and does not distinguish marginally out-of-range predictions from wildly incorrect ones; tolerance-aware variants (e.g., a soft ϵ-margin metric or coverage-based calibration) would provide a more nuanced reliability picture and are a natural extension.

REFERENCES

Jiajun Bao, Nicolas Boullé, Toni J. B. Liu, Raphaël Sarfati, and Christopher J. Earls. Text-trained llms can zero-shot extrapolate pde dynamics, revealing a three-stage in-context learning mechanism, 2025. URL https://arxiv.org/abs/2509.06322.

Rohan Bhatnagar, Ling Liang, Krish Patel, and Haizhao Yang. From equations to insights: Unraveling symbolic structures in pdes with llms, 2025. URL https://arxiv.org/abs/2503.09986.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes, 2023. URL https://arxiv.org/abs/2208.01066.

Siavash Golkar, Mariel Pettee, Michael Eickenberg, Alberto Bietti, Miles Cranmer, Geraud Krawezik, Francois Lanusse, Michael McCabe, Ruben Ohana, Liam Parker, Bruno Régaldo-Saint Blancard, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. xval: A continuous numerical tokenization for scientific language models, 2024. URL https://arxiv.org/abs/2310.02989.

Nate Gruver, Marc Anton Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=md68e8iZK1.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=X7UnDevHOM.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes, 2024. URL https://arxiv.org/abs/2405.19101.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *J. Mach. Learn. Res.*, 24(1), January 2023. ISSN 1532-4435.

Cooper Lorsung and Amir Barati Farimani. Explain like i'm five: Using llms to improve pde surrogate models with text, 2025. URL https://arxiv.org/abs/2410.01137.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5. URL http://dx.doi.org/10.1038/s42256-021-00302-5.

Xiao Luo, Changhu Wang, Yizhou Sun, and Wei Wang. How do large language models perform on PDE discovery: A coarse-to-fine perspective. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 2684–2697, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.145. URL https://aclanthology.org/2025.findings-emnlp.145/.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Multiple physics pretraining for physical surrogate models, 2024. URL https://openreview.net/forum?id=fH9eqpCcR3.

Roberto Molinaro, Yunan Yang, Björn Engquist, and Siddhartha Mishra. Neural inverse operators for solving PDE inverse problems. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25105–25139. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/molinaro23a.html.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, et al. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Louis Serrano, Armand Kassaï Koupaï, Thomas X Wang, Pierre Erbacher, and Patrick Gallinari. Zebra: In-context generative pretraining for solving parametric pdes, 2025. URL https://arxiv.org/abs/2410.03437.

Junhong Shen, Tanya Marwah, and Ameet Talwalkar. UPS: Efficiently building foundation models for PDE solving via cross-modal adaptation. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=0r9mhjRv1E.

Xingyou Song, Oscar Li, Chansoo Lee, Bangding Yang, Daiyi Peng, Sagi Perel, and Yutian Chen. Omnipred: Language models as universal regressors. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=t9c3pfrR1X.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Anthony Zhou, Zijie Li, Michael Schneier, John R Buchanan Jr, and Amir Barati Farimani. Text2PDE: Latent diffusion models for accessible physics simulation. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=Nb3a8aUGfj.

Hang Zhou, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: Pde-conditional transformers towards universal neural pde solvers, 2025b. URL https://arxiv.org/abs/2405.17527.

A APPENDIX

A.1 DATASET GENERATION

Common setup. We simulate spatio-temporal trajectories on the unit domain $(t, x) \in [0, 1] \times [0, 1]$ and construct supervised pairs $\big(\text{trajectory } u(t, x),\ \text{parameter}\big)$. Unless otherwise stated, all trajectories are sampled on a uniform grid with $N_t = 32$ time steps and $N_x = 32$ spatial nodes, with $\Delta t = 1/(N_t - 1)$ and $\Delta x = 1/(N_x - 1)$. For both PDEs, the initial condition is a randomized smooth profile formed by a finite sine expansion:

$$u_0(x) \;=\; \sum_{k=1}^{K} a_k \sin(2\pi k x + \phi_k), \tag{1}$$

where $K = 5$, $a_k \sim \mathcal{U}[-1, 1]$, and $\phi_k \sim \mathcal{U}[0, 2\pi]$ independently.

1D heat dataset. We generate 10,000 trajectories for the 1D heat equation

$$\frac{\partial u(t,x)}{\partial t} \;=\; \kappa\, \frac{\partial^2 u(t,x)}{\partial x^2}, \qquad (t,x) \in [0,1] \times [0,1], \tag{2}$$

with Dirichlet boundary conditions $u(t, 0) = u(t, 1) = 0$ and initial condition $u(0, x) = u_0(x)$. For each sample, the diffusivity κ is drawn uniformly from $[\kappa_{\min}, \kappa_{\max}] = [0.01, 0.2]$. We discretize equation 2 using backward Euler in time and a second-order finite-difference Laplacian in space. We use 9,000 trajectories for training and 1,000 for testing.

1D advection dataset. To test transfer beyond dissipative dynamics, we additionally generate 10,000 trajectories for the 1D linear advection equation

$$\frac{\partial u(t,x)}{\partial t} \;+\; c\, \frac{\partial u(t,x)}{\partial x} \;=\; 0, \qquad (t,x) \in [0,1] \times [0,1], \tag{3}$$

with periodic boundary conditions $u(t, 0) = u(t, 1)$ and the same initial-condition generator equation 1. For each sample, the constant velocity c is drawn uniformly from $[c_{\min}, c_{\max}] = [0.1, 0.8]$. We discretize equation 3 using a first-order upwind scheme. We again split the data into 9,000 training and 1,000 testing trajectories.

A.2 QUANTIZATION AND SERIALIZATION

This section provides implementation details for the RAWTOKENS representation used in §2. We compute dataset-level range statistics $(u_{\min}, u_{\max})$ once (on the training split), and map each $u_{t,x}^{(i)}$ to an integer $Q_{t,x}^{(i)} \in \mathcal{Z} = [I_{\min}, I_{\max}]$ via

$$Q_{t,x}^{(i)} = \operatorname{clip}\left(\left\lfloor I_{\min} + \frac{u_{t,x}^{(i)} - u_{\min}}{u_{\max} - u_{\min}} \left(I_{\max} - I_{\min}\right) \right\rceil,\ I_{\min},\ I_{\max} \right), \tag{4}$$

where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer and clip clamps values to $[I_{\min}, I_{\max}]$. Using global $(u_{\min}, u_{\max})$ avoids the scale ambiguity introduced by per-trajectory normalization and makes token values comparable across samples.

The quantized grid $Q^{(i)}$ is then serialized row-wise over time into a textual sequence,

$$\texttt{t=0:}\quad \text{tok}(Q_{0,0})\ \text{tok}(Q_{0,1})\ \ldots\ \texttt{t=1:}\quad \text{tok}(Q_{1,0})\ \ldots,$$

and each integer is rendered using a fixed-width tokenization scheme (e.g., `v0491`), where $\texttt{tok}(z) = \texttt{v} + \texttt{zero_pad}(z, 4)$. This "v-scheme" minimizes token-boundary ambiguity and encourages the tokenizer to treat each quantized value as an atomic unit.

A.3 PHYSICS-INFORMED FEATURE ENGINEERING

We compress the full spatio-temporal field $u(t, x) \in \mathbb{R}^{N_t \times N_x}$ into a compact set of *physics-informed summary statistics* that serve as the textual input to the language model. Two information tiers are defined:

- **Physics Summary:** features derivable from forward-model outputs alone, without access to the unknown PDE parameter.
- **Oracle-style Physics Summary:** features that embed a finite-difference proxy for the target parameter, providing a "leaky upper bound" on attainable accuracy.

A.3.1 HEAT EQUATION: PHYSICS SUMMARY

Level-1 features for the heat equation capture the *rate* and *character* of diffusive smoothing without directly exposing the diffusivity κ. The LLM receives six groups of descriptors, serialised as a single text prompt.

(a) Initial- and terminal-state statistics. We record the sample statistics of the spatial profile at the first and last time steps:

$$\boldsymbol{s}_0 = \big(\bar{u}_0,\ \sigma_{u_0},\ \min u_0,\ \max u_0\big), \qquad \boldsymbol{s}_T = \big(\bar{u}_T,\ \sigma_{u_T},\ \min u_T,\ \max u_T\big), \tag{5}$$

where $\bar{u}_t = \frac{1}{N_x}\sum_j u(t, x_j)$ and σ_{u_t} is the population standard deviation at time t. The contrast between $\boldsymbol{s}_0$ and $\boldsymbol{s}_T$ (e.g. $\sigma_{u_0} \approx 1$ vs. $\sigma_{u_T} \approx 0.01$) is a direct consequence of diffusive smoothing.

(b) Log-energy trajectory. The mean squared amplitude at time t defines the energy

$$E(t) \;=\; \frac{1}{N_x}\sum_{j=0}^{N_x-1} u(t, x_j)^2. \tag{6}$$

We sample $\log\big(E(t_k) + \epsilon\big)$ with $\epsilon = 10^{-12}$ at $n_E = 6$ equispaced fractional times $\tau_k = k/5$, $k = 0, \ldots, 5$ (i.e. $\tau \in \{0.0,\ 0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0\}$), yielding

$$\mathbf{e} = \Big\{\big(\tau_k,\ \log\big(E(t_{\mathrm{round}(\tau_k(N_t-1))}) + \epsilon\big)\big)\Big\}_{k=0}^{5}. \tag{7}$$

For a diffusion-dominated process $\log E$ decays approximately linearly in t; the slope of this decay is a monotone function of κ.

(c) Gradient quantiles. The absolute spatial gradient at time t, approximated by forward differences,

$$g_j(t) = \frac{|u(t, x_{j+1}) - u(t, x_j)|}{\Delta x}, \qquad j = 0, \ldots, N_x - 2, \tag{8}$$

is summarised through its $(q_{50},\ q_{90},\ q_{99})$ quantiles at both $t = 0$ and $t = T$:

$$\mathbf{g}_0 = \big(q_{50}^{(0)},\ q_{90}^{(0)},\ q_{99}^{(0)}\big), \qquad \mathbf{g}_T = \big(q_{50}^{(T)},\ q_{90}^{(T)},\ q_{99}^{(T)}\big). \tag{9}$$

The sharp contrast $\|\mathbf{g}_0\| \gg \|\mathbf{g}_T\|$ (typical values $\mathcal{O}(10^1)$ decaying to $\mathcal{O}(10^{-2})$) directly reflects diffusive smoothing.

(d) Spectral centroid shift. For each spatial snapshot we compute the normalised spectral centroid of the mean-subtracted profile:

$$\mathcal{C}(t) = \frac{1}{K}\,\frac{\sum_{k=0}^{K} k\,|F_k|^2}{\sum_{k=0}^{K} |F_k|^2}, \tag{10}$$

where $F_k = \mathrm{rfft}_k\big[u(t, \cdot) - \bar{u}(t)\big]$, the sum runs over the $K+1$ frequency bins returned by the real FFT, and $K = \lfloor N_x/2 \rfloor = 16$. Normalising by K maps $\mathcal{C}$ into $[0, 1]$. The pair $[\mathcal{C}(0) \to \mathcal{C}(T)]$ is reported. Because diffusion attenuates high wave numbers as $e^{-\kappa k^2 t}$, larger κ produces a greater decrease in $\mathcal{C}$.

A.3.2 HEAT EQUATION: ORACLE-STYLE PHYSICS SUMMARY

(a) Diffusivity estimate $\hat{\alpha}$. We exploit the backward-Euler semi-discrete stencil

$$u_j^{n+1} - u_j^n = \alpha \left(u_{j-1}^{n+1} - 2\, u_j^{n+1} + u_{j+1}^{n+1} \right), \qquad \alpha = \frac{\kappa\, \Delta t}{\Delta x^2}, \tag{11}$$

and define, for every interior node $j = 1, \ldots, N_x - 2$ and time step $n = 0, \ldots, N_t - 2$,

$$y_{n,j} = u_j^{n+1} - u_j^n, \qquad x_{n,j} = u_{j-1}^{n+1} - 2\, u_j^{n+1} + u_{j+1}^{n+1}. \tag{12}$$

The oracle coefficient is obtained via origin-constrained ordinary least squares over *all* interior entries (no Laplacian masking):

$$\hat{\alpha} = \frac{\sum_{n,j} x_{n,j}\, y_{n,j}}{\sum_{n,j} x_{n,j}^2}, \tag{13}$$

guarded by the condition $\sum x_{n,j}^2 > 10^{-18}$. The diffusivity is recovered via

$$\hat{\kappa} = \hat{\alpha}\, \frac{\Delta x^2}{\Delta t}. \tag{14}$$

(b) Regression diagnostics. Three quality measures accompany $\hat{\alpha}$:

- the Pearson correlation $r = \mathrm{corr}(x, y)$,
- the coefficient of determination

$$R^2 = 1 - \frac{\mathrm{SS_{res}}}{\mathrm{SS_{tot}}} = 1 - \frac{\sum (y_{n,j} - \hat{\alpha}\, x_{n,j})^2}{\sum (y_{n,j} - \bar{y})^2}, \tag{15}$$

- and the residual standard deviation $\sigma_{\mathrm{res}} = \mathrm{std}(y - \hat{\alpha}\, x)$.

On clean (noiseless) data generated by the backward Euler scheme these satisfy $r \approx R^2 \approx 1$ and $\sigma_{\mathrm{res}} \approx 0$. In the presence of observation noise, departures from these values serve as a built-in quality signal.

(c) Energy decay ratio. The oracle prompt additionally reports

$$\mathrm{energy_decay} = \frac{E(T)}{E(0) + 10^{-12}}, \tag{16}$$

where $E(t)$ is defined in equation 6. For large κ dissipation drives $E(T) \to 0$ and the ratio approaches zero.

A.3.3 ADVECTION EQUATION: PHYSICS SUMMARY

Level-1 features for advection capture conservation properties and coarse transport indicators.

(a) Mass deviation. The relative fluctuation of the spatial mean over time,

$$\delta_M = \begin{cases} \dfrac{\sigma\big(\{\bar{u}^n\}_{n=0}^{N_t-1}\big)}{|\,\overline{\bar{u}}\,| + \epsilon}, & |\,\overline{\bar{u}}\,| > \epsilon, \\ \sigma\big(\{\bar{u}^n\}\big), & \text{otherwise}, \end{cases} \qquad \overline{\bar{u}} = \frac{1}{N_t} \sum_n \bar{u}^n, \quad \epsilon = 10^{-12}, \tag{17}$$

measures numerical mass loss. For well-resolved advection $\delta_M \approx 0$.

(b) Energy variance.

$$\mathrm{energy_var} = \mathrm{Var}\big(\{E^n\}_{n=0}^{N_t-1}\big), \qquad E^n = \frac{1}{N_x} \sum_j (u_j^n)^2. \tag{18}$$

A pure translating wave preserves E^n; temporal fluctuation correlates with the severity of numerical diffusion at higher CFL numbers and hence carries indirect information about c.

(c) Cross-correlation peak strength. For each consecutive pair (u^n, u^{n+1}), both profiles are mean-subtracted and the optimal integer lag is found via

$$\ell_n^* = \arg\max_{\ell \in [-L_{\max},\, L_{\max}]} \frac{\sum_{j=0}^{N_x-1} \tilde{u}_j^n \; \tilde{u}_{(j+\ell) \bmod N_x}^{n+1}}{\|\tilde{u}^n\|_2 \; \|\tilde{u}^{n+1}\|_2}, \tag{19}$$

where $\tilde{u}^n = u^n - \bar{u}^n$, $L_{\max} = \lfloor 0.5\, N_x \rfloor = 16$, and pairs with $\|\tilde{u}^n\|_2 < 10^{-8}$ are skipped. The *correlation peak strength* $\bar{\rho} = \frac{1}{|\mathcal{T}|} \sum_{n \in \mathcal{T}} \rho_n^*$ (where ρ_n^* is the maximum correlation at lag ℓ_n^*) is reported. Values near unity indicate a clean translating pattern.

(d) Lag bin and shift direction. The mean lag $\bar{\ell}$ is deliberately coarsened to prevent trivial inversion of $c = \ell\, \Delta x / \Delta t$. Its absolute value is binned via thresholds $\{0.5,\, 1.5,\, 2.5,\, 3.5\}$ into the discrete set $\{0,\, 1,\, 2,\, 3,\, \geq 4\}$. The sign is reported as "right" ($\bar{\ell} > 0.5$), "left" ($\bar{\ell} < -0.5$), or "ambiguous".

A.3.4 Advection Equation: Oracle-style Physics Summary

The continuous-valued mean lag is converted to a velocity estimate without binning:

$$\hat{c} = \bar{\ell}\, \frac{\Delta x}{\Delta t}, \qquad \sigma_{\hat{c}} = \mathrm{std}(\{\ell_n^*\})\, \frac{\Delta x}{\Delta t}. \tag{20}$$

Because the upwind scheme propagates information by exactly $\mathrm{CFL} = c\, \Delta t / \Delta x$ grid cells per step, the integer-lag estimator $\hat{c}$ is quantised to multiples of $\Delta x / \Delta t$ and therefore constitutes only a first-order approximation to c.

A.3.5 Prompt construction and examples

Chat messages (ZSL vs. ICL). Each query to the LLM is formatted as chat messages. In ZSL, we use two messages: `[system, user]`. In ICL with k demonstrations, we prepend k in-context `(user, assistant)` pairs: `[system, (user, assistant)`$\times k$`, user]`. The system message specifies the task and enforces JSON-only output, while the user message contains one of the three serialized data views.

Prompt templates. Table 4 summarises the field layout for each problem and view. Numerical values are formatted as signed six-decimal floats (e.g., `+0.032258`). Oracle-style Physics Summary appends an oracle block to the Physics Summary fields and is used only as a leaky upper bound.

System prompt (1D Heat)

```
You are an expert in solving inverse problems for the 1-D heat
  equation.
Given the provided input descriptors, estimate the diffusion
  coefficient kappa.
Return ONLY a JSON object: {"kappa": <float>}. Do not include any
  other text.
```

System prompt (1D Advection)

```
You are an expert in solving inverse problems for the 1-D advection
  equation.
Given the provided input descriptors, estimate the advection speed c
  .
Return ONLY a JSON object: {"c": <float>}. Do not include any other
  text.
```

Table 4: Prompt field layout for each problem and view. Oracle-style PHYSSUMMARY (L2) includes all PHYSSUMMARY fields plus the oracle block.

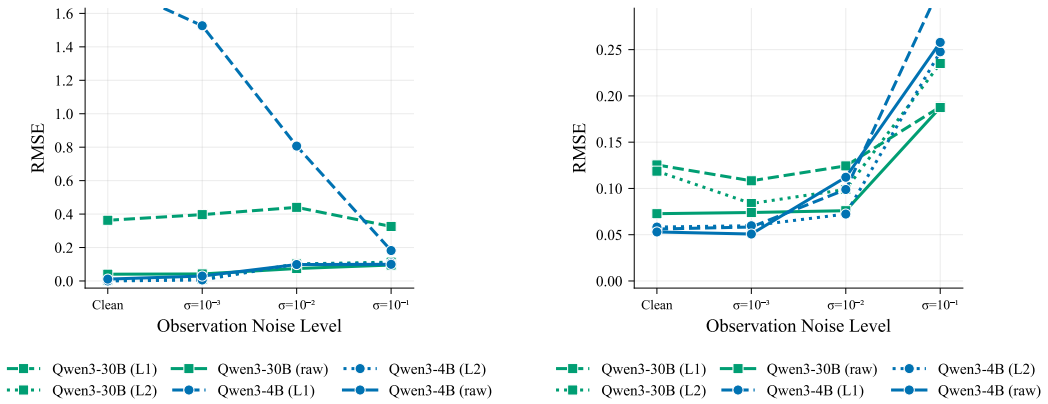
Field	1D Heat	1D Advection
Header	PDE=heat1d; scheme=backward_euler; bc=dirichlet; Grid=(32t,32x)	PDE=advection1d; scheme=upwind; bc=periodic; Grid=(...)
Domain / grid	L=...; T=...	dx=...; dt=...; (optional) L=...; T=...
Physics Summary		
Summary stats	u0_stats=(...); uT_stats=(...)	mass_dev=...; energy_var=...
Dynamics	logE_frac=[(...), ..., (...)]	corr_peak_strength=...
Spectral / shift	grad_abs_q_t0=[...]; grad_abs_q_tT=[...]; spec_centroid=[...]	lag_bin=...; shift_dir=...
Oracle-style Physics Summary(leaky upper bound) — appended		
Oracle block	alpha_hat=...; corr=...; r2=...; resid_std=...; energy_decay=...	c_hat=...; c_hat_std=...
Oracle hint	Hint(ORACLE): alpha_hat ≈ kappa · dt / dx^2	(none)
Output instruction	Return ONLY JSON {"kappa": <float>}	Return ONLY JSON {"c": <float>}

User prompt example: RAWTOKENS (1D Heat)

```
PDE=heat1d; scheme=backward_euler; bc=dirichlet; Grid=(32t,32x)
Encoding: quantized tokens vXXXX in [150,850]
Task: Return ONLY JSON {"kappa": <float>}

t=0:  v0491 v0375 v0412 ... v0448 v0394 v0491
t=1:  v0491 v0431 v0428 ... v0465 v0449 v0491
...   (time steps omitted)
t=31: v0491 v0489 v0488 ... v0497 v0494 v0491

kappa=
```

User prompt example: PHYSSUMMARY-L1 (1D Heat)

```
PDE=heat1d; scheme=backward_euler; bc=dirichlet; Grid=(32t,32x)
L=1.000000; T=1.000000
u0_stats=(mean=+0.001400, std=+1.004900, min=-1.980800, max
  =+2.276900)
uT_stats=(mean=+0.016800, std=+0.008900, min=+0.000000, max
  =+0.027400)
logE_frac=[(0.00,+0.009900), (0.20,-4.588000), ..., (1.00,-7.924100)
  ]
grad_abs_q_t0=[+18.384100, +38.002600, +50.334800]
grad_abs_q_tT=[+0.062000, +0.083900, +0.089900]
spec_centroid=[0.235600 -> 0.066400]
Task: Return ONLY JSON {"kappa": <float>}

kappa=
```



(a) Heat (κ) (b) Advection (c)

Figure 4: SFT noise robustness. RMSE vs. observation noise σ for Qwen3-4B and Qwen3-30B on RawTokens (solid), PhysSummary (dashed), and Oracle L2 (dotted; diagnostic upper bound). Models trained on clean data only. All models achieve 100% parsing across all noise levels.

User prompt example: Oracle PhysSummary (1D Heat)

```
PDE=heat1d; scheme=backward_euler; bc=dirichlet
dx=+0.032258; dt=+0.032258
ORACLE_FEATURES (leaky upper bound):
alpha_hat=+4.218605; corr=+1.000000; r2=+1.000000; resid_std=+1.931e
  -08
energy_decay=+0.000047
Hint(ORACLE): alpha_hat $\approx$ kappa $\cdot$ dt / dx^2
Task: Return ONLY JSON {"kappa": <float>}

kappa=
```

A.4 Evaluation Metrics

We formally define the three evaluation metrics used throughout the paper. Let θ denote the target PDE parameter (κ for heat diffusion, c for advection), $\hat{\theta}^{(i)}$ the model's prediction for test sample i, and N_{total} the number of test samples.

Root Mean Square Error (RMSE). RMSE measures the regression accuracy of the model on successfully parsed predictions:

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{parsed}}} \sum_{i=1}^{N_{\text{parsed}}} \left(\hat{\theta}^{(i)} - \theta^{(i)}\right)^2}, \tag{21}$$

where $N_{\text{parsed}} \leq N_{\text{total}}$ is the count of samples for which the model output was successfully parsed into a valid JSON response (e.g., `{"kappa": <float>}` or `{"c": <float>}`). By computing RMSE only on parsed samples, we isolate estimation accuracy from formatting errors. Lower values indicate better accuracy.

Parsing Rate (Parse %). The parsing rate quantifies format compliance—the fraction of outputs that can be decoded into a valid JSON response:

$$\text{Parse } \% = \frac{N_{\text{parsed}}}{N_{\text{total}}} \times 100. \tag{22}$$

A low parsing rate indicates "format fragility": the model may produce useful numeric estimates but fails to express them through the required output format.

In-Range Rate (InRange %). The in-range rate is our primary reliability metric, measuring the fraction of *all* test samples whose prediction falls within the valid parameter range:

$$\text{InRange }\% = \frac{\left|\{i : \hat{\theta}^{(i)} \in [\theta_{\min},\, \theta_{\max}]\}\right|}{N_{\text{total}}} \times 100, \tag{23}$$

where $[\theta_{\min}, \theta_{\max}]$ is the valid parameter range ($[0.01, 0.2]$ for κ; $[0.1, 0.8]$ for c). Samples that fail to parse are counted as out-of-range, so this metric simultaneously penalises parse failures *and* physically implausible predictions. A model may achieve low RMSE (on parsed outputs) but poor in-range rate if many predictions are syntactically valid yet fall outside the plausible domain.

Metric conventions. In the main paper (§3.2), we report RMSE and InRange % as the two primary metrics. The appendix tables additionally report Parse % for diagnostic completeness (see Tables 9–12).

A.5 Implementation Details

A.5.1 LLM Models

Table 5 summarises the language models used in our experiments. Open-weight models are from the Qwen3 family; closed-API models are accessed through the OpenRouter API.

Table 5: Language models evaluated. "Active params" refers to the parameters used per forward pass (relevant for MoE architectures).

Model	Total / Active Params	Type	Model ID
Qwen3-4B	4B / 4B	Open-weight	`Qwen3-4B-Instruct-2507`
Qwen3-30B-A3B	30B / 3B	Open-weight (MoE)	`Qwen3-30B-A3B-Instruct-2507`
GPT-4o-mini	Undisclosed	Closed API	`openai/gpt-4o-mini`
Gemini-2.5-Flash	Undisclosed	Closed API	`google/gemini-2.5-flash`

A.5.2 Inference Hyperparameters

Table 6 lists the inference settings for ZSL and ICL evaluation. All models use deterministic decoding (`temperature` = 0) and identical prompt templates. For ICL, demonstrations are selected using a *range-based* strategy: k examples are chosen to uniformly cover the target parameter range, and the same fixed set is reused across all test samples.

Table 6: Inference hyperparameters for LLM evaluation. "Max tokens" is the maximum number of new tokens generated per sample.

Hyperparameter	Open-weight (HF)	Closed API
Temperature	0	0
Max new tokens	32	64
Batch size	16 (4B) / 4 (30B)	10
Precision	BF16	—
Chat template	Yes	—
Seed	42	42*
*Best-effort; closed-API providers may change outputs over time.		
In-context learning settings		
Shot counts (k)	{0, 2, 4, 8}	
Shot strategy	Range (uniform parameter coverage)	
Shots per sample	No (fixed demonstration set)	

A.5.3 LoRA SFT Training Hyperparameters

Table 7 details the LoRA fine-tuning configuration. Both models share the same LoRA architecture and optimiser settings; Qwen3-30B uses QLoRA and a smaller per-device batch size to fit in GPU memory.

Table 7: LoRA SFT training hyperparameters. Effective batch size = per-device batch size × gradient accumulation steps.

Hyperparameter	Qwen3-4B	Qwen3-30B-A3B
LoRA configuration		
Rank (r)	64	64
Alpha (α)	16	16
Dropout	0.05	0.05
Target modules	`q/k/v/o/gate/up/down_proj`	
Bias	None	
Training		
Epochs	3	3
Per-device batch size	4	2
Gradient accumulation	4	8
Effective batch size	16	16
Learning rate	2×10^{-4}	2×10^{-4}
Optimizer	AdamW	
LR scheduler	Cosine with warmup	
Warmup ratio	0.05	0.05
Weight decay	0.01	0.01
Gradient checkpointing	Yes	Yes
Precision & memory		
Training dtype	BF16	BF16
4-bit quantization	No	Yes (QLoRA)
Max sequence length (Raw)	6144	6144
Max sequence length (L1/L2)	2048	2048
Seed	0	0

A.6 Baseline Models

To contextualise LLM performance, we train conventional regression models on three input representations: (a) the full 32×32 float grid (1024 features), (b) the L1 PhysSummary feature vector (22 features for heat, 7 for advection), and (c) the L2 oracle features (diagnostic only). Table 8 summarises the architecture and hyperparameters for each baseline model. All models use seed 42; neural baselines (MLP, CNN) use early stopping on validation RMSE.

Table 8: Baseline model architectures and training hyperparameters. "Float-grid" models operate on the flattened 32×32 trajectory; "PhysSummary" models operate on the L1 feature vector. CNN is applicable only to float-grid inputs.

Model	Input	Architecture & Hyperparameters
Float-grid baselines (1024 features)		
Mean	Grid	Constant predictor (training-set mean)
Ridge	Grid	Linear regression with PCA (d=128), α=1.0
MLP	Grid	Hidden layers $[512, 256, 128]$, ReLU, dropout 0.0; AdamW (lr 10^{-3}, wd 10^{-4}), batch 128, 200 epochs, patience 20
CNN	Grid	3 conv blocks $[16, 32, 64]$ channels (Conv2d→BatchNorm→ReLU) → Global Avg Pool → Linear(1); AdamW (lr 10^{-3}, wd 10^{-4}), batch 128, 200 epochs, patience 20
PhysSummary *baselines (L1 features: 22 for heat, 7 for advection)*		
Mean	L1	Constant predictor (training-set mean)
Ridge	L1	Linear regression (no PCA), α=1.0
MLP	L1	Hidden layers $[256, 128, 64]$, ReLU, dropout 0.1; AdamW (lr 10^{-3}, wd 10^{-4}), batch 128, 300 epochs, patience 30
Common settings		
Optimizer: AdamW; Scheduler: ReduceLROnPlateau (factor 0.5); Loss: MSE; AMP: enabled; Seed: 42		

A.7 FULL EXPERIMENTAL RESULTS

Tables 9–12 report the complete shot-scaling results for all models, interfaces, and regimes on both tasks. The L2 (oracle) interface is shown separately as a diagnostic upper bound and must *not* be interpreted as a competing method.

Table 9: Heat (κ prediction): full shot-scaling results across all models and shot counts. RMSE is computed on all parsed predictions (lower is better). Parse % and In-Range % are reported for completeness.

Model	k	RAWTOKENS			PHYSSUMMARY		
		RMSE ↓	Parse %	In-Range %	RMSE ↓	Parse %	In-Range %
Qwen3-4B	0	0.1167	100.0	0.0	0.1183	100.0	0.0
	2	0.0853	100.0	98.3	0.0712	100.0	100.0
	4	0.1243	100.0	100.0	2.6640	99.7	65.4
	8	0.1447	100.0	100.0	0.0990	100.0	100.0
	SFT	0.0112	100.0	100.0	1.7825	100.0	96.1
Qwen3-30B	0	0.1551	100.0	100.0	0.1183	34.5	34.2
	2	0.0842	100.0	99.9	0.6625	83.2	39.4
	4	0.0919	100.0	91.2	0.3027	87.7	83.8
	8	0.1546	100.0	100.0	0.3023	99.1	89.5
	SFT	0.0404	100.0	100.0	0.3629	99.5	85.4
GPT-4o-mini	0	0.1099	100.0	99.4	0.0793	100.0	90.6
	2	0.0903	100.0	99.4	0.0812	100.0	94.5
	4	0.1204	100.0	99.9	0.0816	100.0	100.0
	8	0.1276	100.0	100.0	0.1367	100.0	100.0
Gemini-2.5-Flash	0	0.1061	100.0	51.8	0.0930	100.0	95.1
	2	0.0931	100.0	79.4	0.0756	100.0	86.1
	4	0.0880	100.0	98.8	0.0539	100.0	99.9
	8	0.1215	100.0	99.8	0.0959	100.0	99.7
Baselines (float grid, supervised)							
Mean	–	0.0553	100.0	100.0	0.0553	100.0	100.0
Ridge	–	0.0551	100.0	100.0	0.0551	100.0	100.0
MLP	–	0.0022	100.0	100.0	0.0022	100.0	100.0
CNN	–	0.0035	100.0	100.0	0.0035	100.0	100.0

Table 10: Heat (κ): diagnostic upper bound using the leaky L2/Oracle interface. **Not a valid comparison**—included for analysis only. The L2 interface embeds a finite-difference proxy for κ, making inversion near-trivial with sufficient demonstrations.

Model	k	ORACLE (L2)		
		RMSE ↓	Parse %	In-Range %
Qwen3-4B	0	3.4816	100.0	12.3
	2	0.0114	100.0	100.0
	4	0.0042	100.0	100.0
	8	0.0083	100.0	94.5
	SFT	0.0000	100.0	100.0
Qwen3-30B	0	3.4745	100.0	8.2
	2	3.3412	100.0	53.1
	4	0.0005	100.0	100.0
	8	0.0002	100.0	100.0
	SFT	0.0040	100.0	100.0
GPT-4o-mini	0	3.3469	100.0	7.6
	2	0.0029	100.0	100.0
	4	0.0021	100.0	100.0
	8	0.0019	100.0	100.0
Gemini-2.5-Flash	0	1.2427	100.0	30.3
	2	0.0001	100.0	100.0
	4	0.0001	100.0	100.0
	8	0.0000	100.0	100.0

Table 11: Advection (c prediction): full shot-scaling results across all models and shot counts. Layout mirrors Table 9.

Model	k	RawTokens			PhysSummary		
		RMSE ↓	Parse %	In-Range %	RMSE ↓	Parse %	In-Range %
Qwen3-4B	0	0.4014	100.0	55.2	0.5926	100.0	0.0
	2	0.3241	100.0	95.8	0.3119	100.0	85.4
	4	0.2808	100.0	99.6	0.2031	100.0	75.8
	8	0.4817	100.0	0.0	0.3466	100.0	71.7
	SFT	0.0530	100.0	100.0	0.0561	100.0	100.0
Qwen3-30B	0	0.2650	100.0	87.9	0.5926	100.0	0.0
	2	0.3094	100.0	100.0	0.2926	100.0	84.6
	4	0.3045	100.0	71.9	0.1537	100.0	85.9
	8	0.4887	100.0	0.0	0.4966	100.0	3.8
	SFT	0.0727	100.0	100.0	0.1254	100.0	99.8
GPT-4o-mini	0	0.2129	100.0	99.3	2.8962	100.0	19.3
	2	0.2774	100.0	96.1	0.3939	100.0	63.3
	4	0.2576	100.0	99.2	0.2169	100.0	80.2
	8	0.4098	100.0	100.0	0.4507	100.0	20.1
Gemini-2.5-Flash	0	0.3127	100.0	67.4	0.6021	100.0	0.0
	2	0.3154	100.0	94.3	0.3011	100.0	84.2
	4	0.3881	100.0	19.2	0.1817	100.0	72.2
	8	0.4715	100.0	6.0	0.2579	100.0	22.4
Baselines (float grid, supervised)							
Mean	–	0.2001	100.0	100.0	0.2001	100.0	100.0
Ridge	–	0.2004	100.0	100.0	0.2004	100.0	100.0
MLP	–	0.0054	100.0	100.0	0.0054	100.0	100.0
CNN	–	0.0050	100.0	100.0	0.0050	100.0	100.0

Table 12: Advection (c): diagnostic upper bound using the leaky L2/Oracle interface. **Not a valid comparison**—included for analysis only. Unlike heat, the advection L2 features provide only a coarse estimate of c, so the oracle advantage is much smaller.

Model	k	Oracle (L2)		
		RMSE ↓	Parse %	In-Range %
Qwen3-4B	0	0.3381	100.0	0.0
	2	0.4123	100.0	53.4
	4	0.1967	100.0	68.3
	8	0.2128	100.0	58.2
	SFT	0.0583	100.0	100.0
Qwen3-30B	0	0.3381	100.0	0.0
	2	0.2730	100.0	51.3
	4	0.1801	100.0	81.7
	8	0.4807	100.0	9.2
	SFT	0.1187	100.0	100.0
GPT-4o-mini	0	0.3381	100.0	0.0
	2	0.2810	100.0	62.4
	4	0.2236	100.0	75.6
	8	0.4536	100.0	16.6
Gemini-2.5-Flash	0	0.4044	100.0	0.0
	2	0.2620	100.0	63.6
	4	0.1722	100.0	71.8
	8	0.2503	100.0	20.2