PDEBench-Lang: Does Notation Formation Shape Neural Reasoning About PDEs?

LLMs learn notation, not PDE structure — representation controls generalization.

CSCI 5541 NLP Final Project | Token Efforts

Raghav Anand Peilin Li Yit Xiang Ztang Krivan Semlani

University of Minnesota

THE BIG PICTURE

Same PDE written in different ways. Does the format change how well a model reasons about it?

1 MOTIVATION

LLMs solve PDEs 4–6× faster by predicting operators, but they hardcode Postfix without validation.

LLMs are pretrained on LaTeX and natural language. A representational mismatch may silently degrade reasoning.

OUR QUESTION

Does the notation format of a PDE affect how well a model reasons about it?

SAME PDE, FOUR REPRESENTATIONS

Wave PDE: $u_{tt} = c^2 u_{xx}$

Natural Language	The second time derivative of u equals c squared times the second spatial derivative of u.
LaTeX (code)	`u_{tt} = c^2 u_{xx}`
Prefix (code)	`=(d(d(u,t),t),*(^(c,2),d(d(u,x),x)))`
Postfix (code)	`u t d t d c 2 ^ u x d x d * =`

We train one model per dialect and compare formats.

2 OUR APPROACH

Dataset

- 6 PDE families:
  - Burgers
  - Heat
  - Wave
  - Laplace
  - Klein-Gordon
  - Advection
- 12,000 instances (2,000 per family)
- 4 dialects

Natural | LaTeX | Prefix | Postfix

Augmentations

k-scaling $(u_t = \alpha\, u_{xx} \rightarrow u_t = k \Phi\, u_{xx})$

Directional shuffling $(u_t = u_{xx}\ u_x = w \rightarrow u_x = v\ u_y = w \rightarrow u_y = w)$

Positional shuffling $(u_t = u_{xx} \rightarrow u_t \circ \Phi\, u_{xx} = 0 \text{ or } u_{xx} = u_t)$

Models

Two model families

BART-base (139M)

T5-small (60M)

One model per dialect (4 dialects × 2 models = 8 models)

Evaluation

- In-dialect (test on own format)
- Cross-dialect (test on all 4 formats)

Metrics

- Family Accuracy
- Operator F1
- Trash Score (% invalid / meaningless)

Split: 80% Train / 10% Val / 10% Test (same split across all models)

3 BASELINE (Random Guess)

Dialect	Family Accuracy
LaTeX	16.67%
Postfix	16.67%
Prefix	16.67%
Natural L	16.67%

Random baseline: uniform guess among 6 families (1/6 = 16.67%). All models far outperform random baseline in-dialect, but cross-dialect performance remains challenging.

4 OUR CONTRIBUTIONS

- **First systematic study** of how notation format affects neural reasoning for PDEs.
- A large synthetic PDE dataset (12K instances across 6 families) with 3 augmentation strategies and 4 dialects.
- **Comprehensive comparison across BART and T5** using Family Accuracy, Operator F1, and Trash Score.
- **Key empirical insight: Postfix > Prefix >> Natural, format choice fundamentally shapes generalization.**

5 RESULTS: BART vs T5 (Same Setup, Different Models)

A. In-Dialect Performance (diagonal)

Tested on its own format

Model	Natural	LaTeX	Prefix	Postfix
BART	100% (Family Acc)	100% (Family Acc)	100% (Family Acc)	100% (Family Acc)
T5	100% (Family Acc)	100% (Family Acc)	100% (Family Acc)	100% (Family Acc)

- Both BART and T5 achieve 100% Family Accuracy in-dialect.
- Operator F1 (BART): 70–99% | Operator F1 (T5): 71–99%
- Trash Score (BART): 0–10% | Trash Score (T5): 3–9%

B. Cross-Dialect Generalization (Family Accuracy, 4×4 Matrix)

Train on (rows) → Test on (columns)

BART

	Natural	LaTeX	Prefix	Postfix
Natural	100%	20%	20%	20%
LaTeX	0%	100%	0%	0%
Prefix	0%	0%	100%	50.7%
Postfix	3.9%	44.8%	49.4%	100%

T5

	Natural	LaTeX	Prefix	Postfix
Natural	100%	17%	17%	20%
LaTeX	0%	100%	0%	0%
Prefix	24%	100%	33%	17%
Postfix	34%	33%	100%	17%

High / Low

Average off-diagonal Family Accuracy (higher is better)

Natural	LaTeX	Prefix	Postfix	Best:
20.0%	0.0%	37.7%	49.5%	Postfix

Average off-diagonal Family Accuracy (higher is better)

Natural	LaTeX	Prefix	Postfix	Best:
17.3%	1.0%	24.7%	28.0%	Postfix

C. Metric Comparison: BART vs T5 (Averaged over all cross-dialect pairs)

Metric	BART (higher is better ↓ for Trash)	T5 (higher is better ↓ for Trash)	Winner
Family Accuracy (↑)	26.8%	17.5%	BART
Operator F1 (↑)	80.0%	76.8%	BART
Trash Score (↓)	33.3%	36.3%	BART

BART generalizes better across notation formats on all three metrics.

6 KEY TAKEAWAYS

Notation format matters.
Same PDE, different format → large performance gaps.

Postfix generalizes best.
Most transferable across notation styles (BART & T5).

Prefix is the strongest among symbolic formats (excl. Postfix).
Good structural generalization, especially with T5.

Natural language struggles to transfer.
Overfits to surface wording: weak symbolic generalization.

Trash Score reveals breakdowns.
High trash = invalid or meaningless outputs.

7 LIMITATIONS & FUTURE WORK

Limitations

- Synthetic data may not reflect real-world PDEs.
- Models may learn syntax patterns, not semantics.
- Cross-dialect collapse may persist as models scale.
- Trash Score depends on heuristic validity rules.

Future Work

- Evaluate on real PDE datasets.
- Scale to larger models (T5-large, BART-large).
- Explore more dialects and hybrid formats.
- Improve semantic evaluation beyond syntax.

8 REFERENCES

1. Bhatnagar, A. et al. (2025). From equations to insights.
2. d'Ascoli, S. et al. (2024). ODEFormer: Symbolic regression of dynamical systems. ICLR 2024.
3. Lample, G. & Charton, F. (2020). Deep learning for symbolic mathematics. ICLR 2020.
4. Sun, H. et al. (2024). PROSE-PDE: Multimodal learning of governing equations.

POSTFIX vs PREFIX: WHY THE DIFFERENCE?

Postfix: Sequence-Based (Surface Order)

Operators follow operands immediately. Linear, left-to-right patterns are easy for models to re-match across dialects.

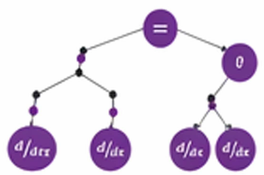
u	x	d	x	d	c	2	^	=

- No brackets
- No precedence tracking
- Strong cross-dialect transfer ✓

VS.

Prefix: Structure-Based (Operator Tree)

Explicit nesting encodes operator depth. Learns structural rules that generalize to unseen, more complex forms.



- Encodes hierarchy
- Captures depth & structure
- Strong zero-shot generalization

Takeaway

Postfix is easier to transfer across formats.

Prefix is better at capturing structure.

Both explore complementary strengths.