Language Models Can Benefit from Domain-Specific Inductive Biases

Damien Teney[1,3] Liangze Jiang[1,2] Zachary Shinnick[3] Hemanth Saratchandran[3] Simon Lucey[3]
[1]Idiap Research Institute [2]EPFL [3]Australian Institute for Machine Learning, Adelaide University

TL;DR Small LMs are surprisingly easy to improve with architectures tuned for each target domain.

1 PROBLEM

LLMs are pretrained on diverse data mixtures, and modern transformers are refined for such general-purpose learning.

Are transformers equally effective across subsets of the pretraining data?

2 METHOD: CUSTOM NONLINEARITIES

Swap GeLUs/softmax for learned splines, **tuned per domain**. Learnable & strongly determine the model's inductive biases. [1]

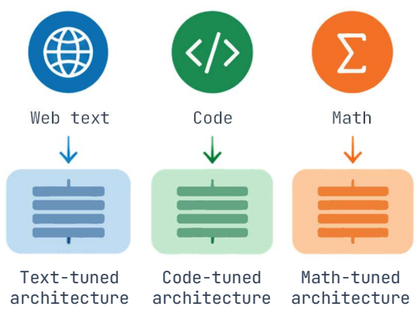
[1] Neural Redshift: Random Networks are not Random Functions, Teney et al. CVPR 2025

```
1 Warm-up on domain A (train weights/splines jointly)
2 Freeze the splines
3 Train from scratch on domain B
```

3 KEY FINDINGS

Domain-specific tuning

Specialized architectures always **train faster and generalize better**.



Natural language	FineWeb, Enwik8 (without formatting tags), Shakespeare, TinyStories
Code	CodeSearchNet-Python and -Java (without language comments)
High-school math	DeepMind QA (50+ question types)
Algorithmic tasks	string manipulation, arithmetic; improves length generalization! [2]

[2] Can Transformers Really Do It All? On the Compatibility of Inductive Biases Across Tasks, Teney et al. ICLR 2026

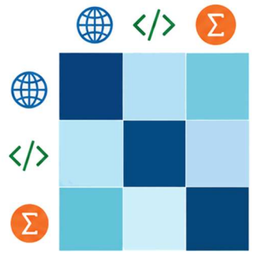
Mix-and-match

Transfer is often domain-specific!

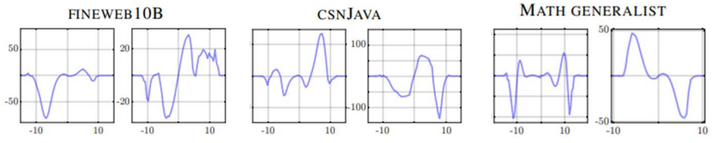
– Architectures tuned on **FineWeb** work well for **language/code** datasets.

– Architectures tuned for **math** or **algorithmic reasoning** completely fail on everything else.

– Architectures tuned for **specific math question types** may not even transfer to other math question types.



Our nonlinearities look nothing like standard activation functions:



Caveat: small models (40–120M), but large study (>6,000 models) covering seeds and hyperparamaters.

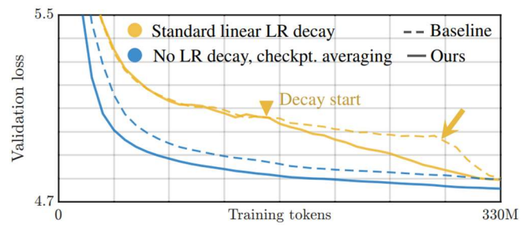
4 WHY DOMAIN-SPECIFIC BIASES HELP

More "Winning Lottery Tickets"

Even in mostly-untrained models (freezing all layers except embeddings), models with tuned activation functions are significantly better (22% → 40% accuracy on Math QA).
They contain **useful circuits for the target task** at initialization.

Wider Loss Basins

Baseline models rely on LR decay to reach sharp local minima.
Ours converge steadily **even without LR decay**.



5 WHAT TO DO WITH THESE FINDINGS?

```
Standard transformers are generalist compromises.
Domain-specific tuning can improve data efficiency.
```

How to train general-purpose models & leverage **diverse inductive biases**?
MoEs with **heterogeneous architectures**?
Other **effective interventions**? Activation functions were just a tool!