Shortening the Loss Plateau

Jet Yue
jyue@ucsd.edu

Tommy Li
wal016@ucsd.edu

Samuel Cho
sjc006@ucsd.edu

Mentor: Tianhao Wang
tianhaowang@ucsd.edu

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

INTRODUCTION

- Transformer models often get “stuck” during training: exhibiting flat learning curves that waste compute and slow development
- These plateaus are not one problem but **two distinct phenomena** occurring at different stages of training

Training Loss Plateau

- Occurs early in training: loss stays near a suboptimal value before suddenly dropping
- Driven by representation collapse, repetition bias in token embeddings, and delayed attention map formation

Generalization Plateau (Grokking)

- Occurs later: model reaches ~ 100% training accuracy quickly, but generalization is massively delayed
- After memorizing training data, models slowly reorganize internal representations under implicit regularization pressure (e.g., weight decay), eventually converging to compact, generalizable structures — causing a sudden spike in validation accuracy

METHODS

Training Loss Plateau

- 1-layer, 1-head Transformer with linear causal attention and 2-layer MLP (GELU activation)
- Trained online with batch size 256; objective is next-token cross-entropy loss
- Tasks: Moving Window Sum, Product, and Difference + Prefix Sum (sequence length 16, modulus p=17)
- Multi-task batches mix sequences from different tasks using task-specific separator tokens

Generalization Plateau (Grokking)

- 2-layer decoder-only Transformer; d_model=128, 4 attention heads, ReLU activation
- Trained with AdamW (lr=0.001, weight decay=0.001) for 400,000 steps; batch size 512
- Weight decay is critical — without it, models memorize but never generalize
- Tasks: Modular Division, Addition, Subtraction, Multiplication (mod 97)
- 50/50 train-validation split; grokking defined as ≥95% validation accuracy

RESULTS

Task Diversity on the Training Loss Plateau

- Mixing modular moving window tasks significantly shortens loss plateau for each task
  - Convergence requires up to 63% fewer training samples as task diversity increases
- The model learns a structured attention map, concentrating on the task-specific separator token

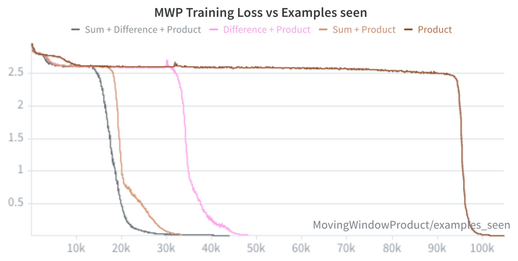


Figure 1: Model is able to learn MWS task with less training samples when mixed

	# Training samples needed for training to converge			Avg Speed Up
# Task	MWS	MWP	MWD	
1	16256	93568	34560	N/A
2	13600	24096	19680	44.55%
3	10542	13288	10794	63.23%

Table 1. Task Diversity allows training to converge on less samples

Transfer Learning

- Pretraining on a task → Fine Tuning on another task
- Eliminates the loss plateau entirely — model converges without any observable stall

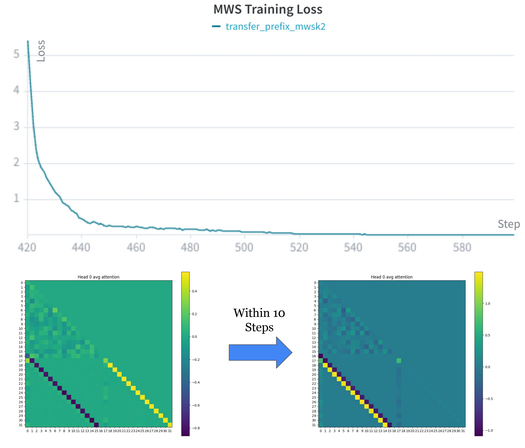
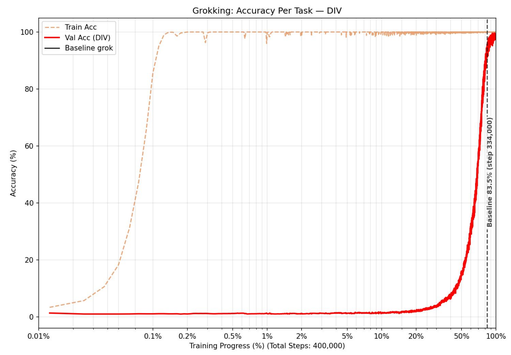


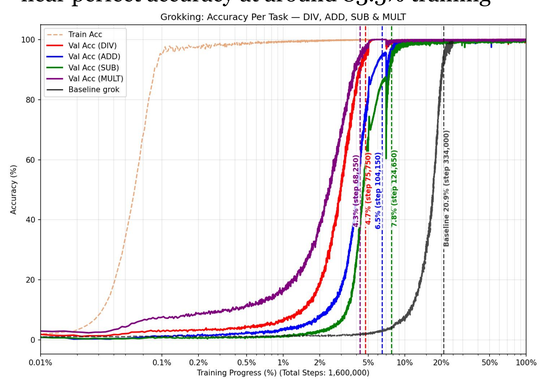
Figure 2. Prefix sum → MWS eliminates loss plateau

RESULTS

Task Diversity on the Generalization Plateau



- Baseline run: Training accuracy hits ~100% by around 0.2% training progress, but validation accuracy stays near 0% before eventually reaching near perfect accuracy at around 83.5% training



- Multi-task run: Training on all four arithmetic tasks simultaneously accelerates grokking across every task achieving an approximate 18x speedup over the baseline.

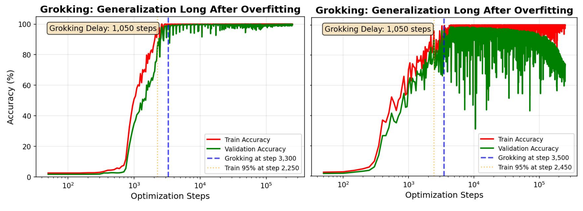


Figure 3: Sparse and Small initialization eliminate the grokking delay, achieving generalization in ~1,050 steps

- Sparse Initialization (90% zeroed) and Small Initialization (scale=0.01) both prevent early memorization entirely
- Generalization achieved in ~1,050 steps — an approximate 100x speedup over the baseline
- Optimizer Choice: SGD (lr=0.005) achieves grokking in ~41,500 steps — a ~60% reduction over AdamW, while maintaining stable post-grokking accuracy

DISCUSSION

Training Loss Plateau

- When pretrained, fine-tuning mostly shifts bias terms in attention matrices
  - Learning the correct hidden representation may be the bottleneck
  - Incorporating task diversity can provide diverse training signals, enabling faster adaptation and more efficient formation of hidden representations.

Generalization Plateau (Grokking)

- We hypothesize that shared modular structure across tasks encourages the model to learn the underlying math rather than memorize task-specific answers, accelerating generalization.

- Both plateaus share a common driver: the model settling into a low-effort, degenerate solution early in training
- Task diversity resolves this by forcing the model to satisfy multiple objectives simultaneously, preventing collapse
- Constrained initialization resolves this by denying the model capacity to memorize, leaving generalization as the only available path
- SGD's inherent gradient noise helps the model escape the memorization minimum faster than AdamW's smoother updates

CONCLUSION

Both optimization plateaus can be significantly shortened by promoting diverse representations via multi-task training, restricting initial network capacity through sparse or small initialization, and introducing gradient noise via SGD — interventions that consistently prevent degenerate early solutions

ACKNOWLEDGEMENT

Special thanks to our mentor Dr. Tianhao Wang for guiding us on this project!