IMAGINE: SELECTIVE WIDTH EXPANSION FOR HARD-TOKEN REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Small language models can fail at a few difficult predictions and spoil an otherwise correct answer. Extra reasoning steps can repair these errors, but serial computation makes users wait. We introduce IMAGINE, a post-training method for selective width expansion within a single Transformer forward. Parallel Slot Expansion gives difficult positions persistent expert states. These states interact across layers and merge only at prediction time. A Lookahead Gain Controller allocates width before the current input enters the model. Dense-to-Hard Training teaches the model to use this capacity, then concentrates it on difficult predictions. Our analysis connects the resulting structured wide network to conditions for lower prediction loss. Across three reasoning benchmarks, Imagine+ 0.6B achieves 5.8% higher average accuracy than plain supervised fine-tuning. Parameter-matched variants also improve average accuracy. Moreover, the single-slot model preserves SFT-level single-user throughput and reduces the mean latency of the slowest 1% of tokens by 34.9% relative to parameter-matched recurrent transformer.

1 INTRODUCTION

Small language models need more capacity at some predictions than at others. Most words in an answer may be routine, but one mistaken sign, variable, or logical connective can invalidate the result. A larger model supplies more capacity everywhere. Extra reasoning steps spend more time on the answer. Neither choice directly matches the needs of a user waiting for a single response: stronger predictions at the few difficult positions, with little additional delay. We study a different allocation of computation: *can a small model become temporarily wider at difficult positions, instead of taking more sequential steps?*

Textual and latent reasoning methods, including STaR and Quiet-STaR, learn intermediate steps (Wei et al., 2022; Zelikman et al., 2022; 2024; Hao et al., 2025; Shen et al., 2025). Selective recurrent models, including Think-at-Hard (TaH) and Mixture-of-Recursions, refine difficult predictions through repeated computation (Fu et al., 2026; Bae et al., 2025b; Geiping et al., 2025; Zhu et al., 2025b). Serial refinement extends the path to the next token. Pause tokens, Hidden Decoding, and Hyper-Connections add intermediate states (Goyal et al., 2024; Liu et al., 2026; Zhu et al., 2025a). These states share transformations or feed a common branch, without assigning each state its own expert parameters. Self-MoE and CARE specialize transformations but merge expert outputs within modules (Zadouri et al., 2024; Kang et al., 2025; Saliencro et al., 2026). Our goal is to retain specialized states across layers and allocate them before execution, without adding serial refinement.

To this end, we propose IMAGINE.[1] Its Parallel Slot Expansion (PSE) gives selected positions additional full-width expert states, called slots. The states exchange information across layers and remain separate until the final prediction. This persistence lets several specialized computations develop together instead of repeatedly compressing their results into one state. Figure 1 illustrates the difference from serial refinement. Ordinary positions use the backbone alone. Selected positions gain additional computational width in the same forward pass.

[1] Code is available at this link.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

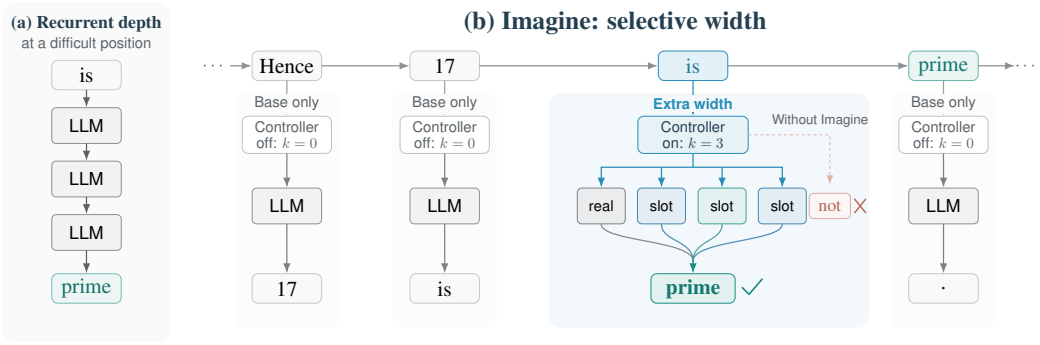


Figure 1: Selective computation during generation. **(a)** Recurrent refinement applies the same LLM serially at a difficult position. **(b)** Imagine uses one forward per position. The controller selects width before execution: off for "Hence" and "17", on for "is", and off again for "prime", with k denoting the number of extra slots. Inputs appear above each step and next-token outputs below. The dashed red "not" is the backbone-only counterfactual, while the expanded path emits "prime". The example is schematic.

Two obstacles make selective width difficult. First, adding states does not give a pretrained decoder a reason to use them. The added paths may stay redundant or contribute little to the final prediction. PSE supplies private transformations, and Dense-to-Hard Training (DHT) teaches the backbone and readout to exploit them before sparse activation. Second, useful expansion must be predicted before its result is available. Current-position hidden states require model layers to run first. Our Lookahead Gain Controller (LGC) instead uses the preceding state and current input embedding to forecast the gain and select width before any current-position layer.

We evaluate 0.6B models on three mathematical reasoning benchmarks. Imagine improves average accuracy by 5.8% relative to plain SFT and outperforms it on all three benchmarks, obtains comparable performance with selective recurrent transformers. Furthermore, we measure single-user decoding with parameter-matched models. The single-slot variant reduces the mean latency of the slowest 1% of tokens by 34.9% and increases throughput by 8.9% relative to TaH, while preserving SFT-level throughput.

Our technical contributions are:

- **Persistent selective width.** PSE retains communicating expert states through one forward, with a structured-width characterization.
- **Trainable pre-forward allocation.** DHT learns useful slot computations, and LGC selects their budget before backbone execution.
- **Quality with responsiveness.** Three-task gains and single-stream measurements demonstrate the benefit of selective width beyond model size.

We next position Imagine among related methods. Sections 3 and 4 develop the architecture and training procedure. Section 5 connects them to prediction improvement, with proofs in Appendix A.2. Sections 6–8 evaluate answer quality, single-user efficiency, and the role of each component.

2 Related Work

We group prior work by how it adds or allocates computation. Appendix E compares individual routes and explains their roles in our experiments.

Latent reasoning and recurrent Transformers. *Chain-of-thought reasoning* introduces textual rationales or continuous latent steps (Wei et al., 2022; Zelikman et al., 2022; 2024; Hao et al., 2025; Shen et al., 2025; Zhang et al., 2025; Fu & Luo, 2026). *Recurrent Transformers* reuse shared layers, optionally with depth-specific adapters (Dehghani et al., 2019; Geiping et al., 2025; Zhu et al.,

2025b; Bae et al., 2025a). *Adaptive computation* assigns recursion or exits by position (Fu et al., 2026; Bae et al., 2025b; Zeng et al., 2026; Schuster et al., 2022). These routes adapt the sequence of reasoning steps or layer applications. Our question is how to strengthen difficult predictions through additional states in the same forward. Imagine selects width before the current input enters the backbone, keeping the extra paths parallel.

Parallel latent reasoning. *Pause tokens and hidden decoding* use pauses, fillers, or repeated token streams (Goyal et al., 2024; Pfau et al., 2024; Wu et al., 2025b; Liu et al., 2026). *Parallel reasoning and parallel scaling* aggregate reasoning or model paths, including looped and recommendation-specific formulations (Wu et al., 2025c; Fan et al., 2026; Tang et al., 2026; Chen et al., 2025; Wu et al., 2025a). *Hyper-Connections* expands the representations carried between layers (Zhu et al., 2025a; Xie et al., 2026; Zhang et al., 2026). Together, they establish that parallel states can provide useful computation. Imagine addresses their selective specialization: each admitted slot carries its own expert identity through the backbone, exchanges information with preceding lanes, and remains distinct until readout. Joint post-training teaches the decoder to use this capacity, and LGC selects where to activate it.

Mixture-of-experts and conditional computation. *Mixture-of-experts models* combine low-rank adapters, self-specialized models, or subtoken experts (Hu et al., 2022; Zadouri et al., 2024; Wu et al., 2024b; Luo et al., 2024; Li et al., 2024; Kang et al., 2025; Wu et al., 2024a). *Adaptive expert routing* varies expert budgets (Li et al., 2023; Huang et al., 2024; Nishu et al., 2025; Saliencro et al., 2026). *Pre-gated routing and dynamic sparsity* anticipate expert execution or selectively use existing heads, layers, and channels (Hwang et al., 2024; Cai et al., 2024; Liu et al., 2023; Raposo et al., 2024; Hu et al., 2026). These mechanisms choose transformations of a carried token state. PSE additionally preserves expert-specific states across layers, allowing later computation to use their distinct intermediate results. Our ablations compare CARE's module-level mixture principle through a controlled selective-expert counterpart (Section 8).

3 Parallel Slot Architecture

PSE gives a difficult prediction distinct expert computations in one Transformer traversal. The states share an input, specialize through private parameters, and exchange information across layers. They merge only at readout.

3.1 Generation Setup

Let x_i be the observed token, $x_{\le i}$ its prefix, and $y_i = x_{i+1}$ the prediction target in vocabulary $\mathcal{V}$. The backbone (Vaswani et al., 2017) has L decoder layers of width d and embedding map E. Lane $j = 0$ carries the real token; lanes $1, \dots, k_i$ are slots. LGC selects $k_i \in \mathcal{K} = \{0, 1, 3\}$ before the current forward. All active lanes traverse the same layers once. Their layer-ℓ states are $h_{i,j}^{(\ell)} \in \mathbb{R}^d$, with $\ell = 0$ denoting the input. Appendix A collects the notation.

3.2 Slot-Specific Experts

Slots need distinct computations to offer more than copies of the backbone. They start from the same observed embedding:

$$h_{i,0}^{(0)} = E(x_i), \qquad h_{i,j}^{(0)} = E(x_i), \quad 1 \le j \le k_i. \tag{1}$$

Each slot specializes the shared attention and feed-forward projections through private LoRA increments (Hu et al., 2022):

$$W_{\ell,P}^{(j)} = W_{\ell,P} + \Delta_{\ell,P,j}, \qquad \Delta_{\ell,P,j} = \frac{\alpha_{\mathrm{L}}}{r} B_{\ell,P,j} A_{\ell,P,j}. \tag{2}$$

Here P indexes a projection, r is the adapter rank, and α_{L} is its scale. Matrices A and B are learned low-rank factors. The real lane uses $\Delta_{\ell,P,0} = 0$. Appendix B.1 specifies matrix shapes and operators. Depth-specific adapters provide a related way to specialize shared weights in recurrent models (Bae et al., 2025a).

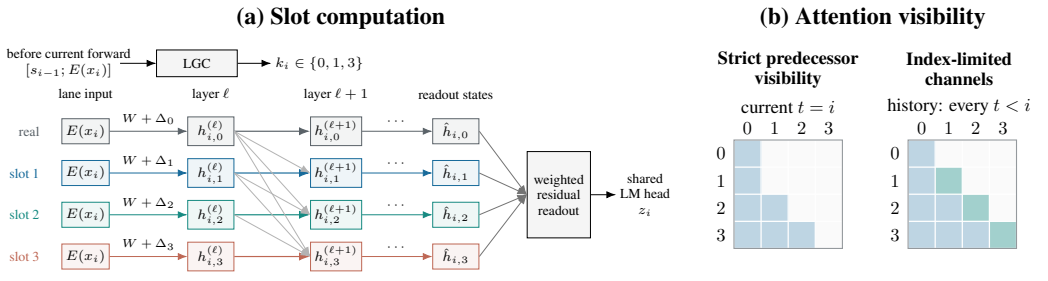


Figure 2: **(a)** Slot computation. The real lane uses the backbone weights ($\Delta_0 = 0$), and slot j uses its private increment Δ_j. The diagram suppresses layer and projection indices in W and Δ_j. The controller receives the preceding-position summary s_{i-1} and current embedding $E(x_i)$. The final vectors $\hat{h}_{i,j}$ include the backbone's output normalization. Residual paths and ordered messages preserve the states until the final readout. **(b)** Attention visibility. Rows index queries j and columns index keys m, with 0 denoting the real lane. Colored cells are visible and white cells are masked. Slot queries read $m < j$ in the current group and $m \leq j$ at every earlier position. Teal cells highlight historical same-index access. Real queries read only real keys. Future and absent keys are masked.

The group state $H_i^{(\ell)} \in \mathbb{R}^{(k_i+1)\times d}$ stacks the lane vectors as rows. Each lane retains its own attention query, nonlinear feed-forward path, and residual state. Communication preserves these separate paths until the final prediction. At $k_i = 0$, the current trained backbone supplies the output.

3.3 Slot Communication

Ordered communication lets later slots use earlier lanes without collapsing their states. Each attention or mixing operation reads a common pre-update tensor, so visibility constraints do not require serial backbone passes.

The real lane reads the causal real-token prefix, including itself. Slot j reads that prefix and current-position lanes $m < j$ (Figure 2). Its own residual path carries its state across layers. Appendix B.1 gives the explicit binary mask.

Each layer also applies an ordered residual mixer:

$$H_i^{(\ell)} = \bar{H}_i^{(\ell)} + \gamma_\ell \, \mathrm{GELU}\Big(M_{\ell,k_i} \, \mathrm{LN}(\bar{H}_i^{(\ell)})\Big). \tag{3}$$

Here $\bar{H}_i^{(\ell)}$ is the decoder-layer output before mixing, γ_ℓ is a learned scale, and M_{ℓ,k_i} contains learned nonnegative weights only at $m < j$. Slot rows sum to one; the real row is zero. LN normalizes each lane and GELU acts elementwise. This mixer preserves the complete group state (Lemma 8, proof in Appendix A.11). Unlike unordered set architectures (Zaheer et al., 2017; Lee et al., 2019), PSE uses fixed slot identities and ordered messages.

3.4 Latent History

Earlier expanded predictions can inform later ones. At position $t < i$, slot j may read existing historical slots $m \leq j$, as well as the causal real-token prefix (Figure 2). Real queries read only real keys. The current real token and its slots share a positional index. These rules preserve the real trajectory at fixed parameters and prefix (Lemma 2, proof in Appendix A.3).

3.5 Slot Readout

A learned residual readout converts the separate states into one prediction. Let $\hat{h}_{i,j}$ be the final normalized state, $q \in \mathbb{R}^d$ a learned pooling query, β_j a learned slot bias, α a learned scale, and

$U \in \mathbb{R}^{|\mathcal{V}| \times d}$ the shared bias-free vocabulary projection. For $k_i > 0$:

$$g_{i,j} = q^\top \hat{h}_{i,j}/\sqrt{d} + \beta_j, \qquad \omega_{i,j} = \frac{\exp(g_{i,j})}{\sum_{m=1}^{k_i} \exp(g_{i,m})}, \tag{4}$$

$$h_i^\star = \hat{h}_{i,0} + \alpha \sum_{j=1}^{k_i} \omega_{i,j}(\hat{h}_{i,j} - \hat{h}_{i,0}), \quad z_i = U h_i^\star. \tag{5}$$

At $k_i = 0$, the model uses $h_i^\star = \hat{h}_{i,0}$. Write $r_i = \sum_j \omega_{i,j}(\hat{h}_{i,j} - \hat{h}_{i,0})$ for the pooled residual, with $r_i = 0$ at zero budget. The logit correction is $\alpha U r_i$ relative to $z_i^{(0)} = U\hat{h}_{i,0}$ (Lemma 3, proof in Appendix A.4). Shared backbone weights remain jointly trainable.

4 Training Imagine

The model must learn useful expanded computations before a controller can select them. DHT jointly trains the backbone and slot modules. A second, frozen-model stage trains LGC from comparisons among the learned budgets.

4.1 Dense-to-Hard Curriculum

DHT starts with broad slot exposure and gradually restricts it to fixed reference mismatches. This addresses the learning difficulty highlighted by pause and filler-token studies (Goyal et al., 2024; Pfau et al., 2024). Admitted positions sample one or three slots; other positions use the backbone alone. Appendix B.3 gives the exact schedule, sampling rule, and length-cap handling.

Let θ collect all jointly trained parameters, m_i indicate a valid assistant target, and z_i be the logits actually served at position i. Training minimizes one loss per next real token:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{\sum_i m_i} \sum_i m_i \log \mathrm{softmax}(z_i)_{y_i}. \tag{6}$$

Gradients update the shared backbone and active slot modules. The same objective applies to ordinary and expanded predictions. Appendix B.4 specifies masking and distributed normalization.

4.2 Training the Controller

LGC learns which budget benefits a prediction before observing its output. We freeze the trained model and compare budgets $k \in \mathcal{K}$ on teacher-forced candidates:

$$c_i = \min\big(\arg\min_{k \in \mathcal{K}} \mathcal{C}_i(k)\big), \qquad \mathcal{C}_i(k) = -\log p^{(k)}(y_i \mid x_{\le i}). \tag{7}$$

Here $p^{(k)}$ is the trained model's distribution at budget k, and $k = 0$ disables its slots. Ties favor the smaller budget. Each budget follows its own teacher-forced latent history. Lemma 4 formalizes the finite-table loss ordering (Appendix A.5). CE labels capture changes in target probability even when the top token stays unchanged. They retain information that binary correction labels discard.

LGC combines the preceding composite state s_{i-1} with the current embedding $E(x_i)$ in a residual MLP trained by weighted classification CE. It outputs $\pi_{i,k} = \pi_\psi(k \mid s_{i-1}, E(x_i))$, where ψ denotes its parameters. For threshold $\tau \in [0, 1]$, expansion occurs when $\pi_{i,1} + \pi_{i,3} \ge \tau$. The larger positive-class probability chooses the budget, with ties favoring one slot. All inputs are available before the current forward. Appendix B.7 gives the feature variants and training configuration. We evaluate preset thresholds on MATH-500 and report the resulting activation rates (Section 6).

5 Theoretical Analysis

We connect persistent states to prediction improvement through structured width and an error-aligned readout. Auxiliary lemmas and all proofs appear in Appendix A.2; the results use exact arithmetic and fixed real prefixes. We use CE for next-token cross-entropy. Pause-token expressivity provides related motivation (London & Kanade, 2025).

5.1 Structured Width

The slot group implements a structured wide computation while sharing the backbone parameters.

Proposition 1 (Structured width). *At a position with fixed budget $k \in \mathcal{K}$, each projection in Equation 2 is exactly the action of*

$$\widetilde{W}_{\ell,P} = \mathrm{diag}(W_{\ell,P}, W_{\ell,P} + \Delta_{\ell,P,1}, \ldots, W_{\ell,P} + \Delta_{\ell,P,k}) \tag{8}$$

on stacked lane inputs. Here diag *forms a block-diagonal matrix. At decoder-layer boundaries, the stacked hidden dimension is $(k+1)d$. Coordinatewise nonlinearities and per-lane residuals preserve this structured multi-state representation. The masked attention and mixer add constrained cross-lane communication.*

The proof is in Appendix A.6. The structured wide computation shares backbone parameters while retaining distinct nonlinear paths that DHT can specialize.

Distinct queries can retain more independently varying context components (Lemma 5, Appendix A.7). A reducing merge has local rank at most d, while the ordered residual mixer preserves its full workspace (Lemmas 9 and 8, Appendices A.12 and A.11).

5.2 Prediction Improvement

The readout changes logits by $\alpha U r_i$. The CE identity in Lemma 6 gives a sufficient condition for this change to improve prediction (Appendix A.8).

Proposition 2 (Selective improvement). *Fix a finite prediction set $\mathcal{I}$, selected positions $S \subseteq \mathcal{I}$, real prefixes, states, pooling weights, and backbone. Let $w_i \geq 0$ with $\sum_{i\in\mathcal{I}} w_i > 0$, $d_i = U r_i$, and $p_i^{(0)} = \mathrm{softmax}(z_i^{(0)})$, with $r_i = 0$ outside S. Define*

$$G = \sum_{i\in S} w_i (e_{y_i} - p_i^{(0)})^\top d_i, \qquad D = \sum_{i\in S} w_i \|d_i\|_2^2. \tag{9}$$

Here e_{y_i} is the target's one-hot vector and $\|\cdot\|_2$ is the Euclidean norm. If $G > 0$, any $0 < \alpha < 4G/D$ strictly lowers the weighted CE relative to the same backbone with expansion disabled.

Appendix A.9 gives the proof. The served CE directly trains the directions appearing in this condition: its readout-scale gradient is their prediction-error alignment (Lemma 1). DHT exposes these paths broadly before concentrating their use on difficult predictions.

5.3 Selective Optimization

The served objective trains the alignment required above. Budget-selection errors limit the gains available to LGC (Lemma 7, proof in Appendix A.10).

Lemma 1 (Prediction-aligned learning). *Under the fixed-state setup of Proposition 2, define the weighted mean served CE as a function of readout scale:*

$$\mathcal{E}(\alpha) = \frac{\sum_{i\in\mathcal{I}} w_i \,\mathrm{CE}(z_i^{(0)} + \alpha d_i, y_i)}{\sum_{i\in\mathcal{I}} w_i}. \tag{10}$$

Then $\mathcal{E}'(0) = -G/\sum_{i\in\mathcal{I}} w_i$. A parameter-independent finite curriculum gives an unbiased sampled gradient of its expected served-loss objective.

The proof is in Appendix A.13. Thus the trained readout receives the same alignment signal that appears in Proposition 2.

6 Experimental Setup

We test answer quality and single-user efficiency under a common model family and length setting. Parameter-matched variants and ablations distinguish selective width from simply adding parameters.

Table 1: Main results at maximum generation length 4096. Each entry is accuracy with activation rate in smaller parentheses, both in percent. Deterministic branches omit activation rates. Imagine model names are bold. The highest accuracy in each row is bold and the second highest is underlined. Average is the arithmetic mean across tasks.

	SFT	TaH+ 0.6B		**Imagine-1 0.6B**		**Imagine-3 0.6B**		**Imagine+ 0.6B**	
Task		NoThink	Think	NoImagine	Imagine	NoImagine	Imagine	NoImagine	Imagine
MATH-500	40.20	38.20	39.20(0.93)	38.20	<u>41.40</u>(8.41)	34.40	40.20(9.98)	37.00	**42.40**(6.37)
AMC23	20.94	18.44	20.62(0.82)	**22.50**	21.25(8.42)	20.31	<u>21.88</u>(10.30)	21.56	21.56(6.18)
OlympiadBench	13.33	13.78	13.93(0.86)	13.48	<u>14.67</u>(7.99)	12.89	13.78(10.95)	12.15	**14.81**(6.64)
Average	24.82	23.47	24.58(0.87)	24.73	<u>25.77</u>(8.27)	22.53	25.29(10.41)	23.57	**26.26**(6.40)

Models and training. All models start from Qwen3-0.6B-Base (Yang et al., 2025). We follow TaH's OpenR1-Math preparation (Fu et al., 2026; Lozhkov et al., 2025) and retain training and held-out sequences of at most 4096 real tokens. Imagine jointly trains the backbone and slot modules. SFT uses full-model fine-tuning; we select epoch 4 with held-out perplexity taken into account after a five-epoch run. TaH+ 0.6B follows the official recipe with training and generation lengths set to 4096. Imagine+ 0.6B retains the full backbone. The parameter-matched Imagine-1 and TaH variants remove one layer, and Imagine-3 removes two. Our CARE-style control uses ordinary expert mixing under the common controller workflow (Saliencro et al., 2026). Appendices B and C.2 give training settings and parameter counts.

Controller and evaluation. LGC training freezes the backbone. We scan preset numerical thresholds on MATH-500, then keep each reported threshold fixed across tasks (Appendix C.4). We evaluate MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024; HuggingFaceH4, 2024), AMC23 (math-ai, n.d.), and English mathematics problems from OlympiadBench (He et al., 2024), with at most 4096 new real tokens. MATH-500 and OlympiadBench use one answer per problem. AMC23 averages eight sampled answers per problem rather than computing pass@8. Task averages weight benchmarks equally. Activation rate counts scored real-token predictions with any positive slot or iteration budget, not the number of slots. Appendix B.8 records sampling settings, operating points, and batch/stopping differences. Single-stream efficiency uses batch size one on the same questions for all models (Appendix C.5).

7 Results

We first evaluate answer accuracy across three reasoning benchmarks. We then measure single-stream throughput, token latency, and memory to test the practical benefit of placing extra computation in parallel.

7.1 Downstream Performance

We test whether the learned extra capacity improves answers over both the model's disabled branch and plain SFT. Table 1 includes full and parameter-matched variants to assess this benefit under both parameter settings. Imagine+ 0.6B achieves the highest three-task average, 26.26%, compared with 24.82% for SFT and 24.58% for TaH+ 0.6B. This is an approximately 1.4-point average gain over SFT, or 5.8% in relative terms. Imagine+ 0.6B exceeds SFT on each benchmark, by 2.20 points on MATH-500, 0.62 on AMC23 and 1.48 on OlympiadBench. On MATH-500, it improves from 37.0% to 42.4% while activating slots at 6.37% of scored predictions. Its OlympiadBench accuracy rises from 12.15% to 14.81%, while AMC23 accuracy is preserved.

The parameter-matched variants also benefit from selective slots. Imagine-1 0.6B raises MATH-500 accuracy by 3.2 points and improves its three-task average from 24.73% to 25.77%. Imagine-3 0.6B improves on all three tasks relative to NoImagine, including a 5.8-point MATH-500 gain. Their average accuracies exceed SFT by 0.95 and 0.46 points, respectively. Imagine+ 0.6B also outperforms TaH+ 0.6B on all three tasks. These results support selective width under both parameter settings.

Appendix C.1 reports the accuracy cost of restricting Imagine+ 0.6B to a one-slot budget.

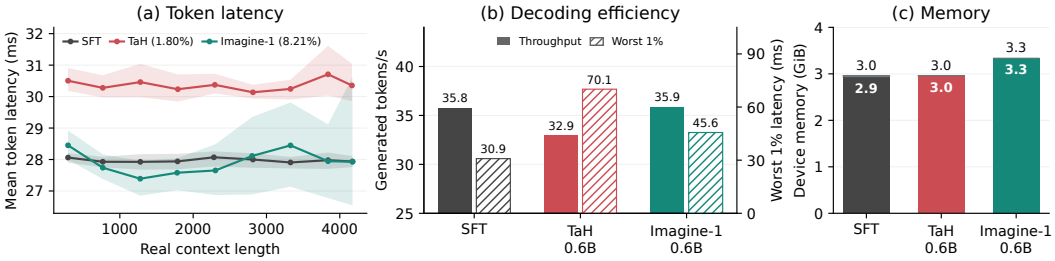


Figure 3: Single-stream inference for parameter-matched models. (a) Mean inter-token latency by real context length; shaded bands are question-bootstrap confidence intervals. Legends show observed activation rates. (b) Decode throughput (solid bars, left axis) and mean latency of the slowest 1% of token intervals (hatched bars, right axis). (c) Mean device memory with the peak-minus-mean increment stacked above it; labels show mean and peak GiB.

Table 2: MATH-500 ablations at maximum length 4096. Imagine+ 0.6B with threshold 0.85 is the reference. Entries report answer accuracy.

Imagine+ 0.6B: 42.4%					
Model structure		*Slot readout*		*Controller target*	
Controlled MoE (CARE-like)	40.8	Logit summation	38.0	Successful correction	38.6
Training setup		*Slot attention*		*Controller input*	
Hard-only	33.0	Non-causal, all visible	36.6	Previous hidden + new embedding	40.4
Refresh oracle each epoch	17.2	Causal, history invisible	37.4	Previous multi-layer hidden	36.4

7.2 Single-Stream Efficiency

Quality improvements matter to an interactive user only alongside the time spent waiting for tokens. We therefore compare single-stream decoding with SFT and parameter-matched selective depth, measuring throughput, latency, and the memory cost of retaining additional states.

Imagine-1 0.6B reduces worst-1% token latency from TaH's 70.091 ms to 45.625 ms, a 34.9% reduction (Figure 3b). This metric averages the slowest 1% of inter-token intervals. It captures long pauses in a single response that mean latency can obscure. The mean-latency reduction is smaller: 8.2%, from 30.36 to 27.89 ms.

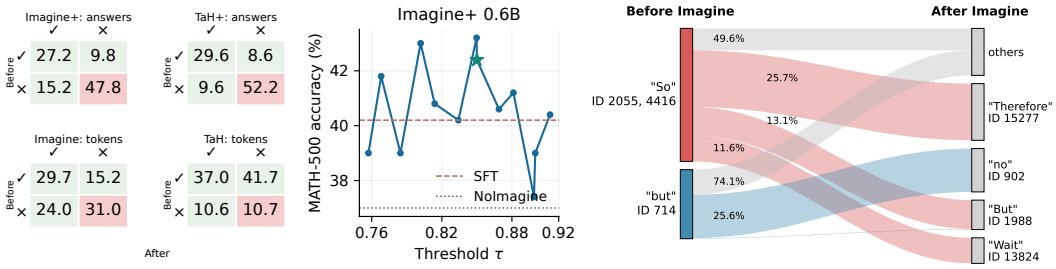
Imagine-1 also sustains 35.86 tokens/s versus TaH's 32.93, an 8.9% throughput advantage. It reduces tail latency despite activating extra computation more frequently (8.21% versus 1.80%). Selective width thus improves both overall decoding speed and the longest token-delivery delays in this comparison.

This benefit has a modest memory cost. Mean device usage is 3.34 GiB for Imagine-1 0.6B and 2.96 GiB for TaH 0.6B. SFT reaches 35.75 tokens/s with 2.94 GiB mean memory. The single-slot model therefore approaches SFT's throughput while adding selective reasoning capacity. Larger slot budgets increase latency, as shown by the full configurations in Appendix C.5.

8 Discussion

Table 2 tests changes to the Imagine+ recipe at length 4096. The full model scores 42.4%. We summarize each change and its effect here. Appendix F gives the detailed analysis.

Expert states and training. We replace persistent slots with ordinary low-rank expert mixtures under the same controller workflow. This CARE-style control performs worse, supporting persistent expert computation. We also replace DHT with hard-only training and refresh reference mismatches every epoch in separate ablations. Both changes reduce accuracy. These results favor dense-to-hard exposure and stable mismatch labels.



(a) Prediction transitions (b) Threshold sweep (c) Token transitions

Figure 4: Selective-width diagnostics. (a) Rows and columns denote correctness before and after additional computation; cells are percentages. The top row uses MATH-500 answers. The bottom row uses Imagine's teacher-forced candidate tokens and TaH's published token transitions (Fu et al., 2026). (b) Imagine+ 0.6B threshold scan, with the reported operating point starred. (c) Changes from *So* and *but* on a 100-question MATH-500 sample. Other-target ribbon widths are scaled by 0.2; labels retain the observed percentages.

Math500 example (ID107). How many elements are in the intersection of the set of all the prime numbers less than 30 and the set of all the odd numbers greater than zero?
Reference answer: 9.

Imagine+ 0.6B ($k = 0$); incorrect
[Earlier text omitted.] Since all primes are odd, the intersection is exactly the set of primes less than 30.
Thus, the number of elements in the intersection is $\boxed{10}$.

Imagine+ 0.6B (controller threshold 0.85); correct
[Earlier text omitted.] The common elements are: 3, 5, 7, 11, 13, 17, 19, 23, 29.
4. **Count of common elements**: - There are 9 common elements.
Thus, the number of elements in the intersection is $\boxed{9}$.

TaH+ 0.6B (controller threshold 0.90); correct
[Earlier text omitted.] The primes less than 30 are:3, 5, 7, 11, 13, 17, 19, 23, 29. There are 9 elements in this set.
Thus,the number of elements in the intersection is $\boxed{9}$.

SFT; incorrect
[Earlier text omitted.] Listing the primes (2, 3, 5, 7, 11, 13, 17, 19, 23, 29) confirms that all of them are odd. Therefore, the intersection of these two sets is simply the set of these primes.
The number of elements in this intersection is 10. $\boxed{10}$

Figure 5: Recorded MATH-500 answer correction. Red text marks changes from each method's no-thinking response.

Communication and readout. We allow unrestricted slot visibility, remove latent history, or replace weighted readout with logit summation. Each change lowers accuracy. The results support ordered communication, access to earlier slot states, and a learned combination of their outputs.

Controller design. We replace minimum-CE labels with successful-correction labels. Accuracy drops. We also test a single preceding hidden state with the current embedding, and preceding multi-layer states alone. Both score below the full controller. The tested recipe benefits from loss-based supervision and richer pre-forward features.

Prediction changes. Figure 4 shows more corrections than regressions and a non-monotonic response to activation. Extra computation therefore needs selective allocation. Figure 5 illustrates one corrected answer. Appendix F relates these observations to similar architectures, the training objective, and inference costs.

9 Conclusion

Imagine combines persistent expert states, pre-forward allocation, and dense-to-hard training. All three variants exceed SFT's average accuracy, and Imagine+ improves on SFT across all three mathematical benchmarks. Single-slot decoding also improves latency and throughput over parameter-matched selective depth. By replacing recurrent reasoning steps with single-forward parallel capacity expansion, Imagine avoids extra sequential backbone passes, making it particularly suitable for small-language-model deployment on edge GPUs and latency-sensitive serving.

REFERENCES

Sangmin Bae, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, and Tal Schuster. Relaxed Recursive Transformers: Effective Parameter Sharing with Layer-wise LoRA. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/54d6a55225cebbdc16fbb0e45c5bdf2b-Abstract-Conference.html.

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, and Se-Young Yun. Mixture-of-Recursions: Learning Dynamic Recursive Depths for Adaptive Token-Level Computation. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 96572–96617. Curran Associates, Inc., 2025b. doi: 10.52202/085713-3229. URL https://papers.nips.cc/paper_files/paper/2025/hash/8b08bbf8b420faa6eeb4020720582ec7-Abstract-Conference.html.

Ruisi Cai, Yeonju Ro, Geon-Woo Kim, Peihao Wang, Babak Ehteshami Bejnordi, Aditya Akella, and Zhangyang Wang. Read-ME: Refactorizing LLMs as Router-Decoupled Mixture of Experts with System Co-Design. In *Advances in Neural Information Processing Systems*, volume 37, pp. 116126–116148, 2024. doi: 10.52202/079017-3687. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/d298cf34e4539f9134db7f38b42f69fe-Abstract-Conference.html.

Mouxiang Chen, Binyuan Hui, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Jianling Sun, Junyang Lin, and Zhongxin Liu. Parallel Scaling Law for Language Models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 118958–118998, 2025. doi: 10.52202/085713-3970. URL https://papers.nips.cc/paper_files/paper/2025/hash/ac6de776b8de8c9aed1d356997eb54b8-Abstract-Conference.html.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal Transformers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyzdRiR9Y7.

Ying Fan, Anej Svete, and Kangwook Lee. Bridging the Gap Between Latent and Explicit Reasoning with Looped Transformers. *arXiv preprint arXiv:2606.31779v2*, 2026. URL https://arxiv.org/abs/2606.31779v2.

Renyu Fu and Guibo Luo. SeLaR: Selective Latent Reasoning in Large Language Models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7073–7087. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.320. URL https://aclanthology.org/2026.acl-long.320/.

Tianyu Fu, Yichen You, Zekai Chen, Guohao Dai, Huazhong Yang, and Yu Wang. Think-at-Hard: Selective Latent Iterations to Improve Reasoning Language Models. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2511.08577.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1380. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3b01972cf31e6fa0fe29e4b8b5c2a0a1-Abstract-Conference.html.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training Language Models With Pause Tokens. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/76917808731dae9e6d62c2a7a6afb542-Abstract-Conference.html.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training Large Language Models to Reason in a Continuous Latent Space. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Itxz7S4Ip3.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

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. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Haozhe Hu, Hao Wu, Peiran Yin, Chao Han, Yunpu Ma, and Xiaoyu Shen. WIDE: Boosting Adaptive LLM Inference via Token-level Dynamic Width Pruning. *arXiv preprint arXiv:2607.28418*, 2026. URL https://arxiv.org/abs/2607.28418.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. Harder Task Needs More Experts: Dynamic Routing in MoE Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12883–12895. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.696. URL https://aclanthology.org/2024.acl-long.696/.

HuggingFaceH4. MATH-500. Dataset card for the MATH subset released with PRM800K, 2024. URL https://huggingface.co/datasets/HuggingFaceH4/MATH-500.

Ranggi Hwang, Jianyu Wei, Shijie Cao, Changho Hwang, Xiaohu Tang, Ting Cao, and Mao Yang. Pre-gated MoE: An Algorithm-System Co-Design for Fast and Scalable Mixture-of-Expert Inference. In *Proceedings of the 51st Annual International Symposium on Computer Architecture*, 2024. URL https://ieeexplore.ieee.org/document/10609634.

Junmo Kang, Leonid Karlinsky, Hongyin Luo, Zhen Wang, Jacob Hansen, James Glass, David Cox, Rameswar Panda, Rogerio Feris, and Alan Ritter. Self-MoE: Towards Compositional Large Language Models with Self-Specialized Experts. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/558a100caa93422df215fadb9e9b1dd7-Abstract-Conference.html.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 3744–3753. PMLR, 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. MixLoRA: Enhancing Large Language Models Fine-Tuning with LoRA-based Mixture of Experts. *arXiv preprint arXiv:2404.15159*, 2024. URL https://arxiv.org/abs/2404.15159.

Jiamin Li, Qiang Su, Yitao Yang, Yimin Jiang, Cong Wang, and Hong Xu. Adaptive Gating in Mixture-of-Experts based Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3577–3587. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.217. URL https://aclanthology.org/2023.emnlp-main.217/.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Aiwei Liu, Cheng Shi, Chuhan Wu, Ci Lei, Di Lu, Donald He, Fan Zhang, Fanhao Kong, Feifei Zhang, Guan Wang, Haicheng Wang, Haoyu Liu, Houjin Yu, Jiachen Ding, Jiayi Feng, Jie Zhou, Jijun Chi, Jindi Shi, Jing Lei, Junjie Zhang, Laiyi Li, Le Tian, Linhao Zhang, Miao Fan, Sijun Zhang, Wei Jia, Weiwei Shi, Wenhan Li, Wentao Zhao, Wenteng Liang, Xiao Zhou, Xiaojin Zhou, Xihuai Wang, Xinyu Gao, Xuanliang Wang, Xuyang Ao, Yang Yu, Yangxiu You, Yinuo Zhao, Yufei Kuang, Yufei Wang, Yuan Liu, Yuan Liu, Yuwen Chen, Zhencong Tian, Zhongyin Zhao, Zilin Yu, and Zitao Wang. Hidden Decoding at Scale: Latent Computation Scaling for Large Language Models. *arXiv preprint arXiv:2607.08186*, 2026. URL https://arxiv.org/abs/2607.08186.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Ré, and Beidi Chen. Deja Vu: Contextual Sparsity for Efficient LLMs at Inference Time. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 22137–22176. PMLR, 2023. URL https://proceedings.mlr.press/v202/liu23am.html.

Charles London and Varun Kanade. Pause Tokens Strictly Increase the Expressivity of Constant-Depth Transformers. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/817c0ed5303e396e803d6a972f83c475-Abstract-Conference.html.

Anton Lozhkov, Hynek Kydlíček, Loubna Ben Allal, Guilherme Penedo, Edward Beeching, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. OpenR1-Math-220k. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/open-r1/OpenR1-Math-220k.

Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. MoELoRA: Contrastive Learning Guided Mixture of Experts on Parameter-Efficient Fine-Tuning for Large Language Models. *arXiv preprint arXiv:2402.12851*, 2024. URL https://arxiv.org/abs/2402.12851.

math-ai. AMC23. Hugging Face dataset, n.d. URL https://huggingface.co/datasets/math-ai/amc23. 40-problem test split from the 2023 American Mathematics Competitions; accessed September 2026.

Kumari Nishu, Sachin Mehta, Samira Abnar, Mehrdad Farajtabar, Maxwell Horton, Mahyar Najibi, Moin Nabi, Minsik Cho, and Devang Naik. From Dense to Dynamic: Token-Difficulty Driven MoEfication of Pre-Trained LLMs. In *International Conference on Learning Representations*, 2025. URL https://machinelearning.apple.com/research/dense-to-dynamic.

Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's Think Dot by Dot: Hidden Computation in Transformer Language Models. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=NikbrdtYvG.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-Depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024. URL https://arxiv.org/abs/2404.02258.

Tom Saliencro, Rohan Desai, Priya Nair, Maya Lindqvist, and Daniel Whitmore. Spend Experts Where You Are Unsure: Confidence-Adaptive Routing for Mixture-of-Experts LoRA. *arXiv preprint arXiv:2607.26052v2*, 2026. URL https://arxiv.org/abs/2607.26052v2.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident Adaptive Language Modeling. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17456–17472, 2022. doi: 10.52202/068431-1269. URL https://papers.neurips.cc/paper_files/paper/2022/hash/6fac9e316a4ae75ea244ddcef1982c71-Abstract-Conference.html.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing Chain-of-Thought into Continuous Space via Self-Distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

Jiakai Tang, Xu Chen, Wen Chen, Jian Wu, Yuning Jiang, and Bo Zheng. Parallel Latent Reasoning for Sequential Recommendation. *arXiv preprint arXiv:2601.03153*, 2026. URL https://arxiv.org/abs/2601.03153.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Bohong Wu, Mengzhao Chen, Xiang Luo, Shen Yan, Qifan Yu, Fan Xia, Tianqi Zhang, Hongrui Zhan, Zheng Zhong, Xun Zhou, Siyuan Qiao, and Xingyan Bin. Parallel Loop Transformer for Efficient Test-Time Computation Scaling. *arXiv preprint arXiv:2510.24824*, 2025a. URL https://arxiv.org/abs/2510.24824.

Bohong Wu, Shen Yan, Sijun Zhang, Jianqiao Lu, Yutao Zeng, Ya Wang, and Xun Zhou. Efficient Pretraining Length Scaling. *arXiv preprint arXiv:2504.14992*, 2025b. URL https://arxiv.org/abs/2504.14992.

Haoyi Wu, Zhihao Teng, and Kewei Tu. Parallel Continuous Chain-of-Thought with Jacobi Iteration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2025c. doi: 10.18653/v1/2025.emnlp-main.47. URL https://aclanthology.org/2025.emnlp-main.47/.

Xun Wu, Shaohan Huang, Wenhui Wang, Shuming Ma, Li Dong, and Furu Wei. Multi-Head Mixture-of-Experts. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. doi: 10.52202/079017-2984. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ab05dc8bf36a9f66edbff6992ec86f56-Abstract-Conference.html.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of LoRA Experts. In *International Conference on Learning Representations*, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/ce806d8b4bf38cd92d483e5a0490d983-Abstract-Conference.html.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Mingyu Xu, Kuai Yu, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. mHC: Manifold-Constrained Hyper-Connections. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/61870.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng

Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermiş, Acyr Locatelli, and Sara Hooker. Pushing Mixture of Experts to the Limit: Extremely Parameter Efficient MoE for Instruction Tuning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6d00071564ec447466fc4577743cf1b3-Abstract-Conference.html.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping Reasoning With Reasoning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/639a9a172c044fbb64175b5fad42e9a5-Abstract-Conference.html.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-STaR: Language Models Can Teach Themselves to Think Before Speaking. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=oRXPiSOGH9.

Boyi Zeng, Yiqin Hao, He Li, Shixiang Song, Feichen Song, Zitong Wang, Siyuan Huang, Yi Xu, ZiWei He, Xinbing Wang, and Zhouhan Lin. Pretraining with Token-Level Adaptive Latent Chain-of-Thought. *arXiv preprint arXiv:2602.08220*, 2026. URL https://arxiv.org/abs/2602.08220.

Xiangdong Zhang, Xiaohan Qin, Sunan Zou, Tuo Dai, Xiaoming Shi, Huaijin Wu, Yebin Yang, Zhuo Xia, Shaofeng Zhang, Lin Yao, Yuliang Liu, Yu Cheng, and Junchi Yan. xHC: Expanded Hyper-Connections. *arXiv preprint arXiv:2607.14530*, 2026. URL https://arxiv.org/abs/2607.14530.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, and Xin Eric Wang. Soft Thinking: Unlocking the Reasoning Potential of LLMs in Continuous Concept Space. In *Advances in Neural Information Processing Systems*, volume 38, pp. 168990–169012. Curran Associates, Inc., 2025. doi: 10.52202/085713-5629. URL https://papers.nips.cc/paper_files/paper/2025/hash/f7396d1c54d51416958d63e285377103-Abstract-Conference.html.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-Connections. In *International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f1e8ff97057e0c3f6bcb7018c78d4df1-Abstract-Conference.html.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, Lu Li, Jiajun Shi, Kaijing Ma, Shanda Li, Taylor Kergan, Andrew Smith, Xingwei Qu, Mude Hui, Bohong Wu, Qiyang Min, Hongzhi Huang, Xun Zhou, Wei Ye, Jiaheng Liu, Jian Yang, Yunfeng Shi, Chenghua Lin, Enduo Zhao, Tianle Cai, Ge Zhang, Wenhao Huang, Yoshua Bengio, and Jason Eshraghian. Scaling Latent Reasoning via Looped Language Models. *arXiv preprint arXiv:2510.25741*, 2025b. URL https://arxiv.org/abs/2510.25741.