ICML 2026 · Seoul, South Korea

DTop-p MoE

Sparsity-Controlled Dynamic Top-*p* MoE for Foundation Model Pre-training

Can Jin[*1] Hongwu Peng[*2] Mingcan Xiang[23] Qixin Zhang[4] Xiangchi Yuan[5] Amit Hasan[2]
Ohi Dibua[2] Yifan Gong[2] Yan Kang[2†] Dimitris N. Metaxas[1†]

[1]Rutgers University · [2]Adobe Research · [3]UMass Amherst · [4]Nanyang Technological University · [5]Georgia Tech | [*]Equal Contribution · [†]Equal Advising

Presented by **Can Jin** · Rutgers University · can.jin@rutgers.edu

BACKGROUND

Sparse Mixture-of-Experts (MoE)

Scaling Foundation Models with dense compute is prohibitively expensive. **Sparse MoE** activates only a small subset of experts per token — **decoupling total parameters from compute cost.**

Top-*k* routing

Select a **fixed** number of *k* experts per token. Predictable compute, but the **same capacity for every token** — ignores token difficulty and layer-specific needs.

Top-*p* routing

Select experts until cumulative probability exceeds threshold *p* (nucleus sampling). **Adaptive** per token — but how reliable is it?

This talk: can Top-*p* routing be made both **compute-controllable** *and* **more effective** than Top-*k*?

MOTIVATION

Fixed-Threshold Top-*p* is Unstable

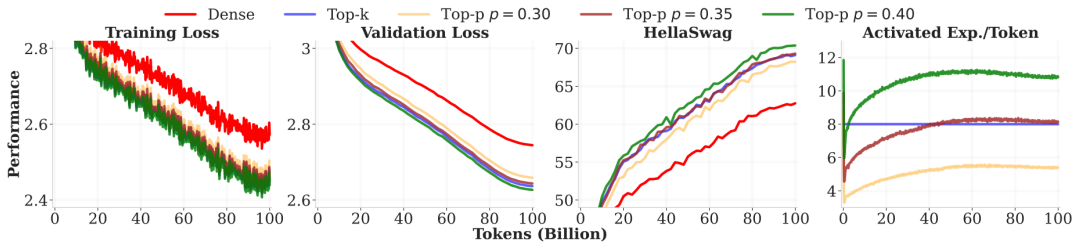

Figure 2. Performance comparison of the Dense model, Top-k MoE, and fixed-threshold Top-p MoE ($p \in \{0.30, 0.35, 0.40\}$). Top-$p$ yields only marginal gains over Top-k MoE at comparable activation levels, while the number of activated experts fluctuates unpredictably.

❶ Uncontrolled compute

Activated-expert count **fluctuates unpredictably** during training — incompatible with strict pre-training FLOP budgets (OOM risk).

❷ Hypersensitive to *p*

p = 0.40 over-activates (>12 experts); p = 0.35 only **matches Top-*k***. Gains are marginal and tuning is costly.

KEY IDEA

Make the Threshold a Control Setpoint

💡 Insight

Top-*p* is naturally adaptive, but the threshold gets **no gradient** (it only binarizes the expert mask). So treat **target sparsity as a setpoint** and make the threshold *effectively learnable* via feedback control.

Contributions:

- **Analysis:** fixed-threshold Top-*p* gives only marginal gains over Top-*k* with uncontrolled cost.
- **DTop-*p* MoE:** a PI controller + Dynamic Routing Normalization → adaptive routing under a strict global budget (plus a layer-wise variant).
- **Comprehensive study** on NLP & CV — better performance and scaling at matched FLOPs.

DTop-*p* MoE — Overview

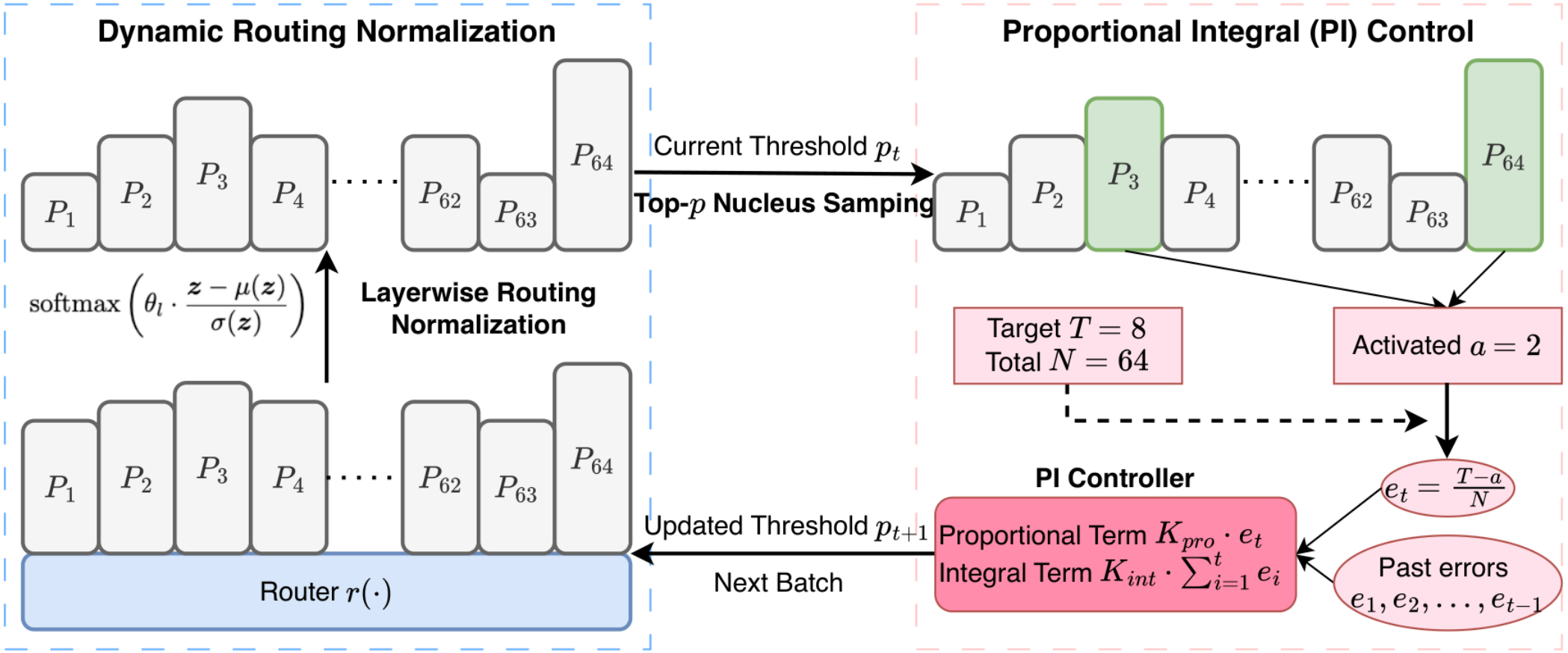

Figure 1. Overview of DTop-*p* MoE. We employ a Proportional-Integral (PI) controller to dynamically adjust the global probability threshold, aligning the number of activated experts with a target value. The Dynamic Routing Normalization modulates layer-wise logit distributions to support varying sparsity needs, enabling distinct patterns across network depths under the global threshold.

METHOD ①

PI Controller — Learnable Sparsity

Track average activated experts a_t per batch; define sparsity error $e_t = (T - a_t) / N$. A discrete Proportional-Integral law updates the threshold:

$$p_{t+1} = p_0 + \underbrace{K_{pro} \cdot e_t}_{\text{Proportional}} + \underbrace{K_{int} \cdot \sum_{i=1}^{t} e_i}_{\text{Integral}} \qquad (6)$$

Proportional term

Reacts immediately to the current deviation from target T.

Integral term

Accumulates past errors → removes steady-state bias, so a_t **converges to** T.

Relies on the monotonicity of nucleus sampling: raising p ⇒ more experts. No gradient needed for the threshold.

METHOD ②

Dynamic Routing Normalization

A single global threshold assumes uniform logit statistics across depth. Instead, normalize each layer's logits and apply a **learnable per-layer scale** θ_l:

$$\boldsymbol{P}(\boldsymbol{x}) = \mathrm{softmax}\left(\theta_l \cdot \frac{\boldsymbol{z} - \mu(\boldsymbol{z})}{\sigma(\boldsymbol{z})}\right), \text{with } \boldsymbol{z} = \boldsymbol{W}\boldsymbol{x} \quad (7)$$

- Large θ_l → **sharper** distribution → fewer experts; small θ_l → **flatter** → more experts.
- Each layer learns a **distinct sparsity pattern** while still respecting the single global budget.

Training Dynamics (100B tokens)

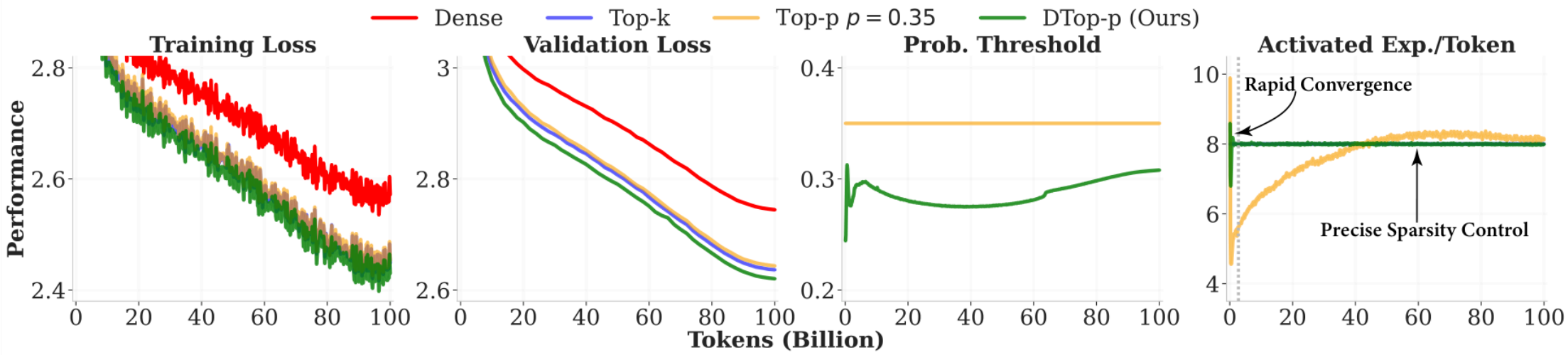

Figure 3. Training and validation performance of the Dense-1.3B and MoE-1.3B-6.9B-64E8A models using Top-k , Top-p , and DTop-p routing on NLP tasks. DTop-p achieves the best overall performance.

DTop-*p* reaches the **best train/val loss** on MoE-1.3B-6.9B-64E8A *and* locks the activated-expert count to **T = 8** within ~1B tokens — fixed Top-*p* overshoots and oscillates.

Downstream Benchmarks (13 datasets)

Table 2. Inference performance comparison between Dense-1.3B and MoE-1.3B-6.9B-64E8A models with Top-k, Top-p, and DTop-p MoE trained on 100B tokens. **Bold** indicates the best performance across all settings. Numbers in parentheses indicate the number of few-shot examples used in evaluation. DTop-p MoE achieves the highest average performance.

Benchmark	Dense-1.3B	MoE-1.3B-6.9B-64E8A		
	Dense	Top-k	Top-p	DTop-p (Ours)
SVAMP(5)	5.3	10.3	8.3	**16.0**
MMLU(5)	25.2	26.6	26.8	**27.4**
ARC-Easy(0)	60.9	65.7	65.5	**67.1**
ARC-Challenge(0)	34.4	40.6	40.9	**41.7**
COPA(5)	69.0	82.0	82.0	**85.0**
PIQA(5)	75.7	**78.9**	77.7	78.1
HellaSwag(0)	62.7	69.1	69.2	**70.9**
WinoGrande(5)	62.7	64.0	66.3	**67.2**
LAMBADA(5)	56.7	61.5	**63.5**	62.5
BoolQ(5)	55.4	63.9	63.9	**65.4**
AGIEval-LSAT-RC(5)	26.2	22.7	23.8	**27.2**
AGIEval-LSAT-LR(5)	26.0	**26.4**	24.9	25.5
AGIEval-SAT-EN(5)	26.5	24.8	**27.7**	**27.7**
Average	45.1	49.0	49.3	**50.9**

+1.9%

average gain over **Top-*k*** MoE
at **matched average FLOPs**

- Best average across the 13 zero/few-shot tasks (**50.9** vs 49.0 / 49.3).
- Large gains on reasoning: **SVAMP +5.7**, **COPA +3.0** over Top-k.

Generalizes to Diffusion Transformers

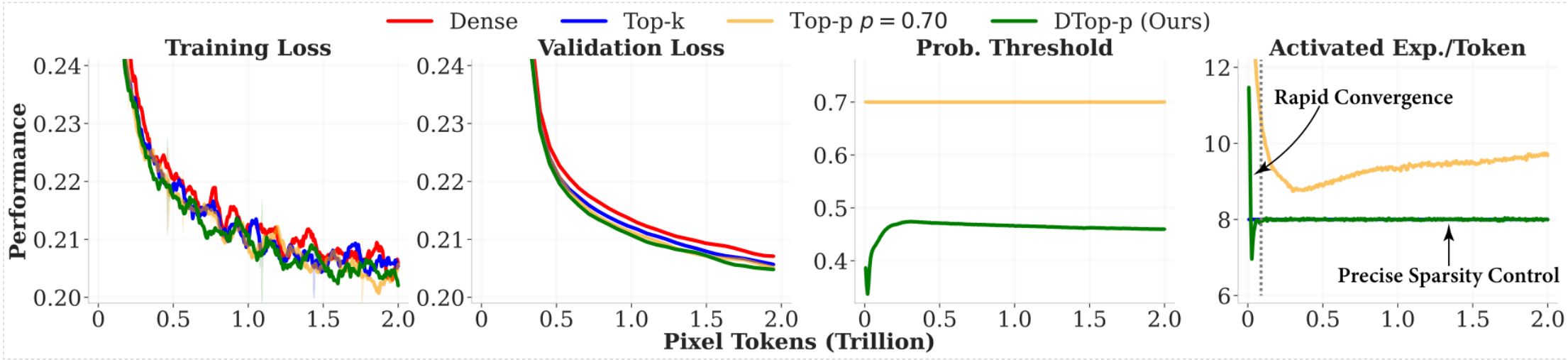

Figure 4. Training and validation performance of the 0.9B Dense model versus the 64E8A MoE model (0.9B activated / 3.4B total parameters) using Top-k , Top-p , and DTop-p MoE on CV tasks. DTop-p achieves the best performance.

On a **0.9B → 3.4B** 64E8A MoE DiT trained on **2T pixel tokens**, **DTop-*p*** again achieves the lowest validation loss while holding precise sparsity — the method is **not NLP-specific**.

ANALYSIS

Precise & Adaptive Sparsity Control

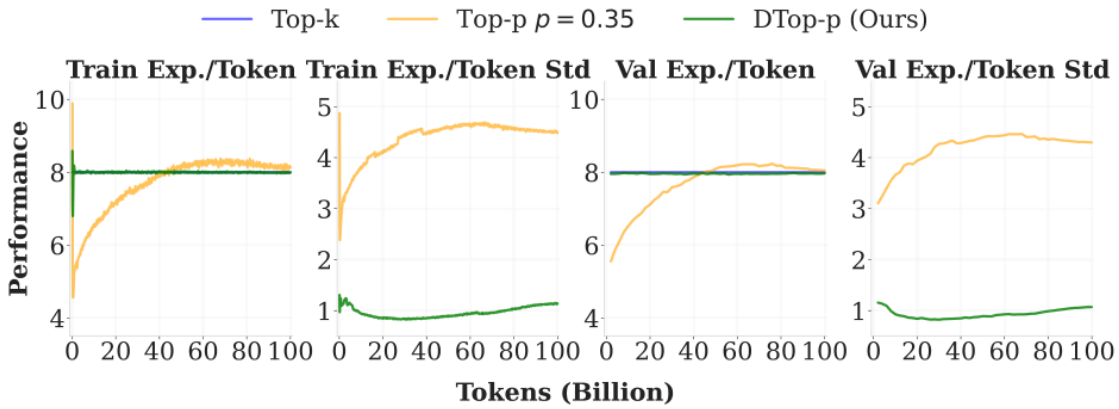

Figure 6. Mean and standard deviation of activated experts per token on training and validation sets for Top-k , Top-p , and DTOP-p MoE. DTOP-p effectively and rapidly converges to the target activation level on both training and validation datasets.

Precise

Converges to **T = 8** with low variance **(σ ≈ 1)**; fixed Top-p drifts with σ ≈ 4.

Adaptive (hierarchical)

Learns to use **fewer experts in shallow layers**, **more in deep layers** — emergent depth-wise specialization.

ANALYSIS

Ablation: Both Components Matter

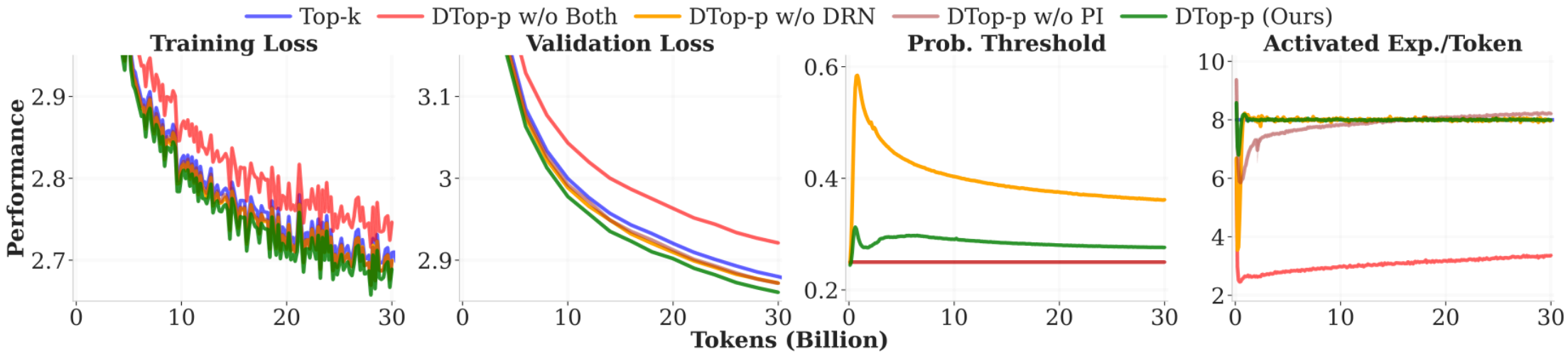

Figure 7. Ablation study of the PI controller (PI) and Dynamic Routing Normalization (DRN).

PI controller — without it, sparsity is unregulated and drifts.

DRN — adaptively rescales layer logits; best loss only with **both** combined.

ANALYSIS

Robust Scaling (0.4B → 2.4B)

Table 8. Inference performance of Dense models vs. varying 64E8A MoE models across different model sizes (0.4B, 1.3B, 2.4B) trained on 100B tokens. DTOP-p consistently achieves the best performance and scales effectively with model size.

Area	Benchmark	Dense-0.4B	MoE-0.4B-3.7B		Dense-1.3B	MoE-1.3B-6.9B		Dense-2.4B	MoE-2.4B-13.6B	
		Dense	Top-k	DTOP-p	Dense	Top-k	DTOP-p	Dense	Top-k	DTOP-p
Symbolic Problem Solving	SVAMP(5)	6.3	4.0	6.6	5.3	10.3	16.0	11.0	13.6	**22.6**
World Knowledge	MMLU(5)	25.0	23.9	24.8	25.2	26.6	27.4	24.6	28.2	**29.0**
	ARC-Easy(0)	52.9	61.3	62.1	60.9	65.7	67.1	64.7	70.2	**70.7**
	ARC-Challenge(0)	26.9	32.5	33.7	34.4	40.6	41.7	37.7	**45.5**	44.6
Commonsense Reasoning	COPA(5)	63.0	72.0	74.0	69.0	82.0	85.0	80.0	84.0	**87.0**
	PIQA(5)	71.2	74.5	75.0	75.7	78.9	78.1	77.2	79.1	**79.5**
Language Understanding	HellaSwag(0)	48.8	60.6	60.7	62.7	69.1	70.9	67.9	73.9	**74.4**
	WinoGrande(5)	55.8	59.7	59.9	62.7	64.0	67.2	67.4	70.6	**71.8**
	LAMBADA(5)	46.8	54.5	56.2	56.7	61.5	62.5	63.8	68.6	**69.4**
Reading Comprehension	BoolQ(5)	58.9	62.4	64.1	55.4	63.9	65.4	69.2	68.0	**71.2**
	AGIEval-LSAT-RC(5)	21.2	23.9	23.9	26.2	22.7	**27.2**	24.2	24.2	26.2
	AGIEval-LSAT-LR(5)	24.5	22.1	23.5	26.0	**26.4**	25.5	24.5	22.1	26.1
	AGIEval-SAT-EN(5)	23.7	22.3	23.3	26.5	24.8	27.7	27.6	27.7	**30.5**
Average		40.4	44.1	45.2	45.1	49.0	50.9	49.2	52.0	**54.1**

DTop-*p* wins at every model size — and the advantage over Top-k **widens with scale**. It also benefits more from finer

RECAP

The Full Training Loop

Algorithm 1 DTop-p MoE

Require: Dataset $\mathcal{D}$, target expert T, initial probability threshold p_0, PI gain coefficient K_{pro}, K_{int}, model parameters Θ (including dynamic scales $\{\theta_l\}_{l=1}^{L}$)

Initialize integral error accumulation $e_{sum} \leftarrow 0$
Initialize probability threshold $p_1 = p_0$
for step $t = 1, 2, \ldots$ with batch $\mathcal{B}_t \in \mathcal{D}$ **do**
  Accumulator for number of activated experts $a_{sum} \leftarrow 0$
  Forward Pass:
  for layer $l = 1$ to L **do**
    Compute raw logits for input representation $\boldsymbol{x}$: $\boldsymbol{z} \leftarrow \boldsymbol{W}\boldsymbol{x}$
    Dynamic Routing Normalization (Equation 7): $\boldsymbol{P} \leftarrow \mathrm{softmax}\left(\theta_l \cdot \frac{\boldsymbol{z}-\mu(\boldsymbol{z})}{\sigma(\boldsymbol{z})}\right)$
    Nucleus Sampling with global threshold p_t:
      Select minimal set of experts S such that $\sum_{i \in S} P_i \geq p_t$
      $r_i(\boldsymbol{x}) \leftarrow \frac{P_i}{\sum_{j \in S} P_j}$ for $i \in S$, else 0 (Equation 4)
    Record number of activated experts: $a_{sum} \leftarrow a_{sum} + |S|$
    Compute layer output: $\boldsymbol{y} \leftarrow \sum_{i \in S} r_i(\boldsymbol{x}) E_i(\boldsymbol{x})$
  end for
  PI controller Update (Equation 6):
  Calculate average activation per token: $a_t \leftarrow a_{sum}/(L \cdot |\mathcal{B}_t|)$, $|\mathcal{B}_t|$ is total tokens in $\mathcal{B}_t$
  Calculate sparsity error: $e_t \leftarrow (T - a_t)/N$
  Update integral term: $e_{sum} \leftarrow e_{sum} + e_t$
  Update global threshold:
    $p_{t+1} \leftarrow p_0 + K_{pro} \cdot e_t + K_{int} \cdot e_{sum}$
  Clip p_{t+1} to range $(0, 1)$
  Optimization:
  Compute Total Loss $\mathcal{L}$
  Update parameters Θ via gradient descent
end for

- **Forward:** per layer, normalize logits (DRN) → nucleus-sample experts at threshold p_t.
- **Feedback:** measure a_t, compute error, update p_{t+1} via PI.
- **Cost:** only a lightweight scalar signal on top of standard MoE training.

CONCLUSION

Takeaways

+1.9%
avg. over Top-*k*
(NLP, matched FLOPs)

σ ≈ 1
stable activated-
expert count

LLM + DiT
wins on both
NLP & vision

- **DTop-*p*** makes Top-*p* routing **compute-controllable** via PI control — no gradient on the threshold.
- **Dynamic Routing Normalization** unlocks per-layer adaptive sparsity under one global budget.
- Beats **Top-*k*** & **fixed Top-*p*** with **robust scaling** across granularity, model & data size.

Thank You! Questions?

DTop-*p* MoE: Sparsity-Controlled Dynamic Top-*p* MoE for Foundation Model Pre-training

Presented by **Can Jin** · Rutgers University | **can.jin@rutgers.edu**

ICML 2026 · PMLR 306 | Rutgers University & Adobe Research