CONTROL-MOE: MIXTURE OF EXPERTS WITH CONTROLLABLE DYNAMIC SPARSITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-Experts (MoE) has become a foundational approach for scaling large models by selectively activating parameters within multi-layer perceptron (MLP) layers. However, conventional expert selection typically relies on a single router per layer, often resulting in an inflexible activation rate. Moreover, fixing the number of experts per token further limits the adaptability of network sparsity. We propose Control-MoE, a routing strategy centered on controllable sparsity. Rather than depending on fixed expert-count rules, Control-MoE employs a dynamic router with flexible sparsity, implemented through parallel Top-1 routers for expert assignment in each layer. The router design is deliberately simple, serving primarily to provide high flexibility. By incorporating a skip expert that bypasses expert computation, Control-MoE enables highly flexible activation levels, ranging from zero to a high upper bound, and allows the model to select an efficient number of experts while maintaining performance. To regulate overall sparsity automatically, we introduce a Model Predictive Control (MPC) framework that dynamically adjusts the skip-loss weight, steering the realized bypass probability toward a target setpoint without manual tuning. Comprehensive evaluations show that Control-MoE improves both model performance and operational efficiency. Across different model scales, Control-MoE achieves an average 2% improvement over Top-K router baselines at comparable activation levels. It also exhibits test-time scaling capability, allowing the average number of activated parameters to be dynamically adjusted.

1 INTRODUCTION

Large language models (LLMs) have achieved remarkable breakthroughs (DeepSeek-AI et al., 2026; Team et al., 2026; Yang et al., 2025) through scaling data and capacity, but at a substantial computational cost. To mitigate this, Mixture-of-Experts (MoE) architectures leverage conditional computation to expand model capacity while sustaining feasible inference, and their integration into Transformers is now ubiquitous in contemporary LLMs, yielding extraordinary gains in both scale and efficacy. Within these frontier systems, MoE designs have largely converged on the Top-K routing strategy, wherein each expert is assigned a score (typically via softmax or sigmoid) and only the K highest-scoring experts are activated. However, because K remains a fixed architectural hyperparameter throughout training and inference, this approach enforces a rigid, uniform activation count for every token, inevitably leading to suboptimal resource allocation: computationally trivial tokens (e.g., punctuation marks) consume just as many experts as semantically dense ones (e.g., Chinese characters). Ideally, the router should dynamically adjust the number of activated experts based on each token's complexity, and it is precisely this principle that underpins the emerging dynamic activation MoE frameworks.

Broadly, existing strategies for dynamic activation in MoE can be grouped into three families. The first infers token complexity directly from internal model signals, such as router logits or attention scores, and uses these signals to decide how many experts to activate (Huang et al., 2024; Jin et al., 2025; Aghdam et al., 2024). The second augments the standard Top-K router with an auxiliary "difficulty sensor" that explicitly predicts the per-token activation count (Chen & Zhang, 2026; Guo et al., 2025; Tao et al., 2026; Wu et al., 2026). The third introduces specialized experts, such as zero-computation or null experts, to regulate sparsity indirectly (Jin et al., 2024; Zeng et al., 2024; Team et al., 2025). Despite their diversity, these approaches suffer from common limitations:

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.

they either enforce a fixed minimum activation floor that constrains sparsity flexibility, or require laborious hyperparameter tuning to achieve a desired sparsity level, and often compromise model performance under high-sparsity regimes. These shortcomings motivate us to design a **flexible and controllable** routing strategy that not only unlocks the full potential of highly sparse MoE architectures but also minimizes the need for manual tuning.

To this end, we propose Control-MoE, a routing strategy centered on controllable sparsity. To achieve it, we use a dynamic router with flexible sparsity, whose design is deliberately simple and intended to facilitate controllability rather than impose fixed expert-count rules. The router comprises K independent gates: for each input token, these gates operate in parallel, with each gate independently selecting its single top-ranked expert (Top-1) from a candidate set augmented by a low-cost "skip" expert that incurs negligible computational overhead. During training, we introduce an auxiliary sparsity loss (skip-loss) to encourage the gates to favor the skip expert, and crucially, different gates are allowed to select the same expert, enabling Control-MoE to dynamically align computational effort with token complexity. In an extreme scenario, all K gates concur to route the token to the skip expert, producing an effect akin to the Mixture-of-Depths (MoD) (Raposo et al., 2024) approach within the expert layer, while conversely, a semantically rich token may activate up to K distinct normal experts, ensuring adequate representational capacity.

To automate sparsity regulation, Control-MoE employs a Model Predictive Control (MPC) (García et al., 1989b) framework that dynamically adjusts the skip-loss weight to steer the actual skip probability toward a target setpoint. In essence, MPC views the skip-loss weight as the control signal and the measured skip probability as the output, forming a feedback loop around the target sparsity: it observes the current response, predicts how future weight changes will affect it, plans a short sequence of adjustments, executes only the first step, and repeats. Concretely, the controller relies on **online linearization** of the router's response through **recursive model identification**, which fits an Autoregressive Exogenous Input (ARX) model to forecast the effect of future weight adjustments on the output. At every control interval, it solves a finite-horizon optimization to compute a sequence of future weights that minimizes tracking error while penalizing abrupt changes, and then applies only the first element of this sequence in a receding-horizon manner. Executed independently per MoE layer, this strategy enables robust and automated sparsity control without manual tuning.

We evaluate Control-MoE on OLMoE-based architectures Muennighoff et al. (2024) ranging from 2B to 10B parameters across multiple activation rates and target sparsity levels. The results show that Control-MoE provides flexible and controllable sparsity, improving the trade-off between computation and accuracy. Moreover, Control-MoE supports test-time activation scaling: its activation level can be adjusted after training to serve different inference budgets without retraining the model.

In summary, the main contributions of this paper are

- **Architectural Flexibility**: The dynamic router with flexible sparsity is a plug-in module that can be integrated into any existing MoE architecture without structural redesign.
- **Learnable and Controllable Sparsity**: Through a tunable skip loss and MPC controller, Control-MoE provides granular and accurate control over activation sparsity during training and inference.
- **Token Adaptive Computation**: The dynamic router successfully implements token-adaptive computation in practice. The Control-MoE offers the flexibility to tune the activation level at testing time. Our results demonstrate a clear and systematic discrepancy in expert allocation between hard and easy tokens.

2 Related Works

Extremely Sparse MoE Architectures Recent large-scale Mixture-of-Experts (MoE) models combine massive total capacity with activation rates of only 3–5% per token. Examples include DeepSeek-V4 (3.1%) DeepSeek-AI et al. (2026), Kimi K3 (3.7%) Team et al. (2026), and the Qwen and MiniMax families (about 4%) Yang et al. (2025); MiniMax AI (2026), compared with approximately 19% for OLMoE Muennighoff et al. (2024). This separation of total and active parameters expands expert capacity without proportional inference cost; Section A and Table 3 provides a broader comparison. However, most extremely sparse MoEs retain fixed Top-K routing, forcing every token to activate the same number of experts regardless of its computational needs.

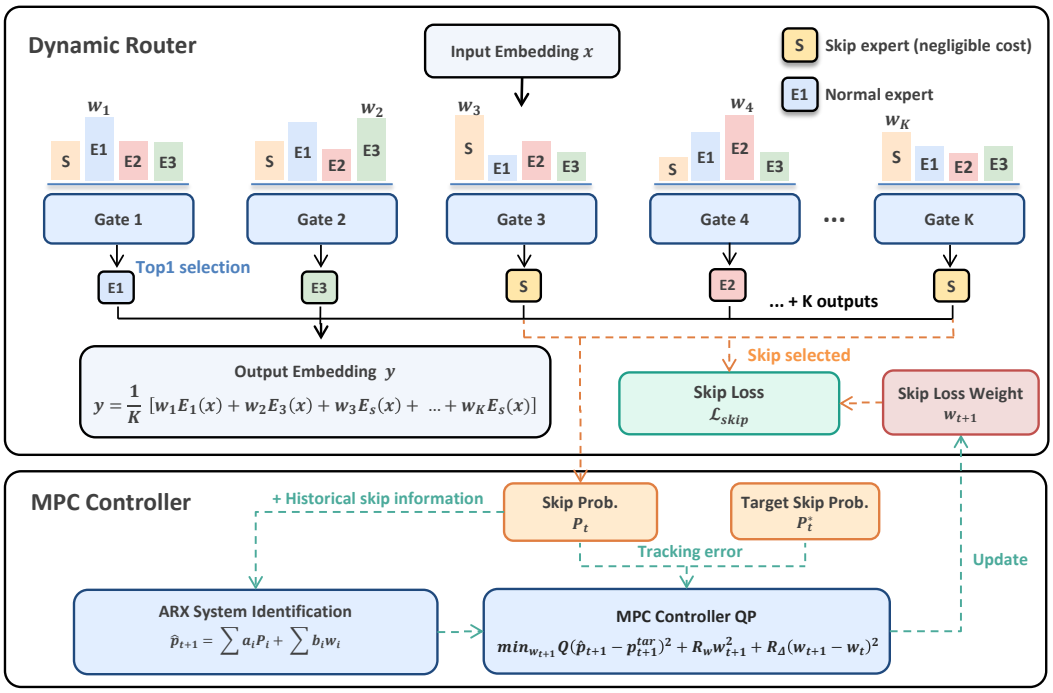

Figure 1: Control-MoE routing structure with skip expert

Dynamic Activation MoE Dynamic activation methods tailor expert count to token difficulty and fall into three families. The first infers difficulty from internal signals: Top-P (Huang et al., 2024) and DTop-P (Jin et al., 2025) stop selection when cumulative router probability reaches a threshold, while DA-MoE (Aghdam et al., 2024) derives the expert count from contextual attention. The second adds a difficulty predictor to Top-K routing. TDMoE (Chen & Zhang, 2026) predicts K per token and layer, DynaMoE (Guo et al., 2025) varies activation between one expert and a layer-specific maximum, DARE (Tao et al., 2026) uses log-perplexity and learned thresholds, and BEAM (Wu et al., 2026) learns token-level binary activation masks. The third uses zero-computation or lightweight experts: MoE++ (Jin et al., 2024) introduces zero, copy, and constant experts, AdaMoE (Zeng et al., 2024) adds null experts, and LongCat (Team et al., 2025) uses a similar mechanism for dynamic capacity. These approaches nevertheless impose a minimum activation floor, require sensitive sparsity tuning, or lack a systematic way to control the computation–accuracy trade-off.

Dynamics Control Feedback control provides a complementary approach to adaptive routing. Washbourne et al. Washbourne et al. (2026), for example, use a PID-inspired bias update based on deviations from uniform expert allocation. PID control Minorsky. (1922) is widely used for tracking and disturbance rejection in physical systems liang Song et al. (2016); Golcuk (2022); Yeung & Huang (2003), but it requires gain tuning and does not explicitly optimize multivariable systems under constraints Åström & Hägglund (2001). LQR and LQG incorporate system dynamics and performance objectives, yet their classical forms do not directly handle hard input and state constraints Qin & Badgwell (2003). Model predictive control instead combines prediction, feedback, and constraint handling in a receding-horizon optimization, at the cost of model dependence and online computation Richalet et al. (1978); García et al. (1989a); Mayne et al. (2000). These properties make MPC a natural fit for regulating MoE sparsity across changing training conditions.

3 METHOD

The Control-MoE routing strategy comprises four components: the dynamic routing mechanism with flexible sparsity, the sparsity controller and the load balancing strategy. The dynamic routing mechanism employs K independent gates operating in parallel per token, each performing Top-1 expert selection. The skip expert incurs negligible cost, enabling gates to bypass heavy computation

when accuracy suffices. The load balancing strategy encourages skip expert usage and uniform activation among standard experts for stable training and efficient allocation. Figure 1 shows the main routing design and sparsity controller.

3.1 Dynamic Routing with Flexible Sparsity

Dynamic Routing Mechanism The Control-MoE routing is designed to systematically to achieve a level of activation flexibility that traditional Top-K routing cannot provide. In essence, it replaces the conventional single router with K independent parallel gates, each performing a Top-1 selection from a total of N experts plus a skip expert. The final output for a token $\boldsymbol{x}$ is the average sum of the outputs of the experts selected across all gates

$$\boldsymbol{y} = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{N} g_k^i E_i(\boldsymbol{x}), \tag{1}$$

where g_i^k is the score from the k-th gate to the i-th expert in a layer. It is computed as

$$g_k^i = \begin{cases} \mathrm{softmax}(G_k^i(\boldsymbol{x})), & \text{if } i = \mathrm{argmax}(G_k(\boldsymbol{x}) + \boldsymbol{b}) \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Here, $G_k(\cdot)$ is a learned linear projection for the k-th gate. The bias $\boldsymbol{b} = [b_{\text{skip}}, \boldsymbol{b}_{\text{load}}] \in \mathcal{R}^{N+1}$ is a shared bias vector used for load balancing among experts in a layer. Each expert, including the skip expert, has a bias. The following sections will elaborate b_{skip} and $\boldsymbol{b}_{\text{load}}$. The gate sum is normalized by K to preserve output scale; with no exclusivity constraint, multiple gates may select the same expert, amortizing cost and reducing FLOPs, so decomposing routing into K independent Top-1 choices that may coincide removes the activation lower bound and yields maximal flexibility.

Skip Expert and Skip Loss To achieve dynamic and token-adaptive computation, we extend the expert pool with a minimal-cost skip expert, accessible to all K parallel gates. This dedicated expert can be realized as a lightweight element-wise weighting:

$$E_{\text{skip}}(\boldsymbol{x}) = \boldsymbol{p} \circ \boldsymbol{x} + \boldsymbol{q}, \tag{3}$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ are two learnable weight and bias vectors for this skip expert. To explicitly regulate the activation sparsity of the model, we introduce a skip loss to encourage choosing the skip expert. For a given token and gate k, the score assigned to the skip expert is derived from the raw gating logits prior to the application of bias, formally:

$$\begin{aligned} \mathcal{L}_{\text{skip}} &= -w_{\text{skip}} \frac{1}{K}\sum_{k=1}^{K}\left(1 - g_k^{\text{skip}}(\boldsymbol{x})\right)^2, \\ g_k^{\text{skip}}(\boldsymbol{x}) &= \mathrm{softmax}\big(G_k^1(\boldsymbol{x}), \ldots, G_k^{N+1}(\boldsymbol{x})\big)_{\text{skip}} = \frac{\exp\left(G_k^{\text{skip}}(\boldsymbol{x})\right)}{\sum_{i=1}^{N+1}\exp\left(G_k^i(\boldsymbol{x})\right)}. \end{aligned} \tag{4}$$

where w_{skip} denotes a tunable coefficient governing the intensity of the skipping incentive. During training, this loss is averaged across all tokens in the batch. By penalizing low skip scores g_k^{skip}, it encourages the routers to direct simpler tokens to the low-cost skip expert. Importantly, w_{skip} acts as a knob that controls the overall sparsity of the model during training: by adjusting this value online through the controller introduced in Section 3.2, the target sparsity can be tracked without human parameter tuning.

3.2 Dynamic Sparcity MPC Controller

To enable automatic and robust regulation of model sparsity, we formulate the control problem in a Model Predictive Control (MPC) framework. At a high level, MPC treats the quantity to be regulated as the output of a dynamical system, uses a model to predict how future control inputs will affect it, solves a finite-horizon optimization to plan a sequence of actions that balances tracking error against abrupt changes, and then applies only the first action in a receding-horizon manner. Here, the core idea is to treat the router's skip probability as the controlled output and the skip-loss weight as the control input, steering the former toward a target setpoint by adjusting the latter. The controller operates in discrete time and is executed independently per MoE layer. For notational brevity, we omit the subscript skip and write w_t to represent the skip-loss weight at time t.

Algorithm 1 Control-MoE MPC Control Step (One-Step)

Require: p_t, w_t, target p^*, history buffer $\mathcal{H}_t$, costs Q, R_w, R_Δ, constraints.
1: Compute setpoint p_{t+1}^{eff} with a designated schedule.
2: Estimate $\boldsymbol{\theta}_t$ via ridge regression on $\mathcal{H}_t$.
3: Compute one-step prediction $\hat{p}_{t+1|t}$ using Equation (5).
4: Solve scalar QP Equation (7) for next skip loss weight w_{t+1}^*.
5: Apply $w_{t+1} \leftarrow \text{clamp}(w_{t+1}^*, w_{\min}, w_{\max})$.
6: Update history $\mathcal{H}_{t+1}$ with (w_{t+1}, p_{t+1}).
7: **return** w_{t+1}

System Modeling and Online Identification The relationship between w_t and the observed skip probability p_t (the actual observed token fraction that is routed to skip expert) is time-varying and depends on the evolving router state. We capture this dynamics using an online-identified ARX model:

$$p_j = -\sum_{i=1}^{n_a} a_i \, p_{j-i} + \sum_{m=0}^{n_b-1} b_m \, w_{j-n_k-m} + e_j, \tag{5}$$

with parameter vector $\boldsymbol{\theta} = [a_1, \ldots, a_{n_a}, \, b_0, \ldots, b_{n_b-1}]^\top$. At each step t, we estimate $\boldsymbol{\theta}_t$ via ridge regression on a sliding window of M last steps:

$$\boldsymbol{\theta}_t = \arg\min_{\boldsymbol{\theta}} \sum_{j=t-M+1}^{t} \left(p_j - \boldsymbol{\phi}_j^\top \boldsymbol{\theta}\right)^2 + \lambda \|\boldsymbol{\theta}\|^2, \tag{6}$$

where $\boldsymbol{\phi}_j = [-p_{j-1}, \ldots, -p_{j-n_a}, \, w_{j-n_k}, \ldots, w_{j-n_k-n_b+1}]^\top$.

Predictive Model and QP Formulation After the model is estimated online, the one-step-ahead prediction of the skip probability for the next time step $\hat{p}_{t+1}$ by simply rolling out the fitted ARX model in Equation (5) To determine the next control action, we solve a quadratic optimization that balances tracking error and control smoothness:

$$\begin{aligned} \min_{w_{t+1}} \quad & Q\left(\hat{p}_{t+1} - p_{t+1}^{\text{tar}}\right)^2 + R_w \, w_{t+1}^2 + R_\Delta \left(w_{t+1} - w_t\right)^2, \\ \text{s.t.} \quad & w_{\min} \le w_{t+1} \le w_{\max}, \\ & |w_{t+1} - w_t| \le \Delta w_{\max}, \end{aligned} \tag{7}$$

where Q penalizes deviation from the target sparsity setpoint p_{t+1}^{tar}, which follows a customizable schedule. Notably, the target setpoint p_t^{tar} can follow a customized schedule (i.e., a warmup schedule from the natural skip probability $1/(1+N)$ to the user-defined target p^* to achieve a full training initially). The additional terms in objective, R_w discourages overly large weights, and R_Δ penalizes abrupt changes. We solve this optimization at each step and clip the output weight w_{t+1}^* with a min max range before apply to the system. The overall algorithm for the controller is sketched in Algorithm 1.

3.3 Expert Load Balancing

We adopt an auxiliary-free load balancing mechanism akin to that employed in DeepSeek models Wang et al. (2024), to ensure balanced expert utilization during training. This method introduces a mechanically updated expert-specific bias term $\boldsymbol{b}_{load}$ to the routing scores before the final argmax operation in the gating function Equation (2).

Normal Expert Bias A single shared bias vector b_{skip} across all K gating heads affords a consistent global perspective on load distribution. This vector is updated through a violation feedback loop, with no additional auxiliary loss terms in the objective function. The load violation error for a normal expert i is defined as the deviation from the average load

$$e_i = c_i - \bar{c}_n, \tag{8}$$

where c_i is the expert's assigned token count and $\bar{c}_n$ is the average load of the normal experts, i.e., total tokens per batch minus those routed to the skip expert.

Algorithm 2 Bias Update for Control-MoE routing in each layer

Input: Batch size B, bias update rate u, elasticity coef η
Initialize bias $\boldsymbol{b} \leftarrow 0$ for normal and skip experts
for each training batch $(\mathbf{x}_n, \mathbf{y}_n)_{n=1}^{B}$ **do**
    1. Compute routing g_k^i via Equation (2).
    2. Count expert loads c_i, skip load c_{skip} and total token load c_{total}.
    3. Compute load violation errors e_i and e_{skip} via Equation (8) and Equation (9).
    4. Update the bias vector for normal experts:
        $b_i \leftarrow b_i - u \cdot \text{sign}(e_i)$.
    5. Update the bias term for the skip expert:
        $b_{\text{skip}} \leftarrow b_{\text{skip}} - u \cdot \text{sign}(e_{\text{skip}})$.
    6. Apply elasticity on the skip bias:
        $b_{\text{skip}} \leftarrow b_{\text{skip}} - \eta \cdot u \cdot \text{sign}\,(b_{\text{skip}})$.
    7. Update the normal bias vector towards a zero mean:
        $b_i \leftarrow b_i - \frac{1}{n}\sum_{i=1}^{n} b_i$.
end for

Skip Expert Bias We impose an upper limit τ_{max} on the skip expert's selection percentage per layer to prevent the model from over-relying on the computationally cheap skip expert. The core mechanism is a violation error calculated for each data batch. Let $\boldsymbol{c}_{\text{skip}}$ be the count of tokens routed to the skip expert in a layer. The violation is triggered only when $\boldsymbol{c}_{\text{skip}}$ exceeds the allowed threshold based on the total expert loads c_{total}:

$$e_{\text{skip}} = \max(c_{\text{skip}} - \tau_{\text{max}} \cdot c_{\text{total}}, 0). \tag{9}$$

If c_{skip} is below the threshold, the error is zero without applying correction. This creates a "soft cap" that only activates to curb excessive skipping. In our experiments, τ_{max} is 0.75.

Bias Elasticity Furthermore, we incorporate an elasticity term that actively retracts the skip-expert bias b_{skip} toward zero. This mechanism mitigates systematic bias and enhances training stability by preventing excessive drift in the skip bias. We employ the skip bias solely as a safeguard against excessive skipping, while primarily relying on the importance and weighting of the skip expert to regulate skip probabilities. Additionally, we apply a normalization step to the biases of the regular experts. At each update, we subtract the mean of the regular-expert bias vector from each of its entries. This normalization is essential because adding the skip expert induces fluctuations in the expected load per regular expert, $\bar{c}_n$, across batches. Without this step, the bias vector for regular experts could accumulate a non-zero mean over time, thereby distorting routing decisions and skip probabilities. Following Wang et al. (2024), we adopt the sign-based variant of the bias update to promote stable training. The complete update procedure, which integrates load-balance violation errors, elasticity for the skip expert, and normalization for the regular experts, is detailed in Algorithm 2. The bias update rate u is set as 0.001 and elasticity coefficient η is 0.25 in our experiments.

Z-loss We adapt the standard Z-loss function in the Control-MoE router. Z-loss function is commonly used to stabilize training by penalizing excessively large routing logits Zoph et al. (2022). As Control-MoE employs K independent gating heads, each producing a separate set of logits, we compute the mean of the logits across all gates. This averaged logit vector is then fed to the original Z-loss function. This straightforward adaptation ensures training stability without compromising the routing design, finalizing Control-MoE as a robust plug-and-play module for existing MoE systems. We also found that w_{skip} usually has the same scale as the Z-loss weight.

4 Experiments

We first describe the models, training recipe, and evaluation protocol in Section 4.1. Section 4.2 compares Control-MoE with Top-K routing at matched numbers of effective experts and studies test-time activation scaling. Section 4.3 compares Control-MoE with public MoE models of similar size, and Section 4.4 examines how experts are allocated across tokens.

Table 1: Benchmark accuracy (%) across model families. 384/64 denote the total number of experts. K4/K6 denote the number of activated effective experts.

Benchmark	2B 384 K4		2B 384 K6		2B 64			10B 384		7B 64	
	Top4	Control-MoE	Top6	Control-MoE	Top4	Control-MoE (K16)	Control-MoE (K24)	Top6	Control-MoE	Top4	Control-MoE
MMLU	36.78	37.28	37.46	37.27	37.52	38.96	38.75	43.09	45.06	44.05	46.30
ARC-Easy	71.77	73.14	73.04	73.70	72.52	75.54	75.60	78.87	80.80	80.62	82.93
ARC-Challenge	39.31	42.59	41.08	42.78	40.58	43.90	44.44	49.27	53.55	50.97	55.75
TriviaQA	30.45	32.95	15.96	33.63	33.66	34.97	36.24	46.75	51.35	49.88	52.86
PopQA	11.00	12.10	11.10	11.30	12.60	12.90	12.30	15.00	16.30	16.30	17.30
BBH	27.57	27.22	30.56	29.85	31.76	30.11	31.21	34.11	34.24	40.10	39.10
HellaSwag	52.02	53.76	53.02	55.38	55.36	57.73	58.17	64.89	66.99	66.91	69.65
PIQA	71.35	72.04	71.72	72.76	72.12	73.14	73.94	76.31	77.36	76.24	77.58
WinoGrande	59.92	62.42	60.25	62.49	65.75	65.83	67.06	70.41	72.06	75.32	76.26
DROP	26.17	26.31	30.08	29.11	37.46	27.04	32.97	46.42	45.21	51.32	50.72
HumanEval	9.43	8.29	13.24	8.14	11.17	18.16	11.28	19.23	21.95	20.66	21.97
MBPP	26.38	26.24	26.15	26.20	29.09	31.61	24.89	40.58	41.46	40.33	45.13
GSM8K	31.92	31.69	33.59	29.11	35.25	42.00	42.76	54.36	59.59	57.16	63.61
Average	**38.01**	**38.92**	**38.25**	**39.36**	**41.14**	**42.45**	**42.28**	**49.18**	**51.22**	**51.53**	**53.78**
Effective experts	**4.00**	**4.00**	**6.00**	**6.01**	**4.00**	**4.00**	**3.97**	**6.00**	**6.02**	**4.00**	**6.09**
Activated params (B)	**0.207**	**0.201**	**0.220**	**0.213**	**0.480**	**0.480**	**0.480**	**0.561**	**0.536**	**1.296**	**1.507**

4.1 Experimental Setup

Models We train OLMoE-derived Control-MoE models at 2B, 7B, and 10B scales Muennighoff et al. (2024). The 7B model retains the 64-expert OLMoE layout, while the 2B and 10B models use 384 fine-grained experts and one shared expert to study higher-sparsity regimes; we additionally evaluate 2B, 64-expert variants. Each configuration is paired with a Top-K baseline using the same backbone, data, and training schedule, with K matched to the target number of effective experts. We define effective experts as the average number of gate selections per token that reach normal experts. Architecture choices, sparsity targets, and parameter counts are reported in Section B.1.

Training Data All configurations are trained once with the same random seed on a 600B-token public corpus. Training uses a 500B-token pretraining mixture followed by 100B tokens of higher-quality mid-training data, together with the OLMoE optimization recipe and a terminal learning-rate decay. The corpus composition and complete optimization settings are provided in Section B.2.

Controller Settings Each MoE layer uses an independent MPC controller that adjusts the skip-loss weight online to track its target skip probability. The same controller configuration is used across model scales and sparsity targets without retuning. Online identification, safety mechanisms, and controller hyperparameters are detailed in Section B.4.

Evaluation We evaluate all matched models in OLMES Gu et al. (2025) on 13 benchmarks spanning knowledge, reasoning, commonsense, reading comprehension, code, and mathematics, and report their unweighted mean. The comparison with public MoE models instead uses the benchmarks reported by those models. Task-specific shot counts and metrics are listed in Section B.3.

4.2 Main Results

Extreme Sparse MoE Table 1 shows that Control-MoE improves the accuracy–activation trade-off across model families. At matched average expert counts, it raises average accuracy from 38.01% to 38.92% for 2B-384-K4, from 38.25% to 39.36% for 2B-384-K6, and from 41.14% to 42.45% and 42.28% for the two 2B-64 configurations. The 384-expert models also use fewer activated parameters than their Top-K baselines: 0.201B versus 0.207B at 2B scale and 0.536B versus 0.561B at 10B scale. The displayed 7B operating point uses more experts than Top-4 (6.09 versus 4.00), so it demonstrates test-time accuracy scaling rather than a compute-matched comparison; at a matched 3.98 experts, the same checkpoint still improves the average from 51.53% to 53.44% (Table 11). These results support the central claim that token-dependent allocation uses a given expert budget more effectively than fixed Top-K routing.

Testing-Time Activation Scaling As discussed in Section 3.3, the skip-expert bias provides direct control over the number of normal experts activated at inference time. Increasing this bias routes more gates to the skip expert and reduces computation, while decreasing it enables greater expert

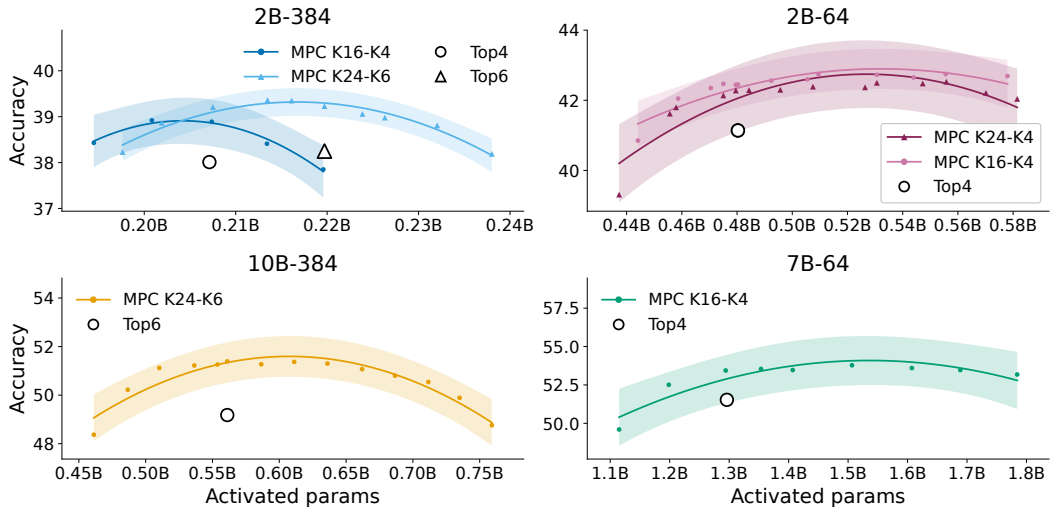

Figure 2: Activation scaling of average accuracy. Each subplot corresponds to one model family, and each curve evaluates the same Control-MoE checkpoint at different activation levels. 384/64 denote the total number of experts; K16-K4 denotes 16 gates and a training target of four effective experts.

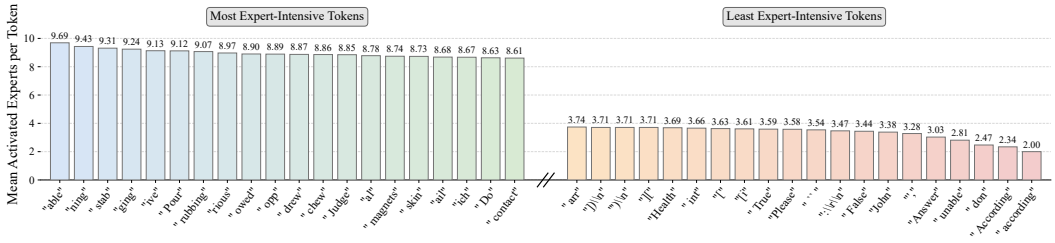

Figure 3: Harder tokens need more experts. Average number of activated experts per token for Control-MoE 10B-384 on all evaluation benchmarks

utilization. This property turns a single trained Control-MoE checkpoint into an entire family of deployment variants, offering the capability of **testing-time activation scaling**: latency-oriented "Flash" and performance-oriented "Pro" models can be instantiated after training simply by selecting different activation points, eliminating the need to train and maintain separate checkpoints. Figure 2 shows that these operating points form a controllable accuracy: compute frontier across all evaluated model scales and expert configurations. More importantly, Control-MoE strictly Pareto-dominates every fixed Top-K reference shown: for each baseline, it achieves higher accuracy while activating fewer parameters. Although performance eventually saturates at excessive activation levels, the deployment budget can be chosen post hoc to maximize accuracy under a given latency or compute constraint. Full task-level sweeps in Section C confirm that this advantage extends beyond the aggregate metric.

4.3 Ablations

Comparison to Similar Sized MoE Model Existing large MoE models are typically trained on massive token budgets, for instance, OpenMoE-8B/32E (1.1T tokens) and OLMoE (5T tokens). To ensure a rigorous and fair comparison, we extended the training of one OLMoE-Control-MoE model to 600B tokens, a point at which its performance on all benchmarks had stabilized. In terms of performance, as shown in Table 2, Control-MoE delivers competitive results with other models that activate 2–3 times more parameters per token. A crucial finding is that it outperforms its direct architectural sibling under the same training token budget, the standard OLMoE employing a static

Table 2: Comparison of Control-MoE @ 600B Performance v.s. Similar-sized MoE Models

Models	LLaMA-MoE	OpenMoE	MoE++	OLMoE	Control-MoE
	3.0B	8B/32E		(Top-4)	Top16-75%
Params	3.0B/7B	2.1B/8B	<1.2B/7B	1.3B/7B	1.3B/7B
SciQ	89.9	92.3	89.7	92.4	**92.6**
PIQA	77.5	74.2	77.6	**78.4**	77.3
WinoGrande	63.6	60.3	63.1	**67.1**	66.5
ARC-E	66.8	64.1	66.5	74.2	**75.2**
ARC-C	40.9	30.3	42.3	43	**44.8**
HellaSwag	70.8	45.5	72.3	70.2	**72**
BoolQ	**71.9**	61.2	68.3	68.3	71.4
MMLU	26.8	27.2	24.6	28.1	**28.3**
Average	42.35	37.92	42.03	43.48	**44.00**

Top-K router with $K = 4$, which demonstrates the advantage of our dynamic Control-MoE routing mechanism. Furthermore, when compared to another prominent dynamic activation model, MoE++ Jin et al. (2024) (trained on 1T tokens), our model achieves superior performance on a majority of benchmarks while maintaining a significantly lower activation rate.

4.4 Analysis

Challenging Tokens Need More Experts To validate the Control-MoE router's capacity for dynamic, difficulty-aware computation, we analyzed the relationship between expert activation counts and linguistic token categories, as shown in Section 4.2. The results reveal two main patterns. First, expert activation varies substantially across tokens: the most demanding tokens activate up to 9.5 experts on average, whereas the least demanding activate only about 2.0. Second, this variation aligns with semantic richness and contextual flexibility. Intuitively, a token that is semantically loaded, ambiguous, morphologically productive, or syntactically variable, such as a stem like `contact`, a modifier like `able`, or a productive suffix such as `ing` can participate in many meanings and continuations. The router therefore benefits from recruiting a broader set of experts to cover these possibilities, leading to higher average expert counts. By contrast, highly predictable structural tokens such as `Answer` or punctuation marks play a largely fixed role determined by syntax or templates, and thus can be handled by fewer experts.

Inference System Implications From a roofline perspective, Control-MoE replaces the fixed expert-compute budget of Top-K routing with token-dependent activation. During prefill, large expert matrix multiplications often approach the compute-bound regime; skipping experts for simpler tokens reduces expert FLOPs and can lower time to first token when routing and dispatch overheads are amortized. During low-concurrency decoding, inference is typically constrained by the memory bandwidth required to load expert weights, so activating fewer experts can reduce weight traffic and token latency. At high concurrency with large-scale expert parallelism DeepSeek-AI (2024), the workload can shift toward a compute-bound regime; lower activation then reduces expert FLOPs and can also decrease all-to-all token traffic, improving potential throughput. These benefits become more important as sparse Fu et al. (2025) and linear Kimi-Team (2025) attention reduce attention cost, or as speculative decoding Leviathan et al. (2023) increases the number of candidate tokens evaluated per step.

5 Conclusion

We introduced Control-MoE, a dynamic MoE router that combines parallel Top-1 gates, a skip expert, and MPC-based sparsity control. Across 2B–10B models, it improves average accuracy by 0.91–2.04 percentage points over matched Top-K baselines while allowing one checkpoint to serve multiple inference budgets. Token-level analysis confirms that it allocates more experts to semantically variable tokens than to predictable structural tokens. These results show that controllable, token-dependent routing can use expert compute more effectively than fixed Top-K activation. Future work should validate this benefit at larger scales and measure end-to-end hardware efficiency.

AI USE STATEMENT

We used generative AI tools to assist with writing and language polishing, literature retrieval and discovery, research ideation and execution, and drafting sections of the paper. We did not use generative AI tools to generate synthetic datasets or prove mathematical claims. We reviewed all AI-assisted work and take responsibility for the final content of this paper, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Maryam Akhavan Aghdam, Hongpeng Jin, and Yanzhao Wu. Da-moe: Towards dynamic expert allocation for mixture-of-experts models, 2024. URL https://arxiv.org/abs/2409.06669.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019. URL https://arxiv.org/abs/1911.11641.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Yuxuan Chen and Zhiwei Zhang. Tdmoe: Tail-probability-based dynamic-k moe. In *2026 Joint International Conference on Automation-Intelligence-Safety (ICAIS) & International Symposium on Autonomous Systems (ISAS)*, pp. 1–7, 2026. doi: 10.1109/ICAISISAS68969.2026.11567792.

Xin Cheng, Rui Tian, Wangding Zeng, Damai Dai, Qinyu Chen, Bingxuan Wang, Zhenda Xie, Kezhao Huang, Xingkai Yu, Chengqi Deng, Shangyan Zhou, Chenggang Zhao, Zhewen Hao, Yukun Li, Han Zhang, Zhengyan Zhang, Yixuan Wei, M. Y. Xu, Huishuai Zhang, Dongyan Zhao, and Wenfeng Liang. Conditional memory via scalable lookup: A new axis of sparsity for large language models, 2026. URL https://arxiv.org/abs/2601.07372.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting

Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2368–2378, 2019. URL https://aclanthology.org/N19-1246/.

Zichuan Fu, Wentao Song, Yejing Wang, Xian Wu, Yefeng Zheng, Yingying Zhang, Derong Xu, Xuetao Wei, Tong Xu, and Xiangyu Zhao. Sliding window attention training for efficient large language models, 2025. URL https://arxiv.org/abs/2502.18845.

Carlos E. García, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989a. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(89)90002-2. URL https://www.sciencedirect.com/science/article/pii/0005109889900022.

Carlos E. García, David M. Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989b. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(89)90002-2. URL https://www.sciencedirect.com/science/article/pii/0005109889900022.

GLM Team . Glm-5: From vibe coding to agentic engineering. *arXiv preprint*, arXiv:2602.15763, 2026. URL https://arxiv.org/abs/2602.15763.

Adem Golcuk. Design and implementation of a hybrid flc+pid controller for pressure control of sleep devices. *Biomedical Signal Processing and Control*, 76:103702, 2022. ISSN 1746-8094. doi: https://doi.org/10.1016/j.bspc.2022.103702. URL https://www.sciencedirect.com/science/article/pii/S1746809422002245.

Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. OLMES: A standard for language model evaluations. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025. URL https://arxiv.org/abs/2406.08446.

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models, 2025. URL https://arxiv.org/abs/2405.14297.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. Harder tasks need more experts: Dynamic routing in moe models, 2024. URL https://arxiv.org/abs/2403.07652.

Can Jin, Hongwu Peng, Mingcan Xiang, Qixin Zhang, Xiangchi Yuan, Amit Hasan, Ohiremen Dibua, Yifan Gong, Yan Kang, and Dimitris N. Metaxas. Sparsity-controllable dynamic top-p moe for large foundation model pre-training, 2025. URL https://arxiv.org/abs/2512.13996.

Peng Jin, Bo Zhu, Li Yuan, and Shuicheng Yan. Moe++: Accelerating mixture-of-experts methods with zero-computation experts, 2024. URL https://arxiv.org/abs/2410.07348.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, 2017. URL https://aclanthology.org/P17-1147/.

Kimi-Team. Kimi linear: An expressive, efficient attention architecture, 2025. URL https://arxiv.org/abs/2510.26692.

Xunhao Lai, Weiqi Xu, Yufeng Yang, Qiaorui Chen, Yang Xu, Lunbin Zeng, Xiaolong Li, Haohai Sun, Haichao Zhu, Vito Zhang, Jinkai Hu, Jiayao Li, Rui Gao, Zekun Li, Songquan Zhu, Jingkai Zhou, and Pengyu Zhao. Minimax sparse attention, 2026. URL https://arxiv.org/abs/2606.13392.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding, 2023. URL https://arxiv.org/abs/2211.17192.

Jia liang Song, Wen long Cheng, Zhi ming Xu, Shuai Yuan, and Ming hou Liu. Study on pid temperature control performance of a novel ptc material with room temperature curie point. *International Journal of Heat and Mass Transfer*, 95:1038–1046, 2016. ISSN 0017-9310. doi: https://doi.org/10.1016/j.ijheatmasstransfer.2015.12.057. URL https://www.sciencedirect.com/science/article/pii/S001793101530346X.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023. URL https://arxiv.org/abs/2212.10511.

D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. ISSN 0005-1098. doi: https://doi.org/10.1016/S0005-1098(99)00214-9. URL https://www.sciencedirect.com/science/article/pii/S0005109899002149.

Meta. Llama 4: Maverick. https://ai.meta.com/blog/llama-4/, 2025. Released April 5, 2025.

MiniMax AI. The minimax-m2 series: Mini activations unleashing max real-world intelligence. *arXiv preprint*, arXiv:2605.26494, 2026. URL https://arxiv.org/abs/2605.26494.

N. Minorsky. Directional stability of automatically steered bodies. *Journal of the American Society for Naval Engineers*, 34(2):280–309, 1922. doi: https://doi.org/10.1111/j.1559-3584.1922.tb04958.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1559-3584.1922.tb04958.x.

Moonshot AI. Kimi k2.6. https://www.kimi.com/ai-models/kimi-k2-6, 2026. Accessed May 2026.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. Olmoe: Open mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2409.02060.

S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

Qwen Team. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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, 2024. URL https://arxiv.org/abs/2404.02258.

J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428, 1978. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(78)90001-8. URL https://www.sciencedirect.com/science/article/pii/0005109878900018.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019. URL https://arxiv.org/abs/1907.10641.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://arxiv.org/abs/2402.00159.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022. URL https://arxiv.org/abs/2210.09261.

Zhou Tao, YongXiang Hua, Chaohu Liu, Shida Wang, and Linli Xu. DARE:difficulty-aware dynamic routing for mixture of experts, 2026. URL https://openreview.net/forum?id=S7o8zBYw4V.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, M. C., Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Guanduo Chen, Guangyu Chen, Guanzheng Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kexin Chen, Peng Chen, Ruijue Chen, Wentao Chen, Xin Chen, Yang Chen, Yanru Chen, Yifei Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Dazhi Cheng, Yean Cheng, Jialei Cui, Jingbing Cui, Anqi Dai, Jiaqi Deng, Hao Ding, Rui Ding, Shaofeng Ding, Mengfan Dong, Mengnan Dong, Yuhao Dong, Yuxin Dong, Angang Du, Chenzhuang Du, Dikang Du, Jusen Du, Yulun Du, Yu Fan, Jing Feng, Qiulin Feng, Yichen Feng, Kelin Fu, Qiang Fu, Fuxuan Gao, Hongcheng Gao, Jingyue Gao, Tong Gao, Weijia Gao, Shangyi Geng, Jie Gong, Linhu Gong, Shengao Gong, Xiaochen Gong, Qizheng Gu, Yicheng Gu, Shuhao Guan, Haiqing Guo, Shiqi Guo, Xiang Guo, Zhengyan Guo, Beixi Hao, Wenxin Hao, Xiaoru Hao, Dailan He, Haotian He, Lehan He, Qi He, Weiran He, Xinran He, Xinyi He, Yibo He, Yunjia He, Chao Hong, Tiange Hong, Hao Hu, Jiaxi Hu, Ruikun Hu, Weiming Hu, Yangyang Hu, Zhenxing Hu, Liang Hua, Jinbin Huang, Ke Huang, Ruiyuan Huang, Siying Huang, Weixiao Huang, Yan Huang, Zhengjie Huang, Zhiqi Huang, Yulong Hui, Chaobo Jia, Yutong Jiang, Zhejun Jiang, Zuoyou Jiang, Wenyi Jin, Xinyi Jin, Yu Jing, Huanjun Kong, Guokun Lai, Aidi Li, Cheng Li, Chengyuan Li, Cong Li, Fang Li, Guanyu Li, Haoyang Li, Jia Li, Junxiong Li, Lei Li, Letian Li, Lincan Li, Weihong Li, Wentao Li, Xintong Li, Yang Li, Yishen Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zhaoxi Li, Zheming Li, Zhengxiao Li, Zhiyuan Li, Jiawei Lin, Xiaohan Lin, Yibo Lin, Zichao Lin, Ziyan Lin, Bill Liu, Boxiao Liu, Chuan Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yipeng Liu, Zhengying Liu, Zhiheng Liu, Enzhe Lu, Haoyu Lu, Linqiang Lu, Tingzhan Lu, Zhiyuan Lu, Aotian Luo, G. Luo, Junyu Luo, Yifan Luo, B. Lyu, Wenzhou Lyu, Shaoguang Mao, Yuan Mei, Xin Men, Minqing Ni, Yixuan Niu, Siyuan Pan, Shujun Peng, Zhangyang Qi, Ruoyu Qin, ZeChao Qin, Zeyu Qin, Haiquan Qiu, Jianxin Qiu, Jiezhong Qiu, Bowen Qu, Yuhao Qu, Zeyu Shang, Youbo Shao, Han Shen, Jincheng Shi, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Wingchun Siu, Pengwei Song, Xiaoxi Song, Jianlin Su, Yunfeng Su, Zhaochen Su, Lin Sui, Jingsong Sun, Junyao Sun, Shaoning Sun, Shuzhe Sun, Tongyu Sun, Yujun Sun, Yunpeng Tai, Chuning Tang, Heyi Tang, Sirui Tang, Zecheng Tang, Chaoran Tian, Rongpeng Tian, Yu Tian, Wei Tu, Chensi Wang, Chuang Wang, Chunjie Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Hao Wang, Hao Wang, Huaqing Wang, Hui Wang, Jiayi Wang, Jinglong Wang, Jinhong Wang, Jiuzheng Wang, Linian Wang, Shaobo Wang, Shenzhi Wang, Shuyi Wang, Si Wang, Siyuan Wang, Tianfu Wang, Wenjue Wang, Xingran Wang, Xinmei Wang, Xinyuan Wang, Xusheng Wang, Yalin Wang, Yangkun Wang, Yao Wang, Yaoyu Wang, Yejie Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhenhao Wang, Zhongsheng Wang, Zifan Wang, Chu Wei, Ming Wei, Shouxin Wei, Zichen Wen, Fan Wu, Haoning Wu, Rucong Wu, Wenhao Wu, Xiaoxue Wu, Yingcong Wu, Yongqi Wu, Yuxin Wu, Zijian Wu, Xinglang Xian, Chenxuan Xiang, Yuye Xiang, Bocheng Xiao, Chenjun Xiao, Xin Xiao, Jin Xie, Xiaotong Xie, Yifeng Xie, Zhe Xie, Bowei Xing,

Yiming Xiong, Baosheng Xu, Boyu Xu, Jiale Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Qingtao Xu, Shuyao Xu, Suting Xu, Tiantian Xu, Tianxiang Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ye Xu, Yueni Xu, Ziyao Xu, Haonan Xue, Junjie Yan, Yaoyao Yan, Fan Yang, Guangyao Yang, Hao Yang, Junwei Yang, Ruoyu Yang, Wenjie Yang, Xiaofei Yang, Xinyu Yang, Yi Yang, Yiling Yang, Ying Yang, Yuchen Yang, Zhen Yang, Zhilin Yang, Zian Yang, Zuhao Yang, Haotian Yao, Dan Ye, Haoran Ye, Wenjie Ye, Zhanbo Ye, Bohong Yin, Haoxiang Yin, Xietong Yin, Chengzhen Yu, Haozhen Yu, Longhui Yu, Shengnan Yu, Shuying Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Tongtian Yue, Wei Yue, Yang Yue, Dunyuan Zha, Haobing Zhan, B. H. Zhang, Dehao Zhang, Fei Zhang, Hao Zhang, Haoyuan Zhang, Huanyu Zhang, Jiapei Zhang, Jiaxuan Zhang, Jin Zhang, Kaiyi Zhang, Miaozhen Zhang, Puqi Zhang, Qinglei Zhang, Rong Zhang, Rui Zhang, Shaoshuai Zhang, Shiyi Zhang, Xiaobin Zhang, Xiaoyun Zhang, Y. Zhang, Yangkun Zhang, Ye Zhang, Yichi Zhang, Yikun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Zijing Zhang, Bin Zhao, Chenguang Zhao, Feifan Zhao, Jinglun Zhao, Jinxiang Zhao, Shuai Zhao, Wenshuo Zhao, Xiangyu Zhao, Xuanle Zhao, Yikai Zhao, Zijia Zhao, Haozhi Zheng, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Haofeng Zhong, Lei Zhong, Longguang Zhong, M. Zhou, Qiankang Zhou, Runjie Zhou, Ruozhang Zhou, Xinyu Zhou, Yiqiao Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yangjunfeng Zhu, Yuxuan Zhu, Zhen Zhu, Chen Zhuang, Weiyu Zhuang, and Xinxing Zu. Kimi k3: Open frontier intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Meituan LongCat Team, Anchun Gui, Bei Li, Bingyang Tao, Bole Zhou, Borun Chen, Chao Zhang, Chao Zhang, Chengcheng Han, Chenhui Yang, Chi Zhang, Chong Peng, Chuyu Zhang, Cong Chen, Fengcun Li, Gang Xu, Guoyuan Lin, Hao Jiang, Hao Liang, Haomin Fu, Haoxiang Ma, Hong Liu, Hongyan Hao, Hongyin Tang, Hongyu Zang, Hongzhi Ni, Hui Su, Jiahao Liu, Jiahuan Li, Jialin Liu, Jianfei Zhang, Jianhao Xu, Jianing Wang, Jiaqi Sun, Jiaqi Zhang, Jiarong Shi, Jiawei Yang, Jingang Wang, Jinrui Ding, Jun Kuang, Jun Xu, Ke He, Kefeng Zhang, Keheng Wang, Keqing He, Li Wei, Liang Shi, Lin Qiu, Lingbin Kong, Lingchuan Liu, Linsen Guo, Longfei An, Mai Xia, Meng Zhou, Mengshen Zhu, Peng Pei, Pengcheng Jia, Qi Gu, Qi Guo, Qiong Huang, Quan Chen, Quanchi Weng, Rongxiang Weng, Ruichen Shao, Rumei Li, Shanglin Lei, Shuai Du, Shuaikang Liu, Shuang Zhou, Shuhao Hu, Siyu Xu, Songshan Gong, Tao Liang, Tianhao Hu, Wei He, Wei Shi, Wei Wang, Wei Wu, Wei Zhuo, Weifeng Tang, Wenjie Shi, Wenlong Zhu, Xi Su, Xiangcheng Liu, Xiangyu Xi, Xiangzhou Huang, Xiao Liu, Xiaochen Jiang, Xiaowei Shi, Xiaowen Shi, Xiaoyu Li, Xin Chen, Xinyue Zhao, Xuan Huang, Xuemiao Zhang, Xuezhi Cao, Xunliang Cai, Yajie Zhang, Yang Chen, Yang Liu, Yang Liu, Yang Zheng, Yaoming Wang, Yaqi Huo, Yerui Sun, Yifan Lu, Yiyang Li, Youshao Xiao, Yuanzhe Lei, Yuchen Xie, Yueqing Sun, Yufei Zhang, Yuhuai Wei, Yulei Qian, Yunke Zhao, Yuqing Ding, Yuwei Jiang, Zhaohua Yang, Zhengyu Chen, Zhijian Liu, Zhikang Xia, Zhongda Su, Ziran Li, Ziwen Wang, Ziyuan Zhuang, Zongyu Wang, and Zunyuan Yang. Introducing longcat-flash-thinking: A technical report, 2025. URL https://arxiv.org/abs/2509.18883.

Team Olmo. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts, 2024. URL https://arxiv.org/abs/2408.15664.

Yudong Wang, Zixuan Fu, Jie Cai, Peijun Tang, Hongya Lyu, Yewei Fang, Zhi Zheng, Jie Zhou, Guoyang Zeng, Chaojun Xiao, Xu Han, and Zhiyuan Liu. Ultra-fineweb: Efficient data filtering and verification for high-quality LLM training data, 2025. URL https://arxiv.org/abs/2505.05427.

Robert Washbourne, Rishi Iyer, Tomas Figliolia, Henry Zheng, Ryan Lorig-Roach, Sungyeon Yang, Pritish Yuvraj, Quentin Anthony, Yury Tokpanov, Xiao Yang, Ganesh Nanduru, Stephen Ebert, Praneeth Medepalli, Skyler Szot, Srivatsan Rajagopal, Alex Ong, Bhavana Mehta, and Beren Millidge. Zaya1-8b technical report, 2026. URL https://arxiv.org/abs/2605.05365.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. RedPajama: an open dataset for training large language models. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2411.12372.

Juntong Wu, Jialiang Cheng, Qishen Yin, Yue Dai, Yuliang Yan, Fuyu Lv, Ou Dan, and Li Yuan. Beam: Binary expert activation masking for dynamic routing in moe, 2026. URL https://arxiv.org/abs/2605.14438.

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, 2025. URL https://arxiv.org/abs/2505.09388.

Kin Yeung and Jie Huang. Development of a remote-access laboratory: a dc motor control experiment. *Computers in Industry*, 52(3):305–311, 2003. ISSN 0166-3615. doi: https://doi.org/10.1016/S0166-3615(03)00133-7. URL https://www.sciencedirect.com/science/article/pii/S0166361503001337. Soft Computing in Industrial Applications.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019. URL https://arxiv.org/abs/1905.07830.

Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. Adamoe: Token-adaptive routing with null experts for mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2406.13233.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models, 2022. URL https://arxiv.org/abs/2202.08906.

K.J. Åström and T. Hägglund. The future of pid control. *Control Engineering Practice*, 9(11):1163–1175, 2001. ISSN 0967-0661. doi: https://doi.org/10.1016/S0967-0661(01)00062-4. URL https://www.sciencedirect.com/science/article/pii/S0967066101000624. PID Control.

A Relative Work: Sparse MoE Activation Rates

Recent frontier MoE models increase total capacity while activating only a small fraction of their parameters for each token. Table 3 summarizes this trend and places the OLMoE backbone used in our experiments in context. Most recent large-scale systems report activation rates near 3–5%, compared with approximately 19% for OLMoE. These figures are not direct efficiency comparisons because architectures and parameter-counting conventions differ, but they motivate evaluating Control-MoE in the fine-grained, high-sparsity regime represented by our 384-expert models. Unlike the systems listed below, our focus is on controlling the number of active experts per token rather than fixing it at inference time.

Table 3: Activation rates of representative sparse Mixture-of-Experts models.

Model	**Release Date**	**Total Params**	**Active Params**	**Activation Rate**
DeepSeek-V4 / V4-Pro (DeepSeek-AI et al., 2026)	2026-04	1.6T	49B	3.1%
Qwen3.5-397B-A17B (Qwen Team, 2026)	2026-02	397B	17B	4.3%
Kimi K2.6 (Moonshot AI, 2026)	2026-04	1T	32B	3.2%
Kimi K3 (Team et al., 2026)	2026-07	2.8T	104B	3.7%
Llama 4 Maverick (Meta, 2025)	2025-04	$\sim$400B	17B	$\sim$4.3%
MiniMax M2 / M2.7 (MiniMax AI, 2026)	2025 / 2026	229.9B	9.8B	4.3%
MiniMax M3 (Lai et al., 2026)	2026-06	428B	$\sim$23B	$\sim$5.4%
GLM-5 / GLM-5.2 (GLM Team , 2026)	2026-02	744B	40B	5.4%
OLMoE (Muennighoff et al., 2024)	2024	7B	1.3B	$\sim$19%

B EXPERIMENTAL DETAILS

This section provides the architecture, training, evaluation, and controller settings needed to reproduce the experiments. The paired Control-MoE and Top-K models use the same backbone, data, optimization schedule, and evaluation harness; their routing mechanisms and associated load-balancing procedures are the principal differences.

B.1 MODEL AND TRAINING CONFIGURATIONS

Table 4 covers three scales and two expert layouts. The 7B configuration retains the 64-expert OLMoE design and uses $K = 16$ with $p^* = 0.75$ to target four effective experts. The default 2B and 10B configurations use 384 smaller experts and $K = 24$ with $p^* = 0.75$ to target six; the four-expert 2B run keeps $K = 24$ and sets $p^* = 5/6$. Additional 2B, 64-expert runs use 16 or 24 gates with a target of four effective experts. For each configuration, the Top-K baseline matches the target expert count. When several gates select the same normal expert, its output is computed only once. Because the parallel gates add router parameters, we report activated parameters both with and without this routing overhead; the latter isolates the expert computation controlled by the sparsity target.

Table 4: Model and training configurations. Parameter counts include the input embedding and the untied output head, and activated parameters are counted at the default target. The expert FFN size is the SwiGLU intermediate dimension, which is half of the `mlp_hidden_size` in the OLMo configuration.

	2B	7B	10B
Architecture			
MoE layers	16	16	16
d_{model}	512	2,048	1,024
Attention heads (MHA)	8	16	8
Expert FFN size	256	1,024	512
Normal experts N	384	64	384
Shared experts	1	0	1
Control-MoE gates K	24	16	24
Sparsity target p^*	0.75	0.75	0.75
Target effective experts	6	4	6
Top-K baseline K	6	4	6
RoPE θ	10,000	10,000	10,000
Context length	4,096	4,096	4,096
Vocabulary size	152,064	152,064	152,064
Parameters			
Total, Control-MoE/ Top-K	2.67B / 2.60B	7.37B / 7.34B	10.22B / 10.07B
Activated, Control-MoE/ Top-K	0.29B / 0.22B	1.33B / 1.30B	0.71B / 0.56B
Activated excluding router	0.22B	1.29B	0.56B
Training			
Tokens	600B (500B stage 1 + 100B stage 2)		
Batch size	1,024 sequences		
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$		
Peak learning rate	4×10^{-4}		
Learning-rate schedule	10B-token warmup, OLMoE cosine over 5T tokens, cosine decay to 4×10^{-5} over the last 50B tokens		
Weight decay	0.1		
Gradient clipping	1.0		
Precision	bfloat16 mixed precision		
Load balancing, Control-MoE	auxiliary-free bias update Wang et al. (2024)		
Load balancing, Top-K	auxiliary loss, weight 0.01		
Router z-loss weight	10^{-3}		

B.2 Training Data and Optimization

All models are trained once with the same random seed on a 600B-token public corpus, approximately one-eighth of the 5T-token budget used by OLMoE. We retokenize the data with the Qwen2 tokenizer Qwen Team (2024) and use a vocabulary and embedding size of $152{,}064$. The first 500B tokens follow the source composition of the Dolma 3 pretraining pool Team Olmo (2025), replacing its Common Crawl component with English and Chinese pages from Ultra-FineWeb Wang et al. (2025). The remaining sources are English Wikipedia from Dolma 1.7 Soldaini et al. (2024), arXiv papers from RedPajama Weber et al. (2024), FineMath-3+ and Stack-Edu code in Python, Markdown, and C Allal et al. (2025), and scientific PDFs from olmOCR Team Olmo (2025). The final 100B tokens use the Dolma 3 Dolmino mid-training mixture Team Olmo (2025), which emphasizes high-quality web and PDF text, code, mathematics, question answering, instruction following, and reasoning.

The optimization settings are summarized in Table 4. All models use AdamW Loshchilov & Hutter (2019), a global batch of 1,024 sequences, gradient clipping at norm 1.0, and bfloat16 mixed precision. The learning rate warms up over 10B tokens and initially follows the OLMoE cosine schedule. Because that schedule spans 5T tokens and would remain within 3% of its peak over our shorter training budget, we replace it during the final 50B tokens with a cosine decay from 4×10^{-4} to 4×10^{-5}. Control-MoE uses the auxiliary-free bias update in Section 3.3, whereas the Top-K baseline uses an auxiliary load-balancing loss with weight 0.01. Both use a router z-loss Zoph et al. (2022) with weight 10^{-3}.

B.3 Evaluation Protocol

The evaluation suite spans knowledge, reasoning, commonsense, reading comprehension, code, and mathematics. Table 5 records the shot count and metric for each task so that the aggregate results can be reproduced. All matched Control-MoE and Top-K comparisons use these same settings in OLMES, and the reported overall score is the unweighted mean across the 13 tasks. The separate comparison with public MoE models uses the task results reported by those models and should therefore be interpreted as contextual rather than strictly controlled.

Table 5: Evaluation settings. All tasks are run in OLMES Gu et al. (2025). We adopt the shots and metrics of Cheng et al. (2026) for all tasks except MBPP, which we evaluate zero-shot instead of 3-shot.

Category	Benchmark	Shots	Metric
Knowledge	MMLU Hendrycks et al. (2021)	5	Acc.
	TriviaQA Joshi et al. (2017)	5	EM
	PopQA Mallen et al. (2023)	15	EM
Reasoning	ARC-Easy Clark et al. (2018)	25	Acc.
	ARC-Challenge Clark et al. (2018)	25	Acc.
	BBH Suzgun et al. (2022)	3	EM
Commonsense	HellaSwag Zellers et al. (2019)	0	Acc.
	PIQA Bisk et al. (2019)	0	Acc.
	WinoGrande Sakaguchi et al. (2019)	5	Acc.
Reading comprehension	DROP Dua et al. (2019)	1	F1
Code	HumanEval Chen et al. (2021)	0	Pass@1
	MBPP Austin et al. (2021)	0	Pass@1
Math	GSM8K Cobbe et al. (2021)	8	EM

B.4 MPC Controller Configuration

The controller settings in Table 6 are shared across all model scales, layers, and sparsity targets. Each MoE layer runs an independent controller, while routing statistics and control history are stored on the CPU. At every control step, ridge regression fits a low-order ARX model to recent skip probabilities and skip-loss weights, with input gains constrained to be nonnegative. The quadratic objective balances skip-rate tracking against the magnitude and rate of change of the skip-loss weight. Input bounds, a deadband, and target warmup prevent abrupt routing changes, while a short startup

probe provides sufficient variation for online identification. Using one configuration throughout tests whether the controller transfers across operating regimes without scale-specific retuning.

Table 6: Shared ARX–MPC controller and safety parameters.

Parameter	**Value**	**Meaning**
ARX orders: `na`, `nb`, `nk`	1, 2, 1	ARX orders: one past output and two input terms. $n_k = 1$ makes the next skip-loss weight the decision variable.
window size M	128	Rolling measurement/control history used to identify the ARX model.
Ridge Weight λ	10.0	Ridge regularization coefficient for ARX fitting.
QP Weight Q	1.0	Penalty on predicted skip-rate tracking error.
QP Weight R_w	0.1	Penalty on the scale of skip loss weight
QP Weight R_Δ	200.0	Penalty on the change skip loss weight between 2 consecutive steps
w_{min} / w_{max}	0.0 / 0.1	Hard lower and upper bounds on the skip loss actuator.
$\Delta w_{\max}$	2×10^{-4}	Maximum change of the skip-loss weight per step.
Deadband	0.01	Tracking errors below this value are treated as zero.
Target warmup	15,000 steps	Steps over which the target rises from $1/(N+1)$ to p^*.
Startup probe	2×10^{-5}, 2,000 steps	Fixed excitation of the skip-loss weight before closed-loop control.

C PER-BENCHMARK ACTIVATION SCALING

The following tables expand the aggregate curves in Figure 2 into task-level results. The first column in each table is the matched fixed Top-K baseline; the remaining columns evaluate a single trained Control-MoE checkpoint at different skip-expert biases. Because routing is token dependent, A reports the measured average number of normal experts selected per token rather than a nominal setting.

2B models with 384 experts The four-expert checkpoint reaches its highest average at $A = 4.00$ and remains essentially unchanged at $A = 5.04$ (38.92 versus 38.89), after which additional activation reduces the average, largely because the code and math tasks degrade. The six-expert checkpoint shows a broader plateau: it exceeds the Top-6 average from $A = 4.17$ through $A = 8.96$ and peaks at 39.36 with $A = 6.01$. Thus, the useful test-time range depends on the sparsity target learned during training rather than increasing monotonically with compute.

Table 7: Per-benchmark accuracy (%) for the 2B, 384-expert checkpoint trained with a target of four effective experts. A denotes the measured number of effective experts at each test-time operating point.

Benchmark	Top4	A3	A4	A5	A6	A7	A8	A9
MMLU	36.78	37.25	37.28	37.31	37.26	37.04	37.06	36.31
ARC-Easy	71.77	72.93	73.14	73.37	73.45	73.00	73.20	72.18
ARC-Challenge	39.31	42.24	42.59	42.70	43.20	41.97	41.51	40.27
TriviaQA	30.45	32.28	32.95	33.16	32.58	32.26	31.50	30.34
PopQA	11.00	12.60	12.10	11.70	11.70	11.50	11.70	11.10
BBH	27.57	27.08	27.22	27.02	27.78	27.08	27.53	25.57
HellaSwag	52.02	53.51	53.76	54.36	54.09	53.71	52.79	52.09
PIQA	71.35	71.86	72.04	72.20	72.59	72.17	72.13	71.37
WinoGrande	59.92	61.27	62.42	62.48	61.89	61.21	61.12	59.71
DROP	26.17	25.50	26.31	25.72	25.55	25.51	24.42	23.08
HumanEval	9.43	8.17	8.29	8.38	7.89	7.91	5.39	4.02
MBPP	26.38	25.74	26.24	26.05	24.39	24.02	21.49	18.76
GSM8K	31.92	29.11	31.69	31.16	26.99	24.72	20.55	16.38
Average	**38.01**	**38.43**	**38.92**	**38.89**	**38.41**	**37.85**	**36.95**	**35.47**
Effective experts	**4.00**	**2.99**	**4.00**	**5.04**	**6.00**	**6.98**	**7.99**	**8.99**

Table 8: Per-benchmark accuracy (%) for the 2B, 384-expert checkpoint trained with a target of six effective experts. A denotes the measured number of effective experts at each test-time operating point.

Benchmark	Top6	A3	A4	A5	A6	A7	A8	A9
MMLU	37.46	36.37	37.12	37.25	37.27	37.26	37.31	37.24
ARC-Easy	73.04	71.46	73.50	73.62	73.70	73.77	74.04	74.20
ARC-Challenge	41.08	40.23	42.12	42.74	42.78	43.36	43.44	43.17
TriviaQA	15.96	30.63	32.48	33.12	33.63	33.63	33.45	33.23
PopQA	11.10	12.00	11.70	11.50	11.30	11.40	11.50	11.60
BBH	30.56	28.30	30.00	30.09	29.85	30.46	29.08	29.15
HellaSwag	53.02	53.04	54.71	55.38	55.38	55.50	55.53	55.19
PIQA	71.72	71.73	72.29	72.46	72.76	72.63	72.88	72.87
WinoGrande	60.25	61.90	62.47	62.38	62.49	62.38	61.62	61.44
DROP	30.08	25.97	28.75	28.50	29.11	28.35	28.23	28.19
HumanEval	13.24	7.60	7.66	8.04	8.14	7.81	8.33	8.10
MBPP	26.15	27.36	25.49	27.15	26.20	25.16	23.54	22.41
GSM8K	33.59	22.29	26.99	27.45	29.11	28.28	27.75	27.75
Average	**38.25**	**37.61**	**38.87**	**39.21**	**39.36**	**39.23**	**38.98**	**38.81**
Effective experts	**6.00**	**3.09**	**4.17**	**5.06**	**6.01**	**7.00**	**8.05**	**8.96**

2B models with 64 experts Both gate configurations outperform the Top-4 baseline before reaching four effective experts: the 24-gate checkpoint obtains 41.62 at $A = 3.02$, and the 16-gate checkpoint obtains 42.05 at $A = 3.14$, compared with 41.14 for Top-4. Their averages are stable over a wide middle range, peaking at 42.53 ($A = 7.00$) and 42.75 ($A = 6.99$), respectively. The

task-level rows also show that the aggregate plateau masks different responses: several knowledge and commonsense tasks improve with activation, whereas DROP declines and code or math tasks vary by checkpoint.

Table 9: Per-benchmark accuracy (%) for the 2B, 64-expert checkpoint with 24 gates and a training target of four effective experts. A denotes the measured number of effective experts at each test-time operating point.

Benchmark	Top4	A2	A3	A4	A5	A6	A7	A8
MMLU	37.52	36.05	38.58	38.75	39.03	38.84	38.82	38.91
ARC-Easy	72.52	72.04	74.64	75.60	76.01	76.35	76.31	76.10
ARC-Challenge	40.58	40.35	43.55	44.44	45.17	45.33	45.52	45.60
TriviaQA	33.66	29.60	35.24	36.24	35.90	36.10	35.86	35.41
PopQA	12.60	10.70	12.80	12.30	12.90	13.70	13.90	13.50
BBH	31.76	29.03	30.76	31.21	31.35	31.01	31.35	30.65
HellaSwag	55.36	53.87	57.54	58.17	58.42	58.59	58.65	58.51
PIQA	72.12	72.13	73.59	73.94	73.80	73.66	73.74	73.53
WinoGrande	65.75	64.64	66.23	67.06	67.52	67.64	67.81	66.91
DROP	37.46	29.08	32.58	32.97	33.59	33.19	32.71	31.92
HumanEval	11.17	8.50	11.05	11.28	10.29	11.81	11.20	9.51
MBPP	29.09	20.11	24.42	24.89	25.37	24.75	24.74	24.08
GSM8K	35.25	28.81	40.03	42.76	41.77	41.47	42.23	41.93
Average	**41.14**	**38.07**	**41.62**	**42.28**	**42.39**	**42.50**	**42.53**	**42.04**
Effective experts	**4.00**	**2.02**	**3.02**	**3.97**	**5.08**	**6.00**	**7.00**	**8.02**

Table 10: Per-benchmark accuracy (%) for the 2B, 64-expert checkpoint with 16 gates and a training target of four effective experts. A denotes the measured number of effective experts at each test-time operating point.

Benchmark	Top4	A2	A3	A4	A5	A6	A7	A8
MMLU	37.52	37.26	38.54	38.96	38.91	38.89	39.07	38.88
ARC-Easy	72.52	73.56	74.89	75.54	75.56	75.64	75.76	75.76
ARC-Challenge	40.58	41.20	42.97	43.90	43.28	43.86	44.09	43.40
TriviaQA	33.66	31.02	34.16	34.97	34.74	34.79	34.83	34.50
PopQA	12.60	11.40	12.80	12.90	13.20	13.10	13.20	12.90
BBH	31.76	28.58	30.06	30.11	29.88	30.49	30.81	30.09
HellaSwag	55.36	55.35	57.49	57.73	57.85	58.06	58.31	58.19
PIQA	72.12	72.34	73.15	73.14	73.22	73.20	73.19	73.24
WinoGrande	65.75	64.54	65.12	65.83	65.97	65.86	65.98	66.16
DROP	37.46	26.99	27.06	27.04	27.50	26.80	25.61	24.81
HumanEval	11.17	10.65	17.21	18.16	19.05	20.22	20.01	21.13
MBPP	29.09	27.73	31.53	31.61	30.87	30.58	30.99	30.31
GSM8K	35.25	31.99	41.62	42.00	43.82	44.05	43.90	45.56
Average	**41.14**	**39.43**	**42.05**	**42.45**	**42.60**	**42.73**	**42.75**	**42.69**
Effective experts	**4.00**	**2.09**	**3.14**	**4.00**	**5.00**	**6.00**	**6.99**	**7.88**

7B and 10B models The same pattern persists at larger scales. The 7B checkpoint already exceeds Top-4 at $A = 3.03$ (52.51 versus 51.53) and remains between 53.44 and 53.78 from roughly four to eight effective experts. The 10B checkpoint exceeds Top-6 with only $A = 4.04$ (50.22 versus 49.18), reaches 51.39 at $A = 7.00$, and maintains a broad plateau before declining at the most compute-intensive settings. These sweeps show that the skip bias provides a meaningful inference-time control, but its best setting is checkpoint and task dependent.

Table 11: Per-benchmark accuracy (%) for the 7B, 64-expert checkpoint across test-time activation levels. A denotes the measured number of effective experts.

Benchmark	Top4	A2	A3	A4	A5	A6	A7	A8
MMLU	44.05	43.15	45.26	45.60	45.92	46.30	46.26	46.31
ARC-Easy	80.62	79.64	82.11	82.39	82.74	82.93	83.01	82.82
ARC-Challenge	50.97	51.85	55.17	55.91	55.71	55.75	55.83	55.79
TriviaQA	49.88	47.25	52.11	52.87	53.07	52.86	52.45	52.32
PopQA	16.30	15.40	17.50	17.30	16.90	17.30	17.40	17.10
BBH	40.10	33.25	37.48	38.60	38.96	39.10	39.87	39.85
HellaSwag	66.91	63.81	67.91	68.96	69.26	69.65	69.69	69.86
PIQA	76.24	76.02	76.99	77.29	77.37	77.58	77.66	77.60
WinoGrande	75.32	71.03	73.68	75.60	75.85	76.26	76.43	76.04
DROP	51.32	43.15	48.77	50.17	50.74	50.72	50.65	49.63
HumanEval	20.66	14.41	22.29	22.45	19.84	21.97	19.99	19.78
MBPP	40.33	39.42	43.66	44.47	45.52	45.13	45.36	44.91
GSM8K	57.16	45.34	59.74	63.15	63.23	63.61	62.24	63.15
Average	**51.53**	**47.98**	**52.51**	**53.44**	**53.47**	**53.78**	**53.60**	**53.47**
Effective experts	**4.00**	**2.00**	**3.03**	**3.98**	**5.10**	**6.09**	**7.09**	**7.90**

Table 12: Per-benchmark accuracy (%) for the 10B, 384-expert checkpoint trained with a target of six effective experts. A denotes the measured number of effective experts at each test-time operating point.

Benchmark	Top6	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
MMLU	43.09	43.85	44.43	45.21	45.06	45.31	45.18	45.22	45.23	45.29	45.19	45.10	44.71	44.76
ARC-Easy	78.87	79.45	80.26	80.76	80.80	80.66	80.57	80.35	80.68	80.47	80.49	80.68	80.26	79.93
ARC-Challenge	49.27	51.35	52.43	53.01	53.55	53.44	53.63	53.59	53.71	54.36	53.75	53.44	53.20	52.32
TriviaQA	46.75	48.91	50.45	51.27	51.35	51.77	51.55	51.65	51.69	51.63	51.42	50.91	50.63	49.48
PopQA	15.00	15.40	16.30	16.30	16.30	15.90	15.70	16.00	16.20	15.80	16.00	16.30	15.90	15.40
BBH	34.11	31.66	33.05	34.51	34.24	34.09	34.42	34.58	34.04	34.31	34.26	34.10	33.77	31.87
HellaSwag	64.89	64.43	66.02	66.73	66.99	67.14	67.17	67.47	67.70	67.53	67.51	67.49	67.39	67.10
PIQA	76.31	76.70	77.27	77.38	77.36	77.33	77.40	77.26	77.43	77.05	77.18	77.09	76.99	77.00
WinoGrande	70.41	70.16	71.70	72.05	72.06	72.26	72.34	72.42	72.29	72.17	71.69	71.59	71.24	70.19
DROP	46.42	40.71	43.65	44.69	45.21	45.37	45.03	45.14	44.36	43.98	43.96	42.78	42.26	40.10
HumanEval	19.23	17.72	20.77	21.42	21.95	21.93	22.35	21.61	21.65	19.95	19.09	19.74	18.81	17.93
MBPP	40.58	38.67	40.71	41.46	41.46	41.52	40.81	41.59	40.94	41.21	41.14	39.96	38.14	35.02
GSM8K	54.36	49.81	55.88	59.74	59.59	61.33	60.35	60.88	60.96	60.12	59.21	57.85	55.27	52.77
Average (13)	**49.18**	**48.37**	**50.22**	**51.12**	**51.22**	**51.39**	**51.27**	**51.37**	**51.30**	**51.07**	**50.80**	**50.54**	**49.89**	**48.76**
Effective experts	**6.00**	**3.03**	**4.04**	**4.98**	**6.02**	**7.00**	**8.01**	**8.99**	**9.98**	**11.01**	**11.99**	**12.98**	**13.91**	**14.87**