EXPERT-GUIDED ROUTER RECOVERY FOR STABLE MIXTURE-OF-EXPERTS PRE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-experts (MoE) language models expand model capacity while limiting per-token computation through sparse expert activation. However, loss spikes during from-scratch MoE pre-training remain a critical yet underexplored failure mode. To address this, we propose Expert-Guided Router Recovery (EGRR), a pre-training recovery method that bridges the gap between router decision-making and expert execution by enabling the router to assign tokens to experts according to capability information derived from the experts themselves. EGRR is based on the insight that around loss spikes, router-selected experts exhibit unusually low activation magnitudes, suggesting limited capability for processing the current representations. In EGRR, once a loss spike is detected, EGRR rolls training back to the latest stable checkpoint and densely activates all experts to guide router tuning with expert-derived capability information, while all non-router parameters are frozen. After 100M tokens recovery period, dense expert activation is removed and standard MoE pre-training resumes. EGRR therefore introduces additional computation only during recovery, with no additional inference cost. We evaluate EGRR in a 100B tokens from-scratch pre-training run of a 700M parameters MoE language model,demonstrating improved training stability and downstream performance over standard MoE training. The code is available at `https://github.com/saz97/Expert-GuidedRouter-Recovery`.

1 INTRODUCTION

Models built with Mixture-of-Experts (MoE, Shazeer et al. (2017); Fedus et al. (2021); Lepikhin et al. (2020); Neitemeier et al. (2026)) technology have become an important architecture for modern large language models (LLMs). The core idea of MoE is to split a large feed-forward network (FFN) into multiple smaller FFNs, referred to as experts, and activate different expert parameters for different inputs. A router selects the experts that process each input token. By activating only a small number of experts for each token, MoE achieves a balance between computational efficiency and parameter scaling, making it possible to train trillion-parameter models. The ability to flexibly combine knowledge from different experts also enhances downstream task performance.

However, a key issue in MoE is often overlooked: the router's decisions and the experts' execution are not fully aligned (Lv et al. (2026)). The router's selections may be misaligned with the capabilities that experts exhibit on the current inputs. As a result, the router must learn more appropriate decisions through repeated trial and error, which can waste many training steps. As the model scale increases, interactions between the router and experts may further destabilize training, with loss spikes being a representative phenomenon (DeepSeek-AI et al. (2026)).

During LLM pre-training, the loss may suddenly diverge (Chowdhery et al. (2022); Zhang et al. (2022)). This phenomenon, known as a loss spike, is a fundamental problem in LLM pre-training: it not only increases the final loss, but may also cause pre-training to fail when the loss diverges completely. To address these problems, we propose a new MoE pre-training paradigm, termed Expert-Guided Router Recovery (EGRR).

Our approach is based on the following observation: near naturally occurring loss spikes, the experts selected by the router often exhibit abnormally small output magnitudes. Reselecting and reweighting the Top-K experts according to their output magnitudes can substantially alleviate loss spikes. Based on this insight, we decouple the synchronous updates of the backbone network and the router.

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

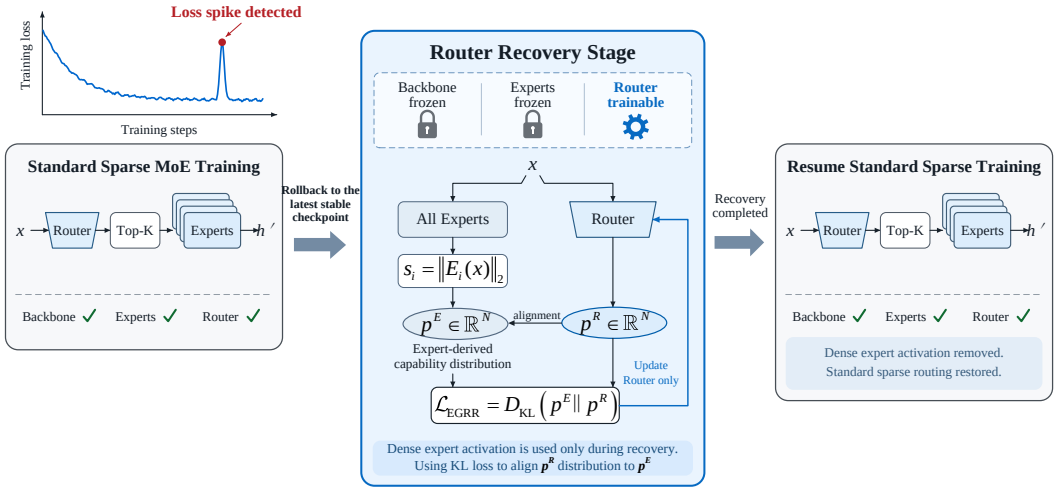


Figure 1: Overview of the Expert-Guided Router Recovery (EGRR) procedure.

When a loss spike is detected, EGRR rolls the training process back to a recent stable checkpoint and enters a recovery stage. During the recovery stage, the model densely activates all experts and constructs a target distribution based on the magnitudes of their outputs. This distribution is used as the supervision signal for updating the router. By fitting this target distribution, the router learns to produce an expert-selection distribution. After a fixed recovery stage of 100M tokens, dense expert computation is disabled and standard Top-K sparse routing is restored.

We conduct pre-training experiments on a 700M-parameter MoE language model and further scale to a 4B-parameter MoE model to evaluate EGRR scalability and downstream task performance. The results show that EGRR effectively reduces the impact of loss spikes on subsequent training, improves training stability, and achieves better downstream performance under the same training budget. Figure 1 illustrates the EGRR procedure.

2 Background

2.1 Mixture-of-Experts

We focus on sparse Mixture-of-Experts (MoE) models and treat each feed-forward network (FFN) module as an expert (Jiang et al. (2024); Dai et al. (2024)). Different experts are expected to learn diverse and distinct capabilities, allowing the model to activate only a small number of experts relevant to each input. This design enables MoE models to increase their parameter capacity while reducing the computational cost per token. Many recent large language models use SwiGLU (Touvron et al. (2023); Qwen et al. (2025); DeepSeek-AI et al. (2025)) as the basic structure of their expert modules.

An MoE layer consists of N experts. The i-th expert is parameterized by three weight matrices, W_g^i, W_p^i, and W_o^i, and is defined as

$$E_i(x) = \left[\mathrm{SiLU}(xW_g^i) \odot (xW_p^i)\right] W_o^i, \tag{1}$$

where $x \in \mathbb{R}^{d_{\text{model}}}$ denotes the input hidden state, $\odot$ denotes element-wise multiplication, $W_g^i, W_p^i \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ffn}}}$, and $W_o^i \in \mathbb{R}^{d_{\text{ffn}} \times d_{\text{model}}}$.

The MoE layer also contains a router with a parameter matrix $R \in \mathbb{R}^{N \times d_{\text{model}}}$. Given an input token x, the router first computes a routing logit for each expert, which is then converted into expert weights using the softmax function:

$$w = \mathrm{softmax}(Rx) \in \mathbb{R}^N. \tag{2}$$

In general, the router selects the Top-K experts with the highest weights to process the input token (Shazeer et al. (2017); Lepikhin et al. (2020)). Let $I = \mathrm{TopK}(w)$ denote the selected expert indices,

and let $\hat{w}$ denote the weights renormalized over the selected experts. The output of the MoE layer is then given by

$$h = \sum_{k \in I} \hat{w}_k E_k(x), \qquad \hat{w}_k = \frac{w_k}{\sum_{j \in I} w_j}. \tag{3}$$

Algorithm 1 presents the basic computation flow of an MoE layer with N experts. The indexing notation follows Python syntax and denotes the corresponding element of a vector or matrix.

Algorithm 1 Forward pass of an MoE layer

Require: Hidden state $x \in \mathbb{R}^{d_{\text{model}}}$; number of experts N; number of selected experts K; router matrix R; expert functions $E_1, \ldots, E_N$
Ensure: Layer output $h \in \mathbb{R}^{d_{\text{model}}}$

$p \leftarrow Rx$
$I \leftarrow \text{TopKIndices}(p, K)$
$p_{\text{selected}} \leftarrow p[I]$
$\hat{w} \leftarrow \text{Softmax}(p_{\text{selected}})$
$h \leftarrow \mathbf{0}$
for $j = 1$ to K **do**
    $i \leftarrow I[j]$
    $h \leftarrow h + \hat{w}_j E_i(x)$
end for
return h

2.2 MoE Training Instability

Training large-scale MoE models presents significant stability challenges. During the pre-training of large language models, the loss may suddenly increase or diverge, resulting in a loss spike. Previous studies have proposed several strategies to mitigate this problem (Chen et al. (2021); Ding et al. (2021); Molybog et al. (2023); Takase et al. (2025); Nishida et al. (2024)) . For example, OPT-175B manually skipped data batches and adjusted hyperparameters to address loss spikes (Zhang et al. (2022)), whereas BLOOM-176B employed an embedding-norm technique (Workshop et al. (2023)) . GLM-130B reduced the gradients of the embedding layer to lower the likelihood of loss spikes (Zeng et al. (2023)) .

However, existing studies have focused primarily on dense Transformers or specific large language model architectures. For MoE Transformers, the relationships among loss spikes, anomalous expert activations, and routing decisions remain insufficiently understood. During the training of a 4B-parameter MoE model, we observed clear loss-spike phenomena, as shown in Figure 2.

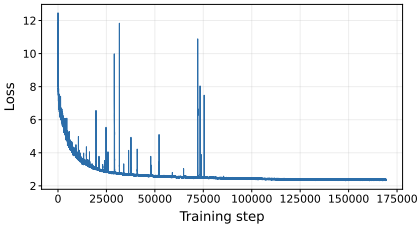


Figure 2: Loss spikes observed during the training of a 4B-parameter MoE model.

Training experience from DeepSeek-V4 suggests that loss spikes are generally associated with outliers in MoE layers (DeepSeek-AI et al. (2026)). The authors proposed Anticipatory Routing to improve training stability. Nevertheless, the underlying mechanism of this method is not yet fully understood. Therefore, our work further investigates loss spikes in MoE models and analyzes their relationship with expert activations and routing mechanisms.

3 Method

We begin by introducing the preliminary experiments that motivate the development of Expert-Guided Router Recovery (EGRR) in Section 3.1. In Section 3.2, we refine the method in the preliminary experiments and address the problem by improving the MoE training paradigm, ultimately deriving the EGRR method.

3.1 An Insight: Abnormal Expert Outputs during Loss Spikes

Inspired by the observation in DeepSeek V4 that training instability and the occurrence of loss spikes are consistently associated with anomalous values in MoE layers (DeepSeek-AI et al. (2026)), we first investigate the behavior of experts when loss spikes occur. We record the outputs of experts selected by the router around a loss spike and analyze their scales. We observe that, at the checkpoint where a loss spike occurs, the scale of the selected experts' outputs exhibits a pronounced abnormal decrease. Figure 3(a) presents a representative training process, where the loss curve is obtained from an actual pre-training run.

This phenomenon motivates us to further investigate whether loss spikes are related to the compatibility between the experts selected by the router and the current tokens. Prior work on Autonomy-of-Experts (AoE, Lv et al. (2025)) for MoE models has shown that the scales of experts' activation or output can, to some extent, reflect how well their capabilities match the current inputs. Based on this work, we hypothesize that the abnormally small outputs of the selected experts during loss spikes may indicate a mismatch between the router's expert selection and the experts' capabilities for current inputs.

To preliminarily validate this hypothesis, we design a preliminary experiment that requires no parameter updates. Specifically, for an MoE training run in which a loss spike has occurred, we roll back to the most recent normal checkpoint and perform forward again using the same data that caused the spike. Unlike the traditional MoE forward pass, we remove the router and allow all experts in the layer to process the current inputs. We then compute the L2 norm of each expert's output and rank the experts according to their output norms. Only the Top-K experts are retained, and their outputs are used to produce the final MoE output. This procedure is illustrated in Figure 3(b). Notably, no model parameters are updated in this experiment, and the input data remain unchanged; only the expert selection mechanism is modified. We use MMLU (Hendrycks et al. (2021)) and ARC-Challenge (Clark et al. (2018)) to investigate the performance of these expert selection mechanisms. The test code is based on the LM Evaluation Harness (Gao et al. (2023)) with a batch size of 50. The detail result is shown in Appendix A.

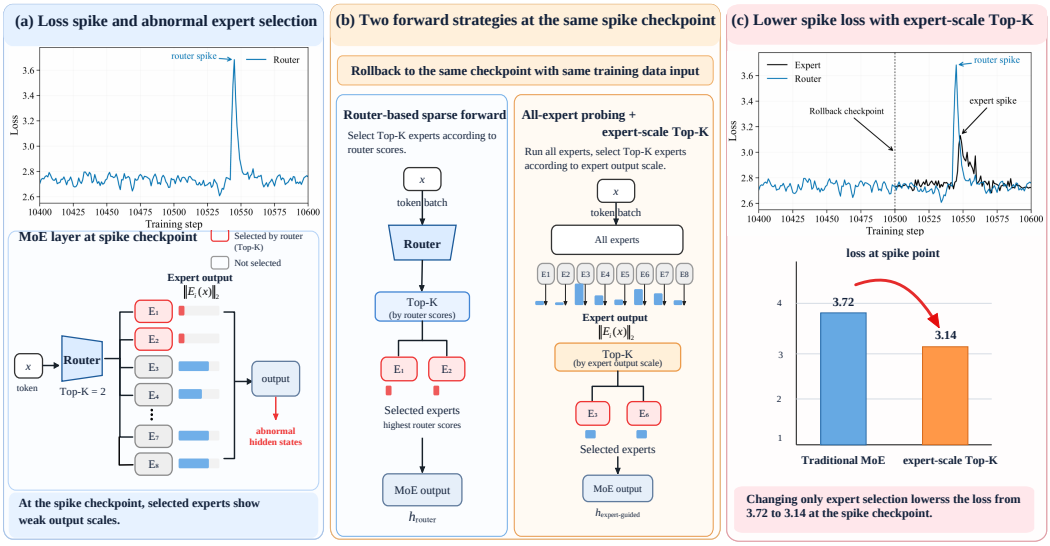


Figure 3: Preliminary observations and experiments motivating Expert-Guided Router Recovery (EGRR). (a) Abnormal expert output scales observed around a loss spike during pre-training. (b) The preliminary expert-guided selection procedure, where the router is removed and experts are ranked according to the L2 norms of their outputs. (c) Comparison between the conventional MoE forward pass and the expert-guided selection strategy at the loss spike.

The results are presented in Figure 3(c). Compared with the conventional MoE, this preliminary approach reduces the peak loss at the spike. These preliminary results motivate us to propose a new training paradigm for MoE models. We aim to leverage capability signals from the experts

themselves to guide router recovery after a loss spike, thereby reducing the impact of loss spikes on subsequent training and improving training stability.

3.2 EXPERT-GUIDED ROUTER RECOVERY (EGRR)

In the following, we focus on how to leverage capability signals from the experts themselves to calibrate the router, thereby improving the training stability of MoE models. The preliminary experiments in Section 3.1 demonstrate that selecting experts according to the scales of their outputs can effectively reduce the peak loss during a loss spike. To utilize this signal while preserving the sparse computation of MoE models, we align the routing distribution with the capability distribution reflected by expert output scales, allowing the router to learn the relative capabilities of experts for the current input.

As defined in Eq. equation 2, given an input token x, the router produces a probability distribution over experts:

$$p^R(x) = w = \mathrm{softmax}(Rx) \in \mathbb{R}^N. \tag{4}$$

where $p_i^R(x)$ denotes the probability assigned by the router to the i-th expert. On the other hand, based on the observation in Section 3.1, we use the scale of an expert's output as a signal of its capability to process the current input. We define the output scale of the i-th expert as

$$s_i(x) = \|E_i(x)\|_2. \tag{5}$$

We further normalize this signal across all experts to obtain the relative capability score of the i-th expert for the current input:

$$p_i^E(x) = \frac{\exp(s_i(x))}{\sum_{j=1}^{N} \exp(s_j(x))}. \tag{6}$$

Here, p^E represents the relative capability distribution exhibited by all experts for the current token, while p^R represents the router's prediction of expert suitability. We treat p^E as a supervision signal produced by the experts themselves and align the router-predicted distribution p^R with this distribution. Based on the router–expert mismatch observed in Section 3.1, we aim to realign the router's predictions with the capabilities currently exhibited by the experts. We therefore define the Expert-Guided Router Recovery loss as

$$L_{\mathrm{EGRR}} = D_{\mathrm{KL}}(p^E \| p^R) = \sum_{i=1}^{N} p_i^E \log \frac{p_i^E}{p_i^R}. \tag{7}$$

A straightforward approach would be to use L_{EGRR} as an auxiliary loss throughout pre-training and imize it with the language modeling loss. However, this approach introduces two problems. First, if the router and expert parameters are updated simultaneously, the gradients induced by EGRR will also alter the expert capability distribution in return. Introducing such an additional optimization objective may interfere with the original language modeling training process. Second, computing p^E requires obtaining the outputs of all experts. Activating all experts for every token, however, would eliminate the computational advantage of sparse activation in MoE models, making this approach impractical for conventional large-scale pre-training.

Based on these considerations, rather than continuously applying L_{EGRR} throughout pre-training, we introduce a short recovery stage that is triggered only after a loss spike occurs. Algorithm 2 presents the complete pre-training procedure with EGRR.

When a loss spike is detected, we first roll back training to the most recent normal checkpoint and then enter EGRR recovery. During the recovery stage, we freeze the Transformer backbone and all expert parameters, leaving only the router parameters trainable. For each input token, we activate all experts to construct the capability distribution p^E and optimize the router solely using L_{EGRR}. No language modeling loss is included during the recovery stage. Since both the backbone and experts remain fixed, the existing model capabilities reflected by p^E remain stable throughout recovery, and optimization only changes how the router maps the existing token representations to different experts. Therefore, the objective of EGRR is not to alter the network's capabilities to adapt to the router, but rather to realign the router with the expert capabilities that have already been established in the network. The recovery stage lasts for 100M tokens. After recovery, we

Algorithm 2 Expert-Guided Router Recovery (EGRR)

Require: MoE model with backbone parameters θ_{backbone}, expert parameters θ_{expert}, and router parameters θ_{router}; number of experts N; recovery budget $T_{\text{rec}} = 100\text{M}$ tokens
Ensure: Recovered MoE model

```
Perform standard sparse Top-K MoE pre-training
if a loss spike is detected then
    Roll back to the most recent stable checkpoint
    Freeze θ_backbone and θ_expert
    Keep only θ_router trainable
    Enable dense activation of all experts
    while processed recovery tokens < T_rec do
        for each input token x do
            Compute router distribution:
    p^R(x) ← softmax(Rx)
            for i = 1 to N do
                Compute expert output E_i(x)
                Compute expert output scale:
    s_i(x) ← ||E_i(x)||_2
            end for
            Construct expert capability distribution:
    p_i^E(x) ← exp(s_i(x)) / Σ_{j=1}^{N} exp(s_j(x))
            Compute recovery loss:
    L_EGRR ← D_KL(p^E(x) || p^R(x))
        end for
        Update only θ_router using L_EGRR
    end while
    Disable dense expert activation
    Restore Top-K sparse routing
    Unfreeze all model parameters
    Resume standard sparse MoE pre-training
end if
```

disable dense expert activation, restore the original router Top-K sparse routing, unfreeze all model parameters, and resume standard sparse MoE language model pre-training. Since EGRR serves only as a temporary recovery mechanism after a loss spike occurs during training, it does not alter the final model architecture, the number of activated parameters, or the inference cost.

4 Experiments

We compare EGRR with several baselines, including vanilla MoE and Anticipatory Routing. We first conduct experiments on small-scale pre-trained language models to provide a detailed analysis of our method. Based on the insights gained from these experiments, we further scale the language models to 4B parameters to evaluate the scalability of EGRR.

4.1 Experimental Setup

We train small language models consisting of 12 layers, each containing 12 attention heads (Vaswani et al. (2023)). Each layer contains 8 experts, with the Top-$K = 2$ experts selected for each token. The models use the Llama vocabulary (Grattafiori et al. (2024)) with a size of 32,000 and adopt the same pre-RMSNorm module (Zhang & Sennrich (2019)). For the MoE models, we set $d_{\text{model}} = 640$ and $d_{\text{ffn}} = 3,240$. The total number of model parameters is 700M, of which 247M parameters are activated.

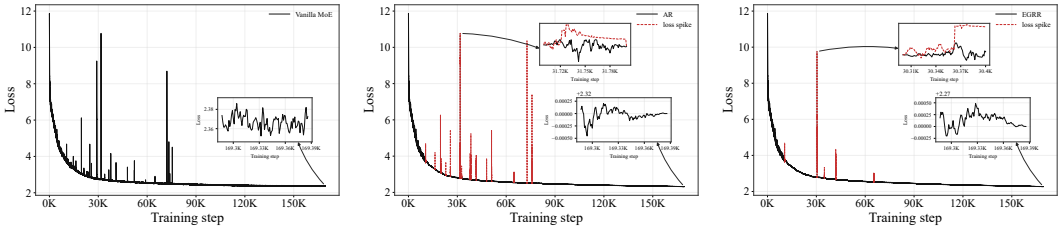
We train the models on 100B tokens from the open-source `dolmap-v1.5-sample` dataset (Soldaini et al. (2024)), with a batch size of 4.90M tokens. We use the AdamW optimizer with

$(\beta_1, \beta_2) = (0.9, 0.95)$ (Loshchilov & Hutter (2019)), a gradient norm clipping threshold of 1, and a weight decay of 0.1. The learning rate is set to 4×10^{-4}. We apply linear warmup over the first 4,800 steps, followed by a cosine decay schedule that reduces the learning rate to 4×10^{-5}. A load balancing loss Fedus et al. (2021) with a weight of 0.01 is consistently applied in all experiments.

We conduct a comprehensive evaluation of the language models across a range of widely used tasks, including ARC-C (Clark et al. (2018)), PIQA (Bisk et al. (2019)), SIQA (Sap et al. (2019)), WINO (Sakaguchi et al. (2019)), HELLA (Zellers et al. (2019)), MNLI (Williams et al. (2018)), QNLI (Wang et al. (2019)) and SST-2 (Socher et al. (2013)).

4.2 Training Stability and Load Balancing

Figures 4(a), 4(b), and 4(c) present the training loss during training for vanilla MoE, Anticipatory Routing, and EGRR, respectively, to compare training stability of the three methods. The red dashed lines indicate the loss spikes detected during training. The black solid lines represent the loss trajectories during normal training, as well as the loss trajectories obtained by retraining with the recovery strategies after a loss spike occurs.



(a) Vanilla MoE (b) Anticipatory Routing (c) EGRR

Figure 4: Pre-training losses. To provide a more intuitive comparison of the training behavior of different methods, we additionally report the final converged loss and compare the loss before rollback at the largest loss spike with that after recovery. For better visualization of the differences among the methods at the largest spike, this comparison is presented on a logarithmic scale.

As shown by the training trajectories, vanilla MoE exhibits pronounced loss spikes during pre-training. Anticipatory Routing effectively mitigates this training instability and reduces the impact of loss spikes on subsequent training. In comparison, EGRR exhibits a more stable training trajectory after recovery and further reduces the impact of loss spikes. Moreover, the training process with EGRR converges to a lower final loss than baseline methods.

We further investigate whether EGRR affects the original load balancing behavior of MoE models. As shown in Figure 5, EGRR is highly compatible with the original load balancing loss. The difference in load balancing loss between the MoE with EGRR and vanilla MoE is on the order of 10^{-5}. Considering that the overall magnitude of the load balancing loss remains around 10^{-2}, this difference is negligible. In comparison, the difference between Anticipatory Routing and vanilla MoE is approximately 1.1×10^{-4}. Although this difference is still small, it is notably larger than that introduced by our method.

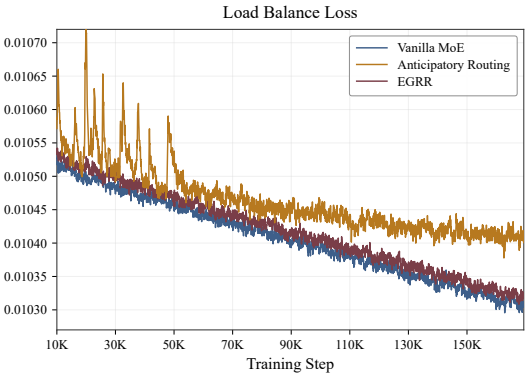


Figure 5: Load balancing loss curves.

In addition, we observe that the load balancing loss of Anticipatory Routing exhibits noticeable abnormal spikes near some rollback points, and these spikes are associated with the timing of rollbacks during training. This phenomenon may arise from the state mismatch between the router and the backbone network introduced by Anticipatory Routing after rollback. Specifically, after rollback, Anticipatory Routing

Table 1: Scaling to 4B parameters: EGRR improves performance on downstream benchmarks.

Method	ARC-C	PIQA	SIQA	WINO	HELLA	MNLI	QNLI	SST-2	Avg.
vanilla MoE	39.10	57.90	35.20	51.20	27.20	32.60	48.30	48.42	42.49
Anticipatory Routing	40.10	58.40	36.10	52.00	28.00	33.20	49.80	50.48	43.51
EGRR	**40.80**	**59.00**	**36.90**	**52.80**	**28.70**	**33.90**	**50.50**	**50.99**	**44.20**

uses a router from an earlier training state than the current backbone network. As a result, there may be a distribution shift between the input representations to which the router was adapted and the token representations produced by the current backbone network. In other words, from the router's perspective, the current representations may deviate from the distribution to which it was originally adapted, leading to transient abnormal routing behavior and manifesting as spikes in the load balancing loss around the rollback points. In contrast, EGRR directly recalibrates the router after rollback based on the capability signals exhibited by the experts at the current checkpoint, and therefore does not exhibit load balancing loss anomalies of a comparable magnitude.

4.3 Scaling Experiments on the 4B-Parameter MoE

To evaluate the effectiveness of EGRR at a larger model scale, we further conduct experiments on an MoE model with approximately 4B total parameters and approximately 1.18B activated parameters per token. The model consists of 24 layers, each with 20 attention heads. For the standard MoE architecture, we set $d_{\text{model}} = 1,280$ and $d_{\text{ffn}} = 5,120$, with an initial learning rate of 3.2×10^{-4}. Except for the above configurations, all other training settings are kept the same as those in Section 4.1. For all methods, we use the auxiliary load balancing loss L_{aux}, with $\alpha_{\text{aux}} = 0.01$.

Table 1 presents the zero-shot performance of vanilla MoE, Anticipatory Routing, and EGRR on eight downstream benchmarks, including ARC-C, PIQA, SIQA, WINO, HELLA, MNLI, QNLI, and SST-2. EGRR achieves the best performance on all evaluation tasks, indicating that its performance gains are not limited to smaller-scale models and can scale well to larger MoE models.

To further analyze the performance changes of vanilla MoE, Anticipatory Routing, and EGRR during training, we present the average downstream task accuracy at different training stages in Figure 6.

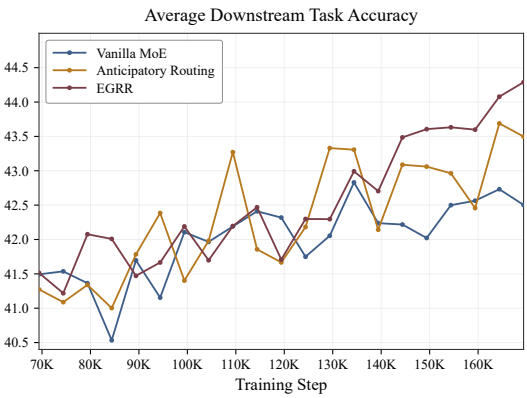
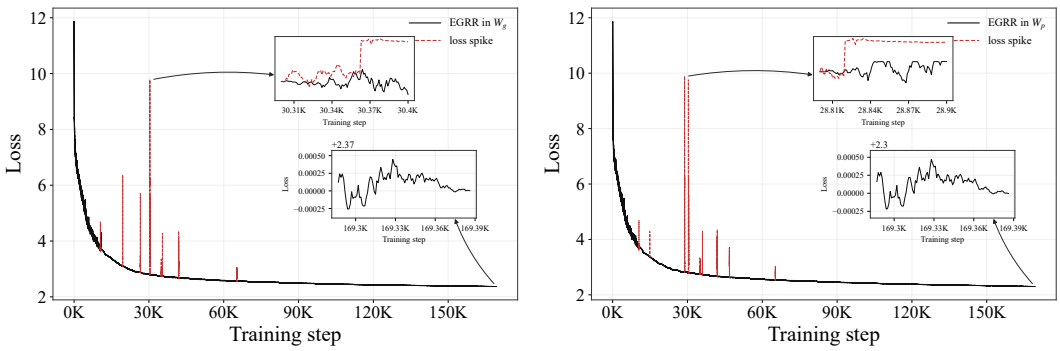


Figure 6: Average downstream task accuracy. We save a checkpoint every 5,000 training steps and perform zero-shot evaluation on the eight benchmarks described above. The average accuracy of each checkpoint is computed as the arithmetic mean of the evaluation results on the eight benchmarks. For detailed task-specific results, please refer to Appendix B.

As can be observed from Figure 6, the performance advantage of EGRR does not appear only at the final checkpoint, but is gradually established and maintained throughout training. This result further demonstrates that, at a larger model scale, EGRR utilizes the information contained in expert scales to adjust routing decisions, effectively addressing the separation between router decision-making and expert execution, thereby leading to more stable downstream performance gains.

4.4 Ablation Study

We consider three candidates for constructing the expert capability signal used in EGRR, using the scales of activations at the following positions: (a) xW_g, as shown in Figure 7a; (b) xW_p, as shown in Figure 7b; and (c) the final expert output, as defined in Eq. equation 1. As shown in Figure 7, using the scales of xW_g and xW_p also improves training stability and achieves some improvement in the



(a) xW_g (b) xW_p

Figure 7: Results of ablation studies.

final loss. However, compared with using the scale of the final expert output, these two alternatives still exhibit a certain performance gap. Therefore, in EGRR, we use the scale of the final expert output as the default signal for constructing the expert capability distribution.

5 Related Work

Auxiliary Losses for MoEs Auxiliary losses are crucial for training large-scale MoE models. A large body of existing work focuses primarily on load balancing to prevent a small number of experts from being overloaded or some experts from being insufficiently trained (Wang et al. (2024); Qiu et al. (2025)). Prior work has proposed the z-loss (Zoph et al. (2022)), which penalizes excessively large logits in the gating network to improve training stability. DeepSeek further proposed a loss-free load balancing method that adjusts the loads of different experts without introducing additional auxiliary losses (Dai et al. (2024)). The MoE concept has also been extended to mixtures of attention heads or entire layers, where auxiliary losses similarly play an important role in effective optimization (Gong et al. (2024); Lin et al. (2024)). Our EGRR is the first method that uses only a stage-wise auxiliary loss for training recovery during MoE pre-training.

Training Stability To improve the training stability of Transformer-based neural language models, extensive studies have investigated model architectures (Xiong et al. (2020); Zhai et al. (2023)), initialization methods (Wang et al. (2022)), training strategies (Li et al. (2022)), and loss functions (Wortsman et al. (2023)). To stabilize LLM pre-training, prior work has applied layer normalization to the embedding layer, while other studies have adopted the shrink embedding gradient technique. In addition, some work has introduced additional parameters to scale the parameters of internal layers to improve training stability.

6 Conclusion

Loss spikes are a fundamental issue in MoE pre-training. We find that around loss spikes, there exists a mismatch between the router's decisions and expert capabilities. Based on this observation, we propose Expert-Guided Router Recovery (EGRR), a loss spike recovery method that uses expert output scales to provide supervision signals for the router. EGRR only introduces a short recovery stage while incurring no additional inference overhead. Experimental results show that EGRR improves training stability and downstream task performance. This work provides a practical method for recovering MoE models from loss spikes, and we hope it will encourage the research community to further explore more powerful MoE-like models.

Reproducibility Statement

Sections 3 and 4 specify the recovery objective, freezing protocol, model scale, routing configuration, data, optimizer family, learning-rate schedule, checkpoint interval, and token budgets. Training

configurations, spike logs, checkpoint identifiers, and evaluation scripts will accompany the supplementary material.

Ethics Statement

This work studies optimization stability and introduces no new data collection or human-subject experiments. It may reduce computation wasted by failed runs but inherits the risks of the underlying language model and training data.

AI Use Statement

Generative AI tools assisted with language editing and LaTeX preparation. The authors reviewed the research claims, design, measurements, and final text and take responsibility for the submission.

References

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.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers, 2021. URL https://arxiv.org/abs/2104.02057.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

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.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2401.06066.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng

Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. 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.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers, 2021. URL https://arxiv.org/abs/2105.13290.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *CoRR*, abs/2101.03961, 2021. URL https://arxiv.org/abs/2101.03961.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL https://zenodo.org/records/10256836.

Zhuocheng Gong, Ang Lv, Jian Guan, Junxi Yan, Wei Wu, Huishuai Zhang, Minlie Huang, Dongyan Zhao, and Rui Yan. Mixture-of-modules: Reinventing transformers as dynamic assemblies of modules, 2024. URL https://arxiv.org/abs/2407.06677.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor

Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

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.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *CoRR*, abs/2006.16668, 2020. URL https://arxiv.org/abs/2006.16668.

Conglong Li, Minjia Zhang, and Yuxiong He. The stability-efficiency dilemma: Investigating sequence length warmup for training gpt models, 2022. URL https://arxiv.org/abs/2108.06084.

Hongzhan Lin, Ang Lv, Yuhan Chen, Chen Zhu, Yang Song, Hengshu Zhu, and Rui Yan. Mixture of in-context experts enhance llms' long context awareness, 2024. URL https://arxiv.org/abs/2406.19598.

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

Ang Lv, Ruobing Xie, Yining Qian, Songhao Wu, Xingwu Sun, Zhanhui Kang, Di Wang, and Rui Yan. Autonomy-of-experts models, 2025. URL https://arxiv.org/abs/2501.13074.

Ang Lv, Jin Ma, Yiyuan Ma, and Siyuan Qiao. Coupling experts and routers in mixture-of-experts via an auxiliary loss, 2026. URL https://arxiv.org/abs/2512.23447.

Igor Molybog, Peter Albert, Moya Chen, Zachary DeVito, David Esiobu, Naman Goyal, Punit Singh Koura, Sharan Narang, Andrew Poulton, Ruan Silva, Binh Tang, Diana Liskovich, Puxin Xu, Yuchen Zhang, Melanie Kambadur, Stephen Roller, and Susan Zhang. A theory on adam instability in large-scale machine learning, 2023. URL https://arxiv.org/abs/2304.09871.

Pit Neitemeier, Jiaze Li, Alessio Serra, Philipp Scholl, and Sohir Maskey. Exact quantile balancing and load-error injection for mixture-of-experts, 2026. URL https://arxiv.org/abs/2609.28053.

Kosuke Nishida, Kyosuke Nishida, and Kuniko Saito. Initialization of large language models via reparameterization to mitigate loss spikes, 2024. URL https://arxiv.org/abs/2410.05052.

Zihan Qiu, Zeyu Huang, Bo Zheng, Kaiyue Wen, Zekun Wang, Rui Men, Ivan Titov, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Demons in the detail: On implementing load balancing loss for training specialized mixture-of-expert models, 2025. URL https://arxiv.org/abs/2501.11873.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions, 2019. URL https://arxiv.org/abs/1904.09728.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *CoRR*, abs/1701.06538, 2017. URL http://arxiv.org/abs/1701.06538.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (eds.), *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

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, 2024. URL https://arxiv.org/abs/2402.00159.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. Spike no more: Stabilizing the pretraining of large language models, 2025. URL https://arxiv.org/abs/2312.16903.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019. URL https://arxiv.org/abs/1804.07461.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers, 2022. URL https://arxiv.org/abs/2203.00555.

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.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2018. URL https://arxiv.org/abs/1704.05426.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan,

Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh Haji-Hosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman,

Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. Bloom: A 176b-parameter open-access multilingual language model, 2023. URL https://arxiv.org/abs/2211.05100.

Mitchell Wortsman, Peter J. Liu, Lechao Xiao, Katie Everett, Alex Alemi, Ben Adlam, John D. Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities, 2023. URL https://arxiv.org/abs/2309.14322.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture, 2020. URL https://arxiv.org/abs/2002.04745.

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.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Peng Zhang, Yuxiao Dong, and Jie Tang. Glm-130b: An open bilingual pre-trained model, 2023. URL https://arxiv.org/abs/2210.02414.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Josh Susskind. Stabilizing transformer training by preventing attention entropy collapse, 2023. URL https://arxiv.org/abs/2303.06296.

Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019. URL https://arxiv.org/abs/1910.07467.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

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.

A Additional Understanding Experiment

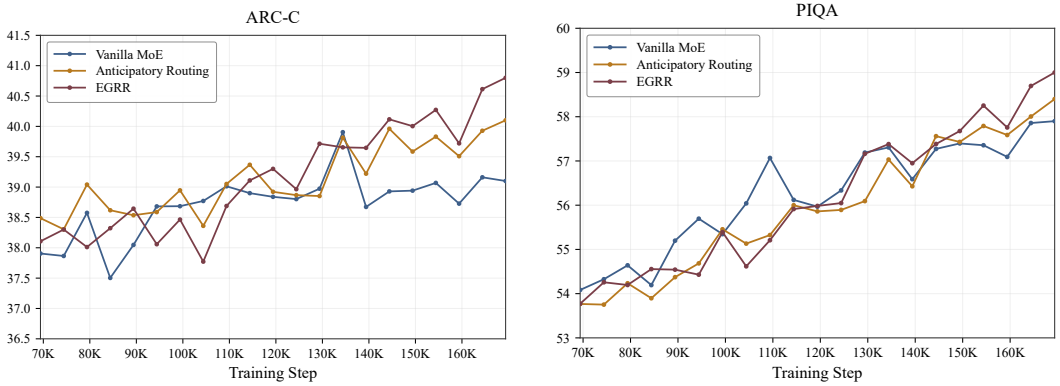
Both configurations in Table 2 start from the same rollback checkpoint and use the same evaluation protocol.

Table 2: Downstream accuracy (%) in the understanding experiment.

Configuration	ARC-C(5-shot)	MMLU(5-shot)
Router selection	19.90(19.65)	37.43(70.17)
Expert-scale Top-K selection	**20.74(17.09)**	**38.49(68.43)**

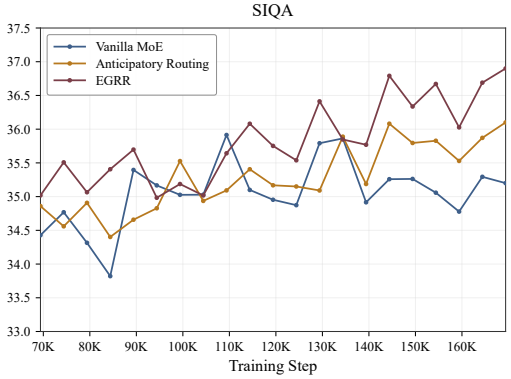
B Task-Specific Evaluation Results

Figure 8 presents the accuracy trajectories for each individual task.

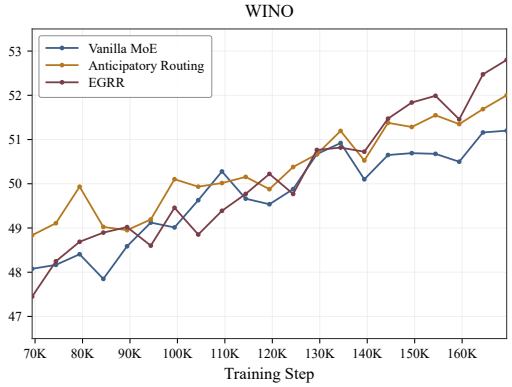


(a) ARC-C

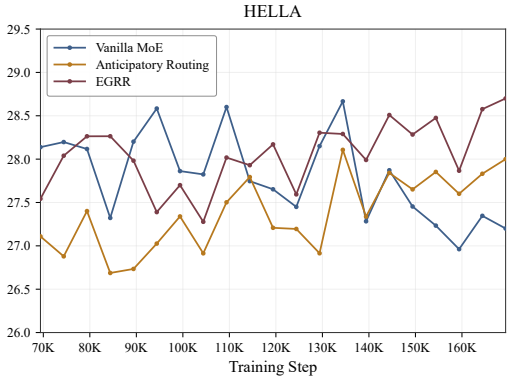
(b) PIQA



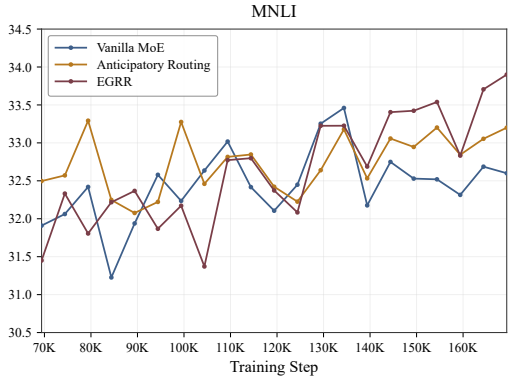
(c) SIQA



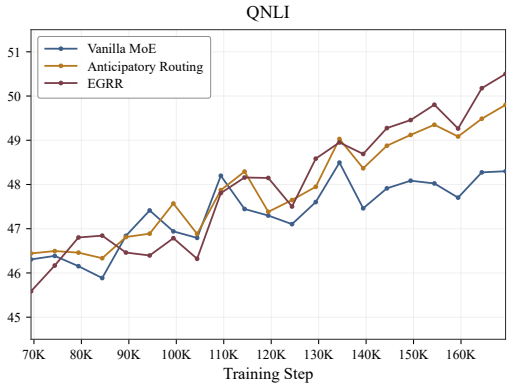
(d) WINO



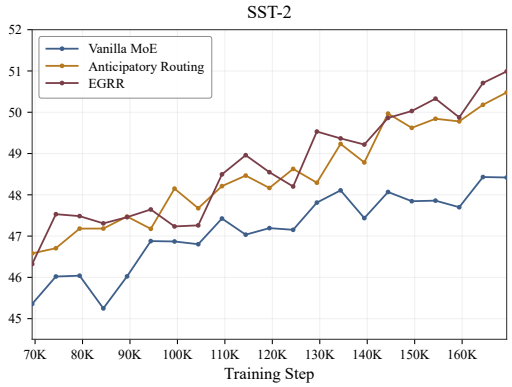
(e) HELLA



(f) MNLI



(g) QNLI



(h) SST-2

Figure 8: Task-specific zero-shot accuracy during pre-training. We save a checkpoint every 5,000 training steps and evaluate each checkpoint on ARC-C, PIQA, SIQA, WINO, HELLA, MNLI, QNLI, and SST-2.