

National Technical University of Athens

School of Electrical and Computer Engineering

Division of Signals, Control and Robotics

Attention-Aware Routing in Mixture of Experts Language Models

Diploma Thesis

Despoina Kallirroi Kosmopoulou

Supervisor: Alexandros Potamianos
Associate Professor, NTUA

Athens, July 2026



National Technical University of Athens

School of Electrical and Computer Engineering

Attention-Aware Routing in Mixture of Experts Language Models

Despoina Kallirroi Kosmopoulou

Supervisor: Alexandros Potamianos
Associate Professor, NTUA

Approved by the examination committee on 6th July 2026.

(Signature)	*(Signature)*	*(Signature)*
.....	.....	.....
Alexandros Potamianos	Athanasios Voulodimos	Athanasios Rontogiannis
Associate Professor, NTUA	Assistant Professor, NTUA	Associate Professor, NTUA

Athens, July 2026

Despoina Kallirroi Kosmopoulou
Diploma in Electrical and Computer Engineering

Copyright © 2026 – All rights reserved.

Despoina Kallirroi Kosmopoulou, 2026.

The copying, storage and distribution of this diploma thesis, in whole or in part, is prohibited for commercial purposes. Reprinting, storage and distribution for non-profit, educational or of a research nature is allowed, provided that the source is indicated and that this message is retained.

The content of this thesis does not necessarily reflect the views of the Department, the Supervisor, or the committee that approved it.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

(Signature)

. . . . . .

Despoina Kallirroi Kosmopoulou

6η Ιουλίου 2026

Στην οικογένειά μου.

Περίληψη

Στα γλωσσικά μοντέλα Μείγματος Ειδικών (Mixture-of-Experts, MoE), ο δρομολογητής επιλέγει και σταθμίζει τους ειδικούς βάσει της κρυφής αναπαράστασης του token, χρησιμοποιώντας περιορισμένη πληροφορία από τα γειτονικά tokens. Προτείνουμε τη **Δρομολόγηση βάσει Προσοχής** (Attention-Aware Routing, AAR), η οποία εμπλουτίζει τον δρομολογητή με χρονικά και φασματικά χαρακτηριστικά που εξάγονται από ένα κυλιόμενο παράθυρο βαρών προσοχής, αποτυπώνοντας την κατάσταση συμφραζομένων του μοντέλου. Διατηρώντας πλήρως παγωμένο τον βασικό μετασχηματιστή (transformer), εκπαιδεύουμε αποκλειστικά τις παραμέτρους δρομολόγησης, απομονώνοντας τη δρομολόγηση ως τη μοναδική μεταβλητή. Η AAR βελτιώνει το GSM8K κατά $+3{,}37$ ποσοστιαίες μονάδες έναντι του SFT παραδοσιακής δρομολόγησης, στο OLMoE. Πέρα από την απόδοση, δείχνουμε ότι η δρομολόγηση και η προσοχή σχηματίζουν ένα συζευγμένο κύκλωμα: οι μεταβολές δρομολόγησης στο επίπεδο l του μετασχηματιστή διαδίδονται και ενισχύουν τα βυθίσματα προσοχής (attention sinks) στο επίπεδο $l{+}1$, αναδιαμορφώνοντας την προσοχή χωρίς καμία άμεση ενημέρωση του μηχανισμού προσοχής. Επιπλέον, η AAR μειώνει την αποκλίνουσα παραγωγή μεγάλων ακολουθιών tokens: οι λανθασμένες απαντήσεις γίνονται συντομότερες, ενώ οι σωστές παραμένουν αμετάβλητες σε μήκος. Αξιοσημείωτα, η επιλεκτική κατά επίπεδο AAR λειτουργεί ως φυσικός ανιχνευτής προσοχής κατά βάθος, αποκαλύπτοντας μια τάση ανταγωνισμού ανάκτησης–συλλογιστικής κατά βάθος. Τα πρώτα επίπεδα αποδεικνύονται κρίσιμα για την ανάκτηση πληροφοριών και ευαίσθητα στις αλλαγές που επάγει η AAR, ενώ τα κέρδη συλλογιστικής εδραιώνονται όταν η AAR εφαρμόζεται στα βαθύτερα επίπεδα.

Τα κέρδη αυτά διατηρούνται όταν προστίθεται μια τυπική βοηθητική συνάρτηση εξισορρόπησης φορτίου (load balancing), αποκλείοντας την άπληστη κατάρρευση των ειδικών ως αιτία τους, και αναπαράγονται σε ένα δεύτερο μοντέλο MoE, το Qwen1.5-MoE-A2.7B, όπου η AAR βελτιώνει το GSM8K κατά $+1{,}88$ ποσοστιαίες μονάδες έναντι του SFT παραδοσιακής δρομολόγησης. Μια συμπληρωματική μελέτη ερμηνευσιμότητας δείχνει ότι η δρομολόγηση επί του ακατέργαστου παραθύρου προσοχής υπερτερεί της δρομολόγησης επί των συνοπτικών στατιστικών του, υποδεικνύοντας ότι η λεπτομερής θεσιακή δομή της προσοχής, και όχι απλώς η κατανομή της, φέρει το σήμα που αξιοποιεί ο δρομολογητής. Συνολικά, τα αποτελέσματα αυτά παρουσιάζουν τη Δρομολόγηση βάσει Προσοχής ως έναν μοχλό δρομολόγησης που βελτιώνει τη μαθηματική συλλογιστική, ενώ ταυτόχρονα αποκαλύπτει μια ερμηνεύσιμη διεπαφή προς τον εσωτερικό υπολογισμό του μοντέλου.

Λέξεις κλειδιά: Μεγάλα Γλωσσικά Μοντέλα, Μείγμα Ειδικών, Δρομολόγηση, Μηχανισμός Προσοχής, Ερμηνευσιμότητα, Μαθηματική Συλλογιστική, Ανάλυση στο Πεδίο της Συχνότητας

Abstract

In Mixture-of-Experts language models, the router typically selects and weights experts based on the token's hidden state, utilizing limited contextual information. We propose **Attention-Aware Routing (AAR)**, which augments the router with temporal and spectral features extracted from a sliding window of attention weights that represent a summary of the model's *contextual* state. Keeping the base transformer entirely frozen, we train only the routing parameters, isolating routing as the sole variable. AAR improves GSM8K by $+3.37$ pp over a routing-only SFT baseline on OLMoE. Beyond performance, we show that routing and attention form a coupled circuit: routing changes at layer l propagate through the residual stream to amplify attention sinks at layer $l+1$, reshaping attention without any direct update to the attention mechanism itself. Further, AAR reduces long diverging generation, with incorrect answers getting shorter, while correct answers remain unchanged in length. Notably, layer-selective AAR serves as a natural depth-wise attention probe, revealing a retrieval-reasoning tension across depth. Early layers prove critical for retrieval and sensitive to AAR-driven changes, while reasoning gains solidify when AAR is applied in deeper layers.

These gains persist when a standard auxiliary load-balancing loss is added, ruling out greedy expert collapse as their source, and replicate on a second MoE model, Qwen1.5-MoE-A2.7B, where AAR improves GSM8K by $+1.88$ pp over the routing-only baseline. A complementary interpretability ablation shows that routing on the raw attention window outperforms routing on its summary statistics, indicating that the fine-grained positional structure of attention, not merely its distribution, carries the signal the router exploits. Taken together, these results frame Attention-Aware Routing as a routing lever that improves mathematical reasoning while exposing an interpretable interface into a model's internal computation.

Keywords: Large Language Models, Mixture of Experts, Routing, Attention Mechanisms, Interpretability, Mathematical Reasoning, Frequency-Domain Analysis, Layer-Selective Routing

Ευχαριστίες

Ευχαριστώ θερμά τον επιβλέποντα καθηγητή μου, Αλέξανδρο Ποταμιάνο, για την καθοδήγηση, την υποστήριξη και τις πολύτιμες συζητήσεις μας καθ'όλη τη διάρκεια της εκπόνησης αυτής της διπλωματικής εργασίας, αλλά και συνολικά τον χρόνο που πέρασα στο εργαστήριο του Speech and Language Processing group. Είχα την ευκαιρία να εκτεθώ σε πολλές και διαφορετικές πτυχές της έρευνας στον τομέα της μηχανικής μάθησης και της επεξεργασίας φυσικής γλώσσας, να πάρω πρωτοβουλίες και να αναπτύξω την ερευνητική μου σκέψη και ανεξαρτησία, εμπειρία που διαμόρφωσε άμεσα και τα επόμενα βήματά μου. Επίσης, θα ήθελα να ευχαριστήσω τον μετα-διδακτορικό ερευνητή Ευθύμη Γεωργίου, με τον οποίο συνεργαστήκαμε στενά σε πολλαπλά projects εντός και εκτός του πλαισίου αυτής της εργασίας, για την ανταλλαγή ιδεών, την καθοδήγηση και τις συμβουλές του. Ακόμη, ευχαριστώ τον Γιάννη Καραμανωλάκη και τον Swastik Roy για τις συμβουλές και τις εύστοχες παρατηρήσεις τους κατά την εκπόνηση αυτής της εργασίας, και τον Τάσο Τσέτσιλα για την συνεργασία στα πρώτα στάδια της δουλειάς αυτής.

Ευρύτερα, ευχαριστώ τους καθηγητές των τομέων Σημάτων, Ελέγχου και Ρομποτικής και Πληροφορικής, για τα εναύσματα που μου έδωσαν να ασχοληθώ με αυτόν τον συναρπαστικό κόσμο κατά τη διάρκεια των σπουδών μου.

Αναπόσπαστο κομμάτι των σπουδών μου ήταν οι συζητήσεις με συμφοιτητές και φίλους, οι οποίες έδωσαν χρώμα και ζωή σε αυτό το ταξίδι. Τους ευχαριστώ βαθιά. Τέλος, ευχαριστώ την οικογένειά μου για την υποστήριξη και την αγάπη τους, που με συνόδευσαν σε όλη τη διάρκεια των σπουδών μου.

Δέσποινα Καλλιρρόη Κοσμοπούλου
Αθήνα, Ιούλιος 2026

Contents

Περίληψη 6

Abstract 8

Ευχαριστίες 10

Εκτεταμένη Ελληνική Περίληψη 19

1 Introduction 31
- 1.1 The Rise of Large Language Models . . . 31
- 1.2 Mixture-of-Experts as Conditional Computation . . . 31
- 1.3 The Routing Bottleneck . . . 32
- 1.4 Thesis Motivation and Contribution . . . 32
- 1.5 Thesis Outline . . . 33

2 Language Models and Mixture of Experts 35
- 2.1 Neural Language Modeling . . . 35
  - 2.1.1 Language Modeling as Next-Token Prediction . . . 35
  - 2.1.2 Training with Gradient Descent . . . 36
  - 2.1.3 Pre-training and Fine-tuning . . . 36
  - 2.1.4 Scale, In-Context Learning, and Reasoning . . . 36
- 2.2 The Transformer Architecture . . . 37
  - 2.2.1 Tokenization and Embeddings . . . 37
  - 2.2.2 Self-Attention . . . 38
  - 2.2.3 Feed-Forward Sublayer . . . 38
  - 2.2.4 Positional Encodings . . . 39
- 2.3 Mixture-of-Experts Language Models . . . 40
  - 2.3.1 The Expert Layer . . . 40
  - 2.3.2 Token Routing . . . 40
  - 2.3.3 Load Balancing . . . 41
  - 2.3.4 Key MoE Models . . . 41

3 Attention-Aware Routing 43
- 3.1 Introduction . . . 43
- 3.2 Related Work . . . 43

3.3 Method . . . 46
3.3.1 Standard MoE Routing . . . 46
3.3.2 AAR-T: Time-Domain Attention Router . . . 46
3.3.3 AAR-TF: Frequency-Domain Extension . . . 46
3.3.4 Router Interpolation . . . 47
3.3.5 Layer-Selective AAR . . . 48
3.4 Experiments . . . 48
3.4.1 OLMoE . . . 48
3.4.2 Training Configuration . . . 48
3.4.3 Baselines . . . 49
3.4.4 Evaluation . . . 49
3.4.5 Main Results . . . 49
3.4.6 Window-Size Ablation: AAR-T Applied to All Layers . . . 49
3.5 Window Size and Signal Structure . . . 51
3.5.1 Intrinsic Correlation Length of the Attention Signal . . . 51
3.5.2 Window Size . . . 51
3.6 From Routing to Generation: Internal and External Effects . . . 51
3.6.1 Internal: AAR Reshapes Attention . . . 51
3.6.2 External: AAR Reduces Long Diverging Generation . . . 54
3.7 Complementary Experiments and Ablations . . . 54
3.7.1 AAR and Load Balancing . . . 54
3.7.2 Routing with Attention Statistics . . . 55
3.8 Summary of Results . . . 56

4 Layer-Selective AAR as a Depth-wise Attention Probe **57**
4.1 Motivation . . . 57
4.2 Overview . . . 57
4.3 Experiments . . . 58
4.3.1 MMLU . . . 58
4.3.2 ARC-Challenge and ARC-Easy . . . 60
4.4 Discussion . . . 62

5 Discussion **65**

6 Closing Remarks **67**
6.1 Conclusion . . . 67
6.2 Limitations . . . 67
6.3 Future Work . . . 68

References **69**

A Evaluation Details **75**
A.1 GSM8K . . . 75
A.2 HumanEval . . . 75
A.3 BBH . . . 76
A.4 IFEval . . . 76
A.5 MMLU . . . 76
A.6 ARC . . . 77

B Training with Different Data **79**
B.1 CodeFeedBack . . . 79
B.2 MetaMath . . . 79

C Frequency-Domain Analysis of Attention Routing **83**
C.1 Shift-Invariance and the Case for Spectral Features . . . 83
C.2 Layer-wise Spectral Structure . . . 84

D Future Work: AAR and Masked Diffusion Language Models **87**

List of Figures

2.1 The Transformer encoder–decoder architecture as introduced by Vaswani et al. [1]. The models in this thesis use only the decoder stack (right half), which applies masked self-attention followed by a position-wise feed-forward sublayer at each layer. 37

2.2 Attention mechanisms from Vaswani et al. [1]. **(a)** Scaled Dot-Product Attention computes query–key compatibilities, scales by $1/\sqrt{d_k}$, applies softmax, and weights the values. **(b)** Multi-Head Attention projects queries, keys, and values into H lower-dimensional subspaces, runs Scaled Dot-Product Attention in each, and concatenates the results. 39

3.1 **PCA of Attention Window Fingerprints**. We visualize the first two principal components of the attention windows (only including the 128-sized attention weight vectors, averaged across heads) for different MMLU subjects. Each point represents the average attention weight vector of the final question's tokens in each 5-shot prompt. At Layer 0, attention patterns exhibit strong subject-specific clustering. This clustering becomes more entangled by Layer 7, showcasing the progression of the internal processing. Notice how semantically similar subjects get relatively closer, compared to dissimilar subjects. 44

3.2 Normalized autocorrelation function (ACF) of the attention window signal $w_i^l(t)$ at layers 4, 9, and 12, for example window sizes $\in \{20, 64, 128\}$. The ACF is averaged over query tokens and prompts from GSM8K (5-shot). The dashed vertical line marks lag $= 20$. Across all layers and window sizes, the ACF has decayed below 0.1 by lag ~ 20, revealing an intrinsic correlation length of 20 tokens in the attention signal. A window of $M = 20$ is thus sufficient to capture most structured information. 52

3.3 Top: Absolute weight to the attention sink, per layer (averaged across prompt's tokens). Bottom: the absolute difference, isolated. The different routing decisions of layer 9, because of AAR, lead to different attention weights produced in the attention submodule, at layer 10. 53

4.1 MMLU accuracy (%) versus attention routing start layer (mean ± SEM over 5 seeds). The dashed red line indicates the routing-only SFT baseline (53.92%). Performance recovers and plateaus near baseline for $l_s \geq 7$, with a slight peak around $l_s = 9$–11. 59
4.2 AAR-T ($M = 20$) accuracy across MMLU subjects as a function of starting layer. Most subjects' accuracy curves follow a bell shape. Retrieval-heavy subjects (Biology, History, Philosophy) are worst when AAR starts early and recover as the starting layer increases; reasoning-heavy subjects improve regardless of starting depth; subjects combining both (Economics, Physics) peak around layers 7–11. 60
4.3 ARC-Challenge accuracy (0-shot, with chat template, log-likelihood) versus start layer. The blue line shows raw accuracy; the orange line shows normalized accuracy. 61
4.4 ARC-Easy accuracy (0-shot, with chat template, log-likelihood) versus start layer. 61

C.1 Layer-wise spectral flatness and Shannon entropy of attention windows in OLMoE. Spectral flatness follows a U-shape: middle layers (∼4–10) exhibit the least flat, most structured frequency content, while the earliest and deepest layers are flatter. The dashed red line (entropy, right axis) does not follow the exact inverse, indicating complementary information. 85
C.2 Layer-wise spectral flatness and Shannon entropy for Qwen1.5-MoE. The U-shape in spectral flatness is more pronounced than in OLMoE, reinforcing the model-agnostic nature of the finding. 85

List of Tables

3.1 Training hyperparameters. 49

3.2 AAR compared against trained baselines. GSM8K consistently improves. AAR-TF achieves best results. $S = \{9, \ldots, 15\}$ for OLMoE; $S_Q = \{14, \ldots, 17\}$ for Qwen1.5. Best in bold. 50

3.3 AAR-T (all layers) results (% accuracy) on reasoning benchmarks (averaged over 5 seeds, mean $\pm$ std). ARC-C = ARC-Challenge, ARC-E = ARC-Easy (0-shot, without a chat template). 50

3.4 AAR-T (all layers) results (% accuracy) on coding, general knowledge and instruction-following benchmarks (averaged over 5 seeds, mean $\pm$ std). For HumanEval we report the pass@10 % accuracy. 50

3.5 Length analysis (characters) on GSM8K averaged across seeds and context sizes M. Ramble Rate denotes responses exceeding 1,000 characters. AAR-T leaves correct responses nearly unchanged while reducing the length of incorrect reasoning, especially in the tail (P99). The answer length reduction also persists for answers that get corrected by AAR-T (Wrong $\rightarrow$ Correct row). 54

3.6 Effect of auxiliary load-balancing loss (weight $\lambda = 0.01$) on AAR-T applied to all OLMoE layers. GSM8K (8-shot CoT), HumanEval (pass@10), and MMLU (5-shot). Bold marks the best AAR variant per column. . . . 55

3.7 AAR-T with raw attention window vs. routing conditioned on attention statistics (mean, 2nd/3rd moment, Gini index, KL divergence from uniform, recency mass, centroid position), applied to all OLMoE layers. GSM8K (8-shot CoT), HumanEval (pass@10), MMLU (5-shot). Bold marks the best AAR variant per column. 55

4.1 MMLU subcategory Δ (absolute % accuracy difference from baseline, 5 seeds) when AAR-T is applied to all layers of OLMoE. Reasoning-oriented subjects (top) are improved; degradation is concentrated in retrieval-heavy subjects (bottom). 59

B.1 AAR-T (raw attention window, all layers) trained on CodeFeedBack. GSM8K (8-shot CoT), HumanEval (pass@10), MMLU (5-shot). Bold marks the best value per column. 80

B.2 Attention-statistics router (mean, 2nd/3rd moment, Gini index, KL divergence from uniform, recency mass, centroid position; all layers) trained on CodeFeedBack. Bold marks the best value per column. 80
B.3 AAR-T trained on MetaMath only. GSM8K (8-shot CoT), MMLU (5-shot). AAR yields a large GSM8K gain ($+2.54$ pp) but at the cost of a severe MMLU drop (-4.76 pp) relative to the MetaMath baseline, consistent with catastrophic forgetting on non-math tasks. The GSM8K gains are over the severely degraded performance of the SFT baseline on GSM8K. Bold marks the best value per column. 81

Εκτεταμένη Ελληνική Περίληψη

1. Εισαγωγή

Τα τελευταία χρόνια, τα γλωσσικά μοντέλα μεγάλης κλίμακας (Large Language Models, LLMs) έχουν μεταμορφώσει την επεξεργασία φυσικής γλώσσας. Οδηγούμενα από την κλιμάκωση σε δισεκατομμύρια παραμέτρους και εκπαιδευμένα σε δεδομένα κλίμακας Διαδικτύου, τα σύγχρονα συστήματα επιδεικνύουν ικανότητες συλλογιστικής πολλών βημάτων, σύνθεσης κώδικα, ακολούθησης οδηγιών και ανοιχτού διαλόγου που ήταν ανέφικτες λίγα χρόνια πριν. Στο επίκεντρο αυτής της προόδου βρίσκεται η αρχιτεκτονική Μετασχηματιστή (Transformer), η οποία επεξεργάζεται ακολουθίες tokens μέσω μιας στοίβας επιπέδων, ανανεώνοντας σε κάθε επίπεδο την αναπαράσταση κάθε token με βάση τα προηγούμενα και μετασχηματίζοντάς την μέσω ενός δικτύου τροφοδότησης προς τα εμπρός (Feed-Forward Network, FFN).

Η ισχυρή τους απόδοση συνοδεύεται από κόστος σε επίπεδο συστημάτων: σε έναν πυκνό Μετασχηματιστή, *κάθε* παράμετρος εφαρμόζεται σε *κάθε* token σε *κάθε* επίπεδο, με αποτέλεσμα το υπολογιστικό κόστος να αυξάνεται ανάλογα με τον αριθμό παραμέτρων. Η αρχιτεκτονική Μείγματος Ειδικών (Mixture-of-Experts, MoE) λύνει αυτό το πρόβλημα μέσω του *κατά συνθήκη υπολογισμού*: αντί για ένα ενιαίο FFN, κάθε επίπεδο MoE διαθέτει E ανεξάρτητους ειδικούς (experts) και έναν εκπαιδευμένο δρομολογητή που ενεργοποιεί μόνο $K \ll E$ από αυτούς για κάθε token. Η συνολική χωρητικότητα κλιμακώνεται με τον αριθμό των ειδικών E σε κάθε επίπεδο, ενώ ο αριθμός ενεργών παραμέτρων ανά εκτέλεση παραμένει σταθερός — επιτυγχάνοντας μεγαλύτερα μοντέλα με ουσιαστικά σταθερό κόστος εκτέλεσης.

Δρομολόγηση. Ο τυπικός δρομολογητής MoE λειτουργεί από περιορισμένη οπτική γωνία. Για ένα token στο επίπεδο l, υπολογίζει μια γραμμική προβολή της *κρυφής κατάστασης μετά την αυτο-προσοχή* $\mathbf{h}_i^l$ και επιλέγει τους ειδικούς στους οποίους αποδίδει τις υψηλότερες βαθμολογίες. Η κρυφή κατάσταση φέρει πληροφορία συγκειμένου, αλλά μόνο *έμμεσα*: καταγράφει πού έχει καταλήξει η αναπαράσταση του token, αλλά όχι πώς διαμορφώθηκε ή ρητά πώς σχετίζεται με τα γύρω tokens.

Παράλληλα, το ίδιο επίπεδο έχει ήδη παράγει — ως αναπόφευκτο υποπροϊόν της αυτο-προσοχής — ένα σχεσιακό σήμα: τα *βάρη προσοχής* $w_i^l(t)$, τα οποία κωδικοποιούν ποια tokens κρίθηκαν σχετικά και σε ποιον βαθμό. Αυτά τα βάρη χρησιμοποιούνται για τον υπολογισμό της εξόδου προσοχής και στη συνέχεια *απορρίπτονται*. Ο δρομολογητής βλέπει την *έξοδο* της διαδικασίας προσοχής, αλλά όχι τη διαδικασία καθ' εαυτή. Η

σχεσιακή δομή που η προσοχή κάνει ρητή είναι διαθέσιμη στο επίπεδο, αλλά αναξιοποίητη κατά τη δρομολόγηση. Αυτό αποτελεί έναν περιορισμό στη δρομολόγηση, τον οποίο διερευνά η παρούσα εργασία.

Ερευνητικό Ερώτημα και προτεινόμενη λύση. Η εργασία έχει σαν στόχο να απαντήσει ένα απλό ερώτημα: *τι συμβαίνει εάν δώσουμε στον δρομολογητή πρόσβαση σε πληροφορία συγκειμένου;*

Προτείνουμε τη **Δρομολόγηση βάσει Προσοχής** (Attention-Aware Routing, AAR), η οποία εμπλουτίζει τον δρομολογητή MoE με ένα κυλιόμενο παράθυρο πρόσφατων βαρών προσοχής ως πρόσθετο σήμα εισόδου. Για να αποδοθεί κάθε παρατηρούμενη μεταβολή αποκλειστικά στη δρομολόγηση, το σύνολο των παραμέτρων του βασικού μετασχηματιστή διατηρείται *παγωμένο* — μόνο οι παράμετροι δρομολόγησης εκπαιδεύονται, απομονώνοντας τη δρομολόγηση ως τη μοναδική ανεξάρτητη μεταβλητή.

Κύριες συνεισφορές.

1. **Βελτίωση μαθηματικής συλλογιστικής**. Η δρομολόγηση βάσει προσοχής (AAR) βελτιώνει το GSM8K κατά έως $+3{,}37$ ποσοστιαίες μονάδες (AAR-TF) έναντι βασικής μεθόδου SFT με παραδοσιακή δρομολόγηση στο OLMoE, ενώ η απόδοση σε άλλα benchmarks διατηρείται.

2. **Σύζευξη δρομολόγησης–προσοχής**. Δείχνουμε εμπειρικά ότι οι αλλαγές δρομολόγησης που επάγει η AAR διαδίδονται μέσω των υπολειμματικών συνδέσεων του δικτύου, ενισχύοντας τα βυθίσματα προσοχής (attention sinks) στο επόμενο επίπεδο — χωρίς καμία άμεση ενημέρωση του μηχανισμού προσοχής.

3. **Ανιχνευτής λειτουργικής εξειδίκευσης κατά βάθος**. Η επιλεκτική κατά επίπεδο AAR αποκαλύπτει διαφορετικούς λειτουργικούς ρόλους ανά βάθος: τα πρώτα επίπεδα είναι σημαντικά την ανάκτηση πληροφορίας και είναι ευαίσθητα σε αλλαγές δρομολόγησης, ενώ τα βαθύτερα επίπεδα εδραιώνουν τα κεδη στη συλλογιστικής.

2. Θεωρητικό Υπόβαθρο

Νευρωνική Γλωσσική Μοντελοποίηση

Η νευρωνική γλωσσική μοντελοποίηση διατυπώνεται ως πρόβλημα πρόβλεψης επόμενου token: δοθέντος ενός προθέματος $x_{<t}$, το μοντέλο ελαχιστοποιεί την αρνητική λογαριθμική πιθανοφάνεια $\mathcal{L}(\theta) = -\sum_t \log P(x_t \mid x_{<t}; \theta)$. Ο στόχος αυτός δεν απαιτεί ανθρώπινες επισημειώσεις: κάθε κείμενο παρέχει από μόνο του τα δεδομένα επίβλεψης, επιτρέποντας προ-εκπαίδευση σε δεδομένα κλίμακας Διαδικτύου. Κατά την προ-

σαρμογή (fine-tuning), ο ίδιος στόχος εφαρμόζεται σε επισημειωμένα ζεύγη εντολής–απάντησης (επιβλεπόμενη προσαρμογή, SFT), εκπαιδεύοντας το μοντέλο να ακολουθεί οδηγίες και να παράγει χρήσιμες απαντήσεις.

Αρχιτεκτονική Μετασχηματιστή

Ο Μετασχηματιστής μόνο-με-αποκωδικοποιητές επεξεργάζεται ακολουθίες tokens μέσω μίας στοίβας L ταυτόσημων επιπέδων. Κάθε επίπεδο συνδυάζει τρία στοιχεία:

- **Αυτο-προσοχή (Self-Attention)**. Κάθε token i συσσωρεύει πληροφορία από τα προηγούμενα tokens με βάρη $w_i^l(t) = \text{softmax}_t\big(\mathbf{q}_i^\top \mathbf{k}_t / \sqrt{d_k}\big)$. Η μήτρα προσοχής $A^l \in \mathbb{R}^{n\times n}$ αποτυπώνει άμεσα ποιο πλαίσιο κρίνει το μοντέλο σχετικό για κάθε token σε κάθε επίπεδο, και αποτελεί το κεντρικό σήμα που αξιοποιεί η AAR.

- **Δίκτυο Τροφοδότησης Εμπρός (FFN)**. Εφαρμόζεται ανεξάρτητα σε κάθε θέση μέσω διεπεκτεινόμενου MLP. Το FFN περιέχει τη συντριπτική πλειονότητα των παραμέτρων και αποτελεί το υποσύστημα που τα μοντέλα MoE αντικαθιστούν με ομάδα ειδικών.

- **Κωδικοποιήσεις Θέσης (RoPE)**. Εισάγουν σχετική πληροφορία θέσης περιστρέφοντας τα διανύσματα ερωτήματος και κλειδιού ανάλογα με τη σχετική τους θέση, ώστε τα αποτελέσματα της αυτο-προσοχής να εξαρτώνται μόνο από τη σχετική απόσταση $i - t$.

Μοντέλα Μείγματος Ειδικών

Τα μοντέλα MoE αντικαθιστούν το FFN κάθε επιπέδου με E ανεξάρτητους ειδικούς $\{F_1, \ldots, F_E\}$ και έναν εκπαιδευμένο δρομολογητή. Το τυπικό σχήμα top-K επιλέγει τους K ειδικούς με τις υψηλότερες βαθμολογίες που προκύπτουν από γραμμική προβολή της κρυφής κατάστασης: $\mathbf{g} = W_r \mathbf{h}_i^l$. Η έξοδος του επιπέδου MoE είναι το σταθμισμένο άθροισμα των επιλεγμένων ειδικών, με τους υπόλοιπους να παραλείπονται. Για την αντιμετώπιση της τάσης κατάρρευσης σε λίγους ειδικούς, εισάγεται βοηθητική συνάρτηση κόστους για εξισορρόπηση φορτίου κατά την προ-εκπαίδευση.

Τα κύρια μοντέλα της εργασίας είναι το **OLMoE** (64 ειδικοί ανά επίπεδο, top-8 δρομολόγηση, 1B ενεργές / 7B συνολικές παράμετροι, 16 επίπεδα) — ένα πλήρως ανοιχτό μοντέλο που είναι ιδανικό για ανάλυση δρομολόγησης — και το **Qwen1.5-MoE** (2,7B ενεργές / 14,3B συνολικές παράμετροι, 24 επίπεδα), που χρησιμοποιείται για επαλήθευση.

3. Δρομολόγηση βάσει Προσοχής

Μέθοδος

Τυπική δρομολόγηση MoE. Στην τυπική δρομολόγηση, τα logits επιλογής ειδικών για το token i στο επίπεδο l υπολογίζονται ως $\mathbf{g}_{\text{std}} = W_r^l \mathbf{h}_i^l$, με $W_r^l \in \mathbb{R}^{E \times d}$. Οι top-$K$ ειδικοί επιλέγονται μετά από softmax και το τελικό hidden state είναι ένα σταθμισμένο άθροισμα των εξόδων τους.

AAR-T: Χρονικός Δρομολογητής Προσοχής. Ορίζουμε το *διάνυσμα βαρών προσοχής* $\mathbf{w}_i^l \in \mathbb{R}^M$ για το token i στο επίπεδο l ως τα βάρη προσοχής — μέσος όρος κεφαλών — σε ένα κυλιόμενο παράθυρο των τελευταίων M θέσεων, με συμπλήρωση με μηδενικά (zero padding) αριστερά για $i < M$. Ο βοηθητικός χρονικός δρομολογητής είναι μία απλή γραμμική προβολή:

$$\mathbf{g}_{\text{attn}} = A_l^\top \mathbf{w}_i^l, \quad A_l \in \mathbb{R}^{M \times E}.$$

Το σήμα $\mathbf{w}_i^l$ αντικατοπτρίζει τη δομή της προσοχής του μοντέλου: ποιες θέσεις λαμβάνουν προσοχή, ποιες αγνοούνται, πού συγκεντρώνεται η εστίαση. Υπολογίζεται ήδη ως μέρος της κανονικής εκτέλεσης και στη συνέχεια απορρίπτεται — η AAR απλώς το ανακατευθύνει στον δρομολογητή.

AAR-TF: Επέκταση στο πεδίο της Συχνότητας. Το παράθυρο $\mathbf{w}_i^l$ μπορεί να θεωρηθεί ως διακριτό σήμα χρόνου. Ο Διακριτός Μετασχηματισμός Fourier (DFT) του σήματος αυτού παράγει ένα συμπληρωματικό φασματικό διάνυσμα με δύο χρήσιμες ιδιότητες: (1) *αναλλοίωτο ως προς μετατοπίσεις*, ώστε πρότυπα προσοχής ίδιου σχήματος αλλά διαφορετικής απόλυτης θέσης εντός του παραθύρου να αντιστοιχούν σε παρόμοιες εισόδους δρομολόγησης, και (2) *μη-γραμμικός μετασχηματισμός*, που επιτρέπει στον γραμμικό φασματικό δρομολογητή να προσπελαύνει χαρακτηριστικά που θα απαιτούσαν διαφορετικά βαθύτερη επεξεργασία. Η φασματική έξοδος $\mathbf{g}_{\text{fft}} = \tilde{A}_l^\top \tilde{\mathbf{w}}_i^l$ υπολογίζεται από τα μέτρα SFT μετά από λογαριθμική συμπίεση δυναμικής εύρους, διατηρώντας μόνο τις $K = \lfloor M/2 \rfloor + 1$ μη-πλεονάζουσες συνιστώσες.

Παρεμβολή Δρομολογητή. Κάθε βοηθητικός δρομολογητής συνοδεύεται από μία πύλη εμπιστοσύνης (confidence gate) εξαρτώμενη τόσο από την κρυφή κατάσταση όσο και από το αντίστοιχο σήμα συγκειμένου:

$$\boldsymbol{\alpha}_1 = \sigma\left(G_l^{(1)}\left[\mathbf{h}_i^l \,\|\, \mathbf{w}_i^l\right]\right), \qquad \boldsymbol{\alpha}_2 = \sigma\left(G_l^{(2)}\left[\mathbf{h}_i^l \,\|\, \tilde{\mathbf{w}}_i^l\right]\right).$$

Τα τελικά logits δρομολόγησης ορίζονται ως:

$$\mathbf{g}_i = \beta_0 \odot \mathbf{g}_{\text{std}} + \beta_1 \odot \mathbf{g}_{\text{attn}} + \beta_2 \odot \mathbf{g}_{\text{fft}},$$

με $\beta_0 + \beta_1 + \beta_2 = \mathbf{1}$ και $\beta_1, \beta_2 \leq \frac{1}{2}$, ώστε ο τυπικός δρομολογητής να διατηρεί το μεγαλύτερο βάρος. Η διατύπωση ανακτά την AAR-T όταν $\boldsymbol{\alpha}_2 \equiv \mathbf{0}$ και την τυπική δρομολόγηση MoE όταν $\boldsymbol{\alpha}_1 \equiv \boldsymbol{\alpha}_2 \equiv \mathbf{0}$.

Πειραματική Διαδικασία

Τα κύρια πειράματα πραγματοποιούνται στο **OLMoE-1B-7B-0125-SFT**. Το πρωτόκολλο εκπαίδευσης κρατά παγωμένο το σύνολο των παραμέτρων του μοντέλου εκτός από τους αρχικούς δρομολογητές και τους νέους δρομολογητές προσοχής. Η εκπαίδευση γίνεται με AdamW (ρυθμός μάθησης 10^{-4}, γραμμική πτώση) επί 25% του συνόλου Tulu-3 SFT ($\sim$220K παραδείγματα, με μέγιστο μήκος ακολουθίας 512). Τα αποτελέσματα παρουσιάζονται ως μέσος όρος 5 τυχαίων αρχικοποιήσεων.

Τα benchmarks αξιολόγησης καλύπτουν: GSM8K (μαθηματική συλλογιστική), MMLU (γενική γνώση), HumanEval (κώδικας), IFEval (ακολούθηση οδηγιών), BBH (σύνθετη συλλογιστική), ARC-Challenge και ARC-Easy. Ως βασικές μέθοδοι για σύγκριση χρησιμοποιούνται το αρχικό OLMoE-0125-SFT και πανομοιότυπες εκτελέσεις SFT αποκλειστικά παραδοσιακής δρομολόγησης (χωρίς βοηθητικούς δρομολογητές), απομονώνοντας την επίδραση του σήματος προσοχής από αυτή της συνεχούς εκπαίδευσης.

Κύρια Αποτελέσματα

Η καλύτερη διαμόρφωση — AAR-TF εφαρμοσμένη στα επίπεδα $S = \{9, \ldots, 15\}$ με παράθυρο $M = 40$ — επιτυγχάνει GSM8K $55{,}36 \pm 0{,}74$, βελτίωση $+3{,}37$ ποσοστιαίων μονάδων έναντι της μεθόδου SFT με παραδοσιακή δρομολόγηση ($51{,}99$). Η AAR-T με $M = 20$ βελτιώνει ήδη το GSM8K κατά $+2{,}58$ μονάδες. Τα αποτελέσματα σε BBH και IFEval διατηρούνται εντός στατιστικού θορύβου, ενώ το HumanEval μειώνεται ελαφρά ($\sim$1 μονάδα) στην καλύτερη διαμόρφωση. Τα κέρδη GSM8K επαληθεύονται επίσης στο **Qwen1.5-MoE** ($+1{,}88$ ποσοστιαίες μονάδες), επιβεβαιώνοντας τη γενίκευση της μεθόδου σε διαφορετικές αρχιτεκτονικές MoE.

Μέγεθος Παραθύρου και Δομή Σήματος

Η ανάλυση της κανονικοποιημένης συνάρτησης αυτοσυσχέτισης (ACF) του σήματος βαρών προσοχής $w_i^l(t)$ αποκαλύπτει ότι, σε όλα τα επίπεδα και μεγέθη παραθύρου, η ACF μηδενίζεται περί το $\tau \approx 20$. Αυτό υποδεικνύει ένα *εγγενές μήκος συσχέτισης* 20 tokens στο σήμα προσοχής: ένα παράθυρο $M = 20$ επαρκεί για την καταγραφή της πλειοψηφίας της γραμμικά δομημένης πληροφορίας. Πράγματι, η μέθοδος AAR-

Τ έχει πολύ καλή επίδοση $M = 20$ στη διαμόρφωση με όλα τα επίπεδα. Η AAR-TF μετατοπίζει αυτή τη βέλτιστη τιμή στο $M = 40$, καθώς ο αριθμός φασματικών συνιστωσών $K = \lfloor M/2 \rfloor + 1$ αυξάνεται με M, ωφελώντας από λεπτότερη φασματική ανάλυση.

Εσωτερικές Επιδράσεις: Αναδιαμόρφωση της Προσοχής

Για να αποκαλυφθεί ο υπολογιστικός μηχανισμός πίσω από τα κέρδη συλλογιστικής, εξετάζουμε αν και πώς διαφέρουν τα βάρη προσοχής μεταξύ των μοντέλων της μεθόδου AAR-T και των μεθόδων σύγκρισης κατά την επεξεργασία GSM8K prompts. Διαπιστώνουμε ότι η AAR-T *ενισχύει τα βυθίσματα προσοχής* (attention sinks), δηλαδή τη συγκέντρωση της μάζας προσοχής, στο πρώτο token της ακολουθίας. Η σύνδεση αυτή τεκμηριώνεται μέσω της επιλεκτικής κατά επίπεδο παραλλαγής: όταν η AAR εφαρμόζεται από το επίπεδο l και έπειτα, η ενίσχυση των βυθισμάτων προσοχής εμφανίζεται ακριβώς στο επίπεδο $l{+}1$.

Ο υποκείμενος μηχανισμός είναι ο εξής: οι αλλαγές δρομολόγησης στο επίπεδο l τροποποιούν την επιλογή ειδικών και τους συντελεστές σύνθεσης, αναδιαμορφώνοντας τις αναπαραστάσεις που τροφοδοτούνται στο υποσύστημα αυτο-προσοχής του επιπέδου $l{+}1$ μέσω των υπολειμματικών συνδέσεων — **χωρίς καμία άμεση ενημέρωση του μηχανισμού προσοχής**. Η δρομολόγηση ενεργεί άρα όχι μόνο ως επιλογέας εμπειρογνωμοσύνης, αλλά ως *έμμεσος ρυθμιστής* της επεξεργασίας προσοχής στα επόμενα επίπεδα. Η ενίσχυση βυθισμάτων προσοχής σταθεροποιεί την αποκωδικοποίηση αγκυρώνοντας τον υπολογισμό σε ένα διακριτό αναφοράς σταθερής θέσης — μηχανισμός συνεπής με βελτιωμένη μαθηματική συλλογιστική.

Εξωτερικές Επιδράσεις: Μείωση Αποκλίνουσας Παραγωγής

Συγκρίνοντας τα μήκη απαντήσεων στο GSM8K μεταξύ ζευγών AAR-T και βασικής γραμμής, προκύπτει εξαιρετικά εκλεκτική επίδραση:

- Όταν αμφότερα τα μοντέλα απαντούν *λανθασμένα*, η AAR-T παράγει σημαντικά συντομότερες απαντήσεις: μέσος όρος $-11{,}8\%$, P99 $-14{,}9\%$, ποσοστό «φλυαρίας» (>1000 χαρακτήρες) $-23{,}9\%$.
- Όταν αμφότερα τα μοντέλα απαντούν *σωστά*, τα μήκη είναι ουσιαστικά αδιάκριτα ($-0{,}8\%$ στον μέσο όρο).

Η επίδραση εντοπίζεται στο καθεστώς λανθασμένης απάντησης και είναι συνεπής σε ολόκληρη την κατανομή μήκους παραγωγής, υποδηλώνοντας ότι η AAR μειώνει την *αποκλίνουσα παραγωγή*: το μοντέλο δεσμεύεται νωρίτερα σε απάντηση όταν κάνει λάθος, αντί να παράγει μακρές λανθασμένες αλυσίδες συλλογισμού. Οι σωστές απαντήσεις δεν επηρεάζονται ουσιαστικά, επιβεβαιώνοντας ότι το μοντέλο δεν έχει απλώς μάθει να είναι ομοιόμορφα συντομότερο.

Συμπληρωματικά Πειράματα

Εξισορρόπηση φορτίου. Εφαρμόζοντας βοηθητική συνάρτηση κόστους για εξισορρόπηση φορτίου ($\lambda = 0{,}01$) κατά την εκπαίδευση, η AAR-T συνεχίζει να βελτιώνει το GSM8K σε όλα τα μεγέθη παραθύρου. Τα κέρδη δεν αποδίδονται λοιπόν σε κατάρρευση σε ελάχιστους ειδικούς, αλλά αντικατοπτρίζουν γνήσια ικανότητα αξιοποίησης του σήματος προσοχής.

Στατιστικά προσοχής έναντι πλήρους παραθύρου. Αντικαθιστώντας το παράθυρο προσοχής με ένα διάνυσμα στατιστικών χαρακτηριστικών (μέση τιμή, 2η και 3η ροπή, δείκτης Gini, απόκλιση KL από ομοιόμορφη κατανομή, μάζα πρόσφατων tokens, μέση θέση κεντροειδούς), ο στατιστικός δρομολογητής επιτυγχάνει κάποια βελτίωση ($+1{,}70$ ποσοστιαίες μονάδες GSM8K για $M = 20$) αλλά δεν φτάνει στο πλήρες επίπεδο της AAR-T. Αυτό δείχνει ότι, ενώ ο δρομολογητής πράγματι εκμεταλλεύεται τα χαρακτηριστικά της κατανομής, η λεπτομερής δομή θέσης των βαρών προσοχής φέρει επίσης σημαντική πρόσθετη πληροφορία.

4. Επιλεκτική Δρομολόγηση κατά Επίπεδο ως Ανιχνευτής Προσοχής

Κίνητρο

Τα αποτελέσματα αποκαλύπτουν μια διαρκή τάση: η δρομολόγηση βάσει προσοχής σε όλα τα επίπεδα του δικτύου βελτιώνει benchmarks έντονης μαθηματικής συλλογιστικής (GSM8K, MMLU-Math) ενώ υποβαθμίζει benchmarks ανάκτησης γνώσης (MMLU συνολικά, $-1{,}5\%$ ή περισσότερο σε κάθε μέγεθος παραθύρου). Ανά μάθημα του MMLU, η υποβάθμιση εντοπίζεται σε μαθήματα ανάκτησης πληροφορίας (Βιολογία $-4{,}5\%$, Ιστορία $-2{,}5\%$, Φιλοσοφία $-3{,}2\%$), ενώ τα μαθήματα συλλογιστικής (Μαθηματικά, Πληροφορική) βελτιώνονται ή παραμένουν σταθερά.

Η υπόθεση είναι ότι η υποβάθμιση οφείλεται σε παρεμβολή στους μηχανισμούς ανάκτησης που εδράζουν στα πρώτα επίπεδα. Σύμφωνα με την υπάρχουσα βιβλιογραφία, τα ρηχά επίπεδα Μετασχηματιστών κυριαρχούν σε εργασίες ανάκτησης πληροφορίας, ενώ τα μεσαία-βαθύτερα επίπεδα είναι κρίσιμα για τη συλλογιστική. Εφαρμόζοντας AAR σε όλα τα επίπεδα, είναι πιθανό να διαταράσσονται πρότυπα επεξεργασίας για ανάκτηση στα πρώτα επίπεδα χωρίς να υπάρχουν ουσιαστικά οφέλη στη συλλογιστική.

Σχεδίαση: Επιλεκτική κατά Επίπεδο Εφαρμογή

Η *επιλεκτική κατά επίπεδο* AAR εφαρμόζει τον μηχανισμό μόνο σε επίπεδα $l \geq l_s$, ενώ τα επίπεδα $l < l_s$ διατηρούν αμετάβλητη την τυπική δρομολόγηση χωρίς βοηθητικές παραμέτρους. Αξίζει να σημειωθεί ότι αυτή δεν είναι απλώς μια αφαιρετική παραλλαγή: η καλύτερη διαμόρφωση OLMoE στα κύρια αποτελέσματα (AAR-T$_S$ με $l_s = 9$) χρησιμοποιεί ακριβώς αυτή τη σχεδίαση.

Σάρωση Επιπέδων: Αποτελέσματα

Εκπαιδεύοντας μοντέλα AAR-T$_S$ για $l_s \in \{1, 3, 5, 7, 9, 11, 13, 15\}$ με $M = 20$:

- **MMLU**: Η ακρίβεια ανακάμπτει μονοτονικά καθώς αυξάνεται το l_s. Για $l_s \geq 7$, η MMLU επιστρέφει στην τιμή της βασικής μεθόδου SFT ($53{,}92\%$), ενώ για $l_s = 9$–11 παρατηρείται ελαφρά υπέρβαση ($\approx 54{,}0\%$). Η ανάκαμψη ανά μάθημα δεν είναι ομοιόμορφη: οι κλάδοι που απαιτούν συνδυασμό ανάκτησης πληροφορίας και συλλογιστικής (Χημεία, Βιολογία, Οικονομικά, Φυσική) επωφελούνται περισσότερο από την προστασία των πρώτων επιπέδων.

- **GSM8K**: Διατηρεί σημαντικό μέρος της βελτίωσης στο εύρος επιπέδων έναρξης $l_s = 7$–9 ($\approx 53{,}5$–$53{,}8\%$ έναντι $54{,}10\%$ με πλήρη δρομολόγηση). Για $l_s \geq 13$, η βελτίωση μειώνεται, δείχνοντας ότι η μαθηματική συλλογιστική αξιοποιεί σήματα προσοχής σε ευρύ φάσμα επιπέδων και όχι μόνο στα πιο βαθιά.

- **ARC**: Παρουσιάζει πιο διαφοροποιημένη εικόνα. Το ARC-Easy ακολουθεί πρότυπο παρόμοιο με το GSM8K (η έγκαιρη εφαρμογή AAR ωφελεί), ενώ το ARC-Challenge ακολουθεί πρότυπο παρόμοιο με το MMLU (η προστασία πρώτων επιπέδων είναι κρίσιμη). Η διαφορά αποδίδεται στο γεγονός ότι τα πιο απλά ερωτήματα επωφελούνται περισσότερο από ενισχυμένη συλλογιστική, ενώ τα πιο απαιτητικά είναι πιο ευαίσθητα σε αλλαγές κατά την ανάκτηση πληροφορίας.

Το Βέλτιστο Εύρος: Επίπεδα 7–11

Τα επίπεδα 7–11 (μηδενική αρίθμηση) αποτελούν το βέλτιστο εύρος εκκίνησης: η MMLU ανακάμπτει στη βασική γραμμή ή την υπερβαίνει, ενώ το GSM8K διατηρεί το μεγαλύτερο μέρος της βελτίωσης. Η διαμόρφωση $l_s = 9$ αντιστοιχεί ακριβώς στην AAR-T$_S$ που αναφέρεται ως καλύτερο μοντέλο OLMoE στα κύρια αποτελέσματα.

Για εφαρμογές που δίνουν προτεραιότητα στη συλλογιστική, ο $l_s = 7$–9 διατηρεί τα ισχυρότερα κέρδη GSM8K ενώ ανακτά την καλή επίδοση στο MMLU. Για εφαρμογές όπου η ανάκτηση πληροφορίας είναι πιο σημαντική, η επιλογή $l_s \geq 11$ επαναφέρει σχεδόν πλήρως την απόδοση MMLU, έναντι ελαφρά μειωμένου κέρδους συλλογιστικής.

Επιλεκτική AAR ως Ανιχνευτής Κατά Βάθος

Πέρα από τα αποτελέσματα απόδοσης, η σχεδίαση αυτή παράγει ως παραπροϊόν έναν *ανιχνευτή προσοχής κατά βάθος*: μεταβάλλοντας μόνο *πού* εφαρμόζεται η AAR, μπορούμε να ανιχνεύσουμε τι είδους πληροφορία φέρουν τα βάρη προσοχής κάθε επιπέδου. Τα βάρη προσοχής ανά επίπεδο λειτουργούν ως αποτύπωμα της επεξεργασίας πληροφορίας εκείνου του επιπέδου. Η μονοτονική ανάκαμψη MMLU αποκαλύπτει ότι τα πρώτα επίπεδα είναι κρίσιμα για την ανάκτηση και αντιστοίχιση προτύπων, ενώ τα βαθύτερα επίπεδα εκτελούν πιο σύνθετη συλλογιστική.

Σε αντίθεση με παραδοσιακές μεθόδους ανίχνευσης που βασίζονται σε βοηθητικούς ταξινομητές εκπαιδευμένους σε αναπαραστάσεις, η επιλεκτική AAR ανιχνεύει τον λειτουργικό ρόλο των βαρών προσοχής σε κάθε επίπεδο μετρώντας τις επιδράσεις μέσω άμεσης παρέμβασης στη δρομολόγηση.

5. Συζήτηση

Αυτορρύθμιση μέσω δρομολόγησης. Παρέχοντας στον δρομολογητή ένα αποτύπωμα της επεξεργασίας συγκειμένου του μοντέλου, όπως κωδικοποιείται στα βάρη προσοχής, επιτρέπουμε μια μορφή αυτορρύθμισης: οι δρομολογητές μπορούν να συγκλίνουν σε διαμορφώσεις κάτω από τις οποίες τα επόμενα πρότυπα προσοχής μετατοπίζονται με τρόπους που συνεμφανίζονται με βελτιωμένη απόδοση. Αυτό δεν είναι ρητά εποπτευόμενο σήμα — αναδύεται από τυπική πρόβλεψη επόμενου token με εκπαίδευση αποκλειστικά των παραμέτρων δρομολόγησης, και τον βασικό μετασχηματιστή παγωμένο.

Από τη δρομολόγηση στην παραγωγή. Τα αποτελέσματα ανιχνεύουν μια αλυσίδα επίδρασης: η δρομολόγηση στο επίπεδο l προκαλεί αναδιαμόρφωση αναπαραστάσεων μέσω υπολειμματικών συνδέσεων, και τα βάρη προσοχής αλλάζουν στο επίπεδο $l{+}1$ (τεκμηριωμένη από ενίσχυση βυθισμάτων προσοχής με χρονική υστέρηση ενός επιπέδου). Ταυτόχρονα, παρατηρείται συντομότερη παραγωγή σε λανθασμένες απαντήσεις. Το γεγονός ότι μια αλλαγή εντοπισμένη αποκλειστικά στις παραμέτρους δρομολόγησης παράγει μετρήσιμες επιδράσεις στη συμπεριφορά παραγωγής δείχνει ότι δρομολόγηση, προσοχή και παραγωγή είναι στενά συζευγμένες.

Τάση ανάκτησης–συλλογιστικής. Τα επιλεκτικά κατά επίπεδο πειράματα δείχνουν ότι το κύκλωμα προσοχής–δρομολόγησης φέρει ποιοτικά διαφορετική πληροφορία σε διαφορετικά βάθη. Τα πρώτα επίπεδα είναι κρίσιμα για ανάκτηση πληροφορίας και ευαίσθητα στις αλλαγές από τη μέθοδο AAR· μετά τα πρώτα επίπεδα, η δομή σχετική με ανάκτηση πληροφορίας έχει εδραιωθεί και η AAR μπορεί να εφαρμοστεί χωρίς να τη διαταράξει. Αυτό υποδηλώνει ότι μια ενιαία παρέμβαση σε όλα τα επίπεδα είναι

υπο-βέλτιστη για εργασίες που συνδυάζουν ανάκτηση και συλλογιστική.

Δρομολόγηση ως βαθμός ελευθερίας. Συνολικά, τα αποτελέσματα ενισχύουν την άποψη ότι η δρομολόγηση δεν είναι απλώς μηχανισμός κατανομής υπολογισμού, αλλά ένας σημαντικός βαθμός ελευθερίας στα γλωσσικά μοντέλα. Μεταθέτοντας τη δρομολόγηση από μια τοπική, ανά-token απόφαση σε μια απόφαση που ενημερώνεται από τα ίδια πρότυπα προσοχής του μοντέλου, επιτυγχάνουμε καλύτερη απόδοση συλλογιστικής και μια πιο ερμηνεύσιμη διεπαφή με τους εσωτερικούς υπολογισμούς του δικτύου.

6. Συμπεράσματα και Μελλοντική Εργασία

Συμπεράσματα. Η παρούσα εργασία δείχνει ότι η επαύξηση των δρομολογητών MoE με βάρη προσοχής — ένα σήμα που υπολογίζεται αλλά απορρίπτεται στα τυπικά μοντέλα — βελτιώνει την απόδοση μαθηματικής συλλογιστικής αποκλειστικά μέσω δρομολόγησης. Η φασματική επέκταση (AAR-TF) παρέχει πρόσθετα κέρδη μέσω φασματικών χαρακτηριστικών του ίδιου παραθύρου. Αυτές οι βελτιώσεις επιτυγχάνονται εκπαιδεύοντας μόνο τις παραμέτρους δρομολόγησης, με τον βασικό μετασχηματιστή πλήρως παγωμένο. Το υποκείμενο κύκλωμα δρομολόγησης–προσοχής εδραιώνει ότι η δρομολόγηση δεν είναι απλώς ένας μηχανισμός κατανομής εργασίας, αλλά ένας κρίσιμος βαθμός ελευθερίας που διαμορφώνει την εσωτερική επεξεργασία και εξωτερική συμπεριφορά του μοντέλου.

Περιορισμοί. Εντοπίζουμε δύο κύριους περιορισμούς. Πρώτον, η AAR απαιτεί ρητή υλοποίηση των βαρών προσοχής, αποκλείοντας τη χρήση FlashAttention και εισάγοντας επιβάρυνση μνήμης ανάλογη με το μήκος ακολουθίας και τον αριθμό ενεργών επιπέδων. Δεύτερον, τα πειράματα εστιάζουν σε προσαρμογή μετά την διαδικασία SFT με παγωμένο μοντέλο, η οποία εισάγει εξάρτηση από το ιστορικό εκπαίδευσης και υψηλότερη διακύμανση σε σχέση με κοινή εκπαίδευση όλων των παραμέτρων.

Μελλοντικές κατευθύνσεις.

- **Ενσωμάτωση κατά την προ-εκπαίδευση**. Η ενσωμάτωση AAR ήδη από την προ-εκπαίδευση θα επέτρεπε στο μοντέλο και τους δρομολογητές να συν-εκπαιδεύονται, ενδεχομένως ενισχύοντας τη σύζευξη δρομολόγησης–προσοχής και μειώνοντας την εξάρτηση από το εύρος επιπέδων.
- **Πλουσιότερες φασματικές αναπαραστάσεις**. Εκπαιδεύσιμα φίλτρα ή εξαγωγείς χαρακτηριστικών ενδέχεται να αποδώσουν επιπλέον κέρδη.
- **Προσαρμοστική επιλογή επιπέδων**. Η τρέχουσα επιλεκτική σχεδίαση βασίζεται σε χειροκίνητη σάρωση υποσυνόλου επιπέδων. Ένας εκπαιδευμένος ή εξαρ-

τώμενος από την είσοδο μηχανισμός επιλογής θα καθιστούσε την AAR ευρέως εφαρμόσιμη χωρίς ρύθμιση προσαρμοσμένη στο εκάστοτε μοντέλο.

- **Δρομολόγηση ως μηχανισμός καθοδήγησης**. Το κύκλωμα δρομολόγησης–προσοχής υποδηλώνει ότι η δρομολόγηση μπορεί να αποτελεί σημείο στοχευμένης παρέμβασης για διαμόρφωση συμπεριφοράς χωρίς τροποποίηση του μηχανισμού προσοχής. Η αξιοποίηση αυτής της προοπτικής στο πλαίσιο ελεγχόμενης παραγωγής, ασφάλειας ή ευθυγράμμισης αποτελεί ελκυστική κατεύθυνση: οι τροποποιήσεις δρομολόγησης είναι ελαφριές, ερμηνεύσιμες και εντοπισμένες σε συγκεκριμένα επίπεδα.

Chapter 1

Introduction

1.1 The Rise of Large Language Models

The past decade has reshaped natural language processing. Driven by the scaling of neural language models to tens and hundreds of billions of parameters, trained on web-scale text, modern systems display capabilities that were out of reach a few years ago: multi-step reasoning, code synthesis, instruction following, and open-ended dialogue. Many of these behaviors are *emergent*, appearing only beyond a certain scale rather than being explicitly engineered [2]. At the center of this progress sits the Transformer [1], which processes a sequence through a stack of layers, each refining a token's representation by attending to its context and then transforming it through a feed-forward network.

This capability comes at a structural cost. A dense Transformer applies *every* parameter to *every* token at *every* layer, so the compute spent at inference grows with the parameter count. As models are scaled to acquire new capabilities, this coupling becomes a fundamental tension: the very growth that yields stronger models also makes each forward pass proportionally more expensive. Resolving this tension— adding capacity without paying for it on every token—is the central motivation of the Mixture-of-Experts (MoE) architecture.

1.2 Mixture-of-Experts as Conditional Computation

Mixture-of-Experts (MoE) models [3, 4] break the link between capacity and compute through *conditional computation*. Instead of applying a single feed-forward network to every token, each MoE layer holds a pool of E independent expert sub-networks and a small, learned *router* that activates only $K \ll E$ of them per token, per layer. The model's total parameter count scales with E, but the number of parameters actually used in a forward pass is fixed by K. Capacity and compute are thus decoupled: a much larger model can be served at roughly the inference cost of a small one.

This design has made MoE a dominant paradigm for efficient large-scale language modeling. Architectures such as Mixtral [5], OLMoE [6], and Qwen-MoE [7] match or exceed dense models of comparable *active*-parameter budget. Within this design, the router is a significant component: it does not merely allocate computation, but it notably

determines which computational pathways shape each token's representation at every layer. The quality of an MoE model is therefore tightly bound to the quality of its routing decisions—which makes the information the router has access to a first-order concern.

1.3 The Routing Bottleneck

The standard router operates from a narrow vantage point. For a token at layer l, it computes a linear projection of the *post-attention* hidden state $h_i^l \in \mathbb{R}^d$ and selects the top-K experts from the resulting scores. The hidden state is a compressed summary of the token together with its context, filtered through l layers of computation. It carries contextual information, but only *implicitly*: it records where the token's representation has ended up, but not how it was shaped, or explicitly, how it relates to its context.

This thesis operates on this limitation: the blindness of the router to relational information regarding the token's surrounding tokens. Specifically, we take advantage of the fact that the model already computes a such relational signal, in the same layer, for free. The *attention weights* $\{w_i^l(t)\}$ produced in the same layer encode the model's contextual focus— which past tokens were judged relevant, and how that relevance was distributed. These weights are computed inside the attention sub-layer, used to form the attention output, and then *discarded*. By the time the router acts, this explicit relational structure has been collapsed into a single vector. The router sees the *output* of the attention process but not the process itself. Thus, the contextual structure that attention makes explicit is available in the layer, yet unused at routing time.

We can identify this as a routing bottleneck: the routers are popularly confined to the post-attention hidden state, while contextual information is available, but unused.

1.4 Thesis Motivation and Contribution

This thesis investigates one deliberately simple question: *what happens if we give the router access to contextual information?*

We propose **Attention-Aware Routing (AAR)**, an augmentation of the MoE router that incorporates a sliding window of recent attention weights as an additional input signal. The router thus conditions its expert selection not only on the hidden state, but on the relational contextual structure that shaped it—which context it attended to, and how that attention was distributed.

To attribute any change cleanly to routing, we adopt a strict experimental protocol: the entire base transformer is kept *frozen*, and only the routing parameters are trained. Routing is thereby isolated as the sole variable, so that observed effects on performance and on the model's internal behavior can be traced to the routing decision itself rather than to continued adaptation of the backbone.

The main contributions of this thesis are:

1. **AAR improves mathematical reasoning**. Temporal and spectral routers that improve GSM8K mathematical reasoning by up to $+3.37$ percentage points (AAR-TF) over a routing-only SFT baseline on OLMoE, while preserving performance on other benchmarks—achieved by training the routing parameters alone, with the base model frozen.

2. **AAR reveals and exploits a routing-attention coupling**. We present empirical evidence that AAR-induced routing changes do not stay confined to the router: a routing change at layer l propagates through the residual stream to reshape attention at layer $l+1$, amplifying attention sinks one layer downstream *without any direct update to the attention mechanism itself*. The same routing changes manifest externally as shorter incorrect generations, while correct answers remain unchanged in length.

3. **Layer-selective AAR can serve as a depth-wise attention probe**. Restricting AAR to a subset of layers reveals a clean retrieval–reasoning tension across transformer depth: early layers are critical for factual retrieval and sensitive to routing changes, whereas reasoning gains consolidate when AAR is applied to deeper layers. Layer-selective AAR thus can serve as a probe that reads off functional specialization across depth through direct intervention, without the use of auxiliary classifiers.

1.5 Thesis Outline

The remainder of this thesis is organized as follows.

Chapter 2 provides the necessary background on Transformer language models, the pre-training and fine-tuning paradigm, and the Mixture-of-Experts architecture. It establishes the notation used throughout, with particular emphasis on the self-attention weights that the routing mechanism of this thesis later exploits, and the token-routing problem that AAR augments upon.

Chapter 3 introduces Attention-Aware Routing in full. It describes the AAR-T and AAR-TF mechanisms, the router interpolation scheme, and the frozen-backbone training setup, and then reports the main experimental results: window-size ablations, the internal routing–attention circuit, the generation-length effects, complementary experiments (load balancing, attention statistics), and generalization to other MoE models.

Chapter 4 studies layer-selective AAR, restricting the mechanism to a subset of layers. It shows that applying AAR only from mid-to-deep layers recovers the MMLU degradation seen under full-layer routing, reveals a depth-wise retrieval–reasoning distinction, and explains why the layer-selective variant outperforms the all-layer baseline—reframing the layer sweep as a depth-wise attention probe.

Chapter 5 discusses the broader implications of the results: what AAR reveals about the role of routing in language models, the coupling between routing, attention, and generation behavior, and the retrieval–reasoning tension as a structural property of transformer depth.

Chapter 6 concludes with a summary of contributions, a discussion of limitations, and directions for future work. Supporting material is provided in the appendices.

Chapter 2

Language Models and Mixture of Experts

This chapter provides the technical background necessary to follow the contributions of this thesis. Section 2.1 introduces neural language modeling and the pre-training and fine-tuning paradigm. Section 2.2 describes the Transformer architecture in detail, with particular attention to the self-attention weights that the routing mechanism of this thesis later exploits. Section 2.3 covers Mixture-of-Experts models, the token routing problem, and the specific models used in our experiments.

2.1 Neural Language Modeling

2.1.1 Language Modeling as Next-Token Prediction

Bengio et al. [8] introduced neural language modeling as a next-token prediction task, where the model learns to predict the probability of the next token (unit of language) x_t given the preceding tokens $x_{<t}$:

$$P(x_t \mid x_{<t}) = \text{softmax}(f_\theta(x_{<t}))\,, \tag{2.1}$$

where f_θ is a neural network parameterized by θ. Factorizing the joint probability of a sequence $x_{1:T}$ autoregressively as $P(x_{1:T}) = \prod_{t=1}^{T} P(x_t \mid x_{<t})$, the model is trained to maximize the likelihood of the training corpus, which is equivalent to minimizing the cross-entropy (negative log-likelihood) loss

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}; \theta). \tag{2.2}$$

This objective requires no human annotation: any text corpus supplies its own supervision, since the target at each position is simply the next observed token. This property is what enables training on web-scale corpora and underlies the self-supervised pre-training paradigm discussed below.

2.1.2 Training with Gradient Descent

To train a neural model we use gradient descent to update the parameters θ in the direction that decreases the loss:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta), \tag{2.3}$$

where η is the learning rate. In practice we use mini-batch stochastic gradient descent (SGD) or, more commonly for language models, the Adam optimizer [9] and its decoupled-weight-decay variant AdamW [10], which adapt per-parameter step sizes from running estimates of the first and second moments of the gradients. Gradients are obtained efficiently by backpropagation, and training proceeds over many passes through the data until convergence.

2.1.3 Pre-training and Fine-tuning

Modern language models are built in two stages. In *pre-training*, the model optimizes the autoregressive objective above on a very large, heterogeneous corpus, acquiring broad linguistic and world knowledge in a task-agnostic manner [11, 12]. The resulting model is then *fine-tuned* for downstream use. The dominant form of fine-tuning for instruction-following models is *supervised fine-tuning* (SFT), in which the same next-token objective is applied to curated prompt–response pairs so that the model learns to follow instructions and adopt a helpful response format [13]. SFT is frequently followed by alignment from human or AI preference feedback, e.g. reinforcement learning from human feedback [14]. The fully open OLMo and OLMoE efforts [15, 6] release the data, code, and checkpoints for both stages, and the open-instruct SFT framework and its Tülu data mixtures [16] are the basis for the router-only fine-tuning experiments in this thesis.

2.1.4 Scale, In-Context Learning, and Reasoning

As models, data, and compute are scaled jointly, language models acquire qualitatively new behaviors. The GPT-3 model [17] demonstrated *in-context learning*: the ability to perform a task from a few demonstrations supplied in the prompt, with no parameter updates. Subsequent work showed that chain-of-thought prompting—eliciting intermediate reasoning steps—substantially improves performance on arithmetic and symbolic reasoning benchmarks such as GSM8K [18], and that several such capabilities appear only beyond a certain scale [2]. These observations motivate efficient ways to grow model capacity, the central concern of Mixture-of-Experts architectures (Section 2.3). The open LLaMA family [19, 20] established that strong performance is attainable at moderate active-parameter counts, and its architectural choices (RMSNorm, SwiGLU, rotary embeddings) are inherited by most of the models considered here.

2.2 The Transformer Architecture

Transformers are the dominant architecture for modern neural language models, introduced by Vaswani et al. [1]. A decoder-only Transformer maps a sequence of tokens to a sequence of contextualized representations through a stack of L identical layers, each composed of a self-attention sublayer and a position-wise feed-forward sublayer, both wrapped in residual connections and layer normalization. We write $\mathbf{h}_i^l \in \mathbb{R}^d$ for the hidden state of token i at layer l, where d is the model dimension.

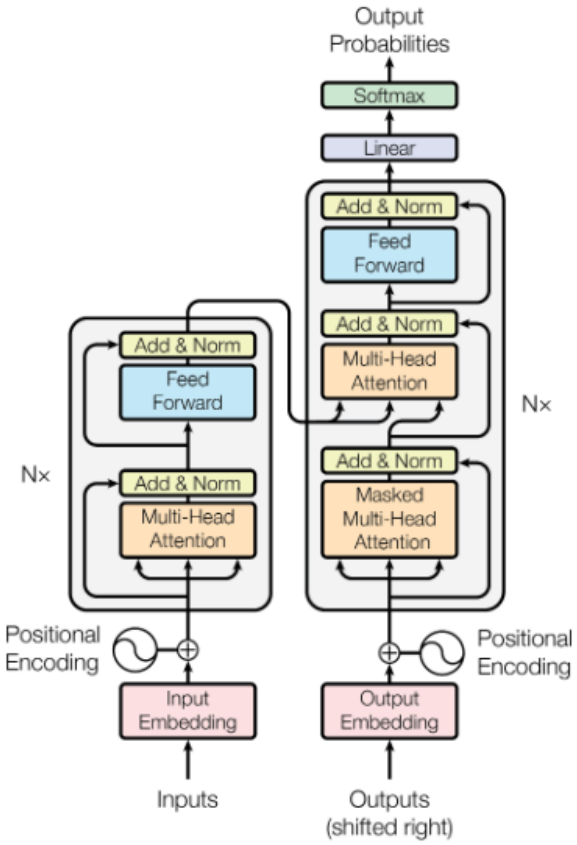


Figure 2.1. *The Transformer encoder–decoder architecture as introduced by Vaswani et al. [1]. The models in this thesis use only the decoder stack (right half), which applies masked self-attention followed by a position-wise feed-forward sublayer at each layer.*

2.2.1 Tokenization and Embeddings

A Transformer does not operate on raw characters but on *tokens* produced by a subword tokenizer. Byte-pair encoding [21] and its byte-level and unigram variants [22] learn a fixed vocabulary that balances sequence length against the rate of out-of-vocabulary symbols, segmenting frequent words into single tokens while decomposing rare words into reusable subword units. Each vocabulary entry is mapped to a learned embedding vector via an embedding matrix $W_E \in \mathbb{R}^{|V| \times d}$, yielding the layer-0 representations $\mathbf{h}_i^0$.

At the output, a (typically tied) projection maps the final hidden states back to vocabulary logits.

2.2.2 Self-Attention

Self-attention lets each token aggregate information from other tokens in the sequence. For each token i, the layer projects its hidden state into a query, key, and value:

$$\mathbf{q}_i = W_Q\,\mathbf{h}_i^l, \qquad \mathbf{k}_i = W_K\,\mathbf{h}_i^l, \qquad \mathbf{v}_i = W_V\,\mathbf{h}_i^l, \tag{2.4}$$

with $W_Q, W_K, W_V \in \mathbb{R}^{d_k \times d}$. The attention output for token i is a convex combination of the value vectors, weighted by the normalized compatibility between its query and every key:

$$\mathrm{Attn}(\mathbf{h}_i^l) = \sum_t w_i^l(t)\,\mathbf{v}_t, \qquad w_i^l(t) = \mathrm{softmax}_t\!\left(\frac{\mathbf{q}_i^\top \mathbf{k}_t}{\sqrt{d_k}}\right). \tag{2.5}$$

In a decoder-only (causal) model the sum runs only over $t \le i$, so a token attends to itself and its predecessors. Stacking over all positions gives the familiar matrix form

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\!\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{2.6}$$

where the scaling by $\sqrt{d_k}$ keeps the dot products in a numerically stable range. In practice the layer uses *multi-head* attention: H heads each operate in a d/H-dimensional subspace with their own projections, and the per-head outputs are concatenated and mixed by an output projection W_O, allowing the model to attend to several relational patterns simultaneously.

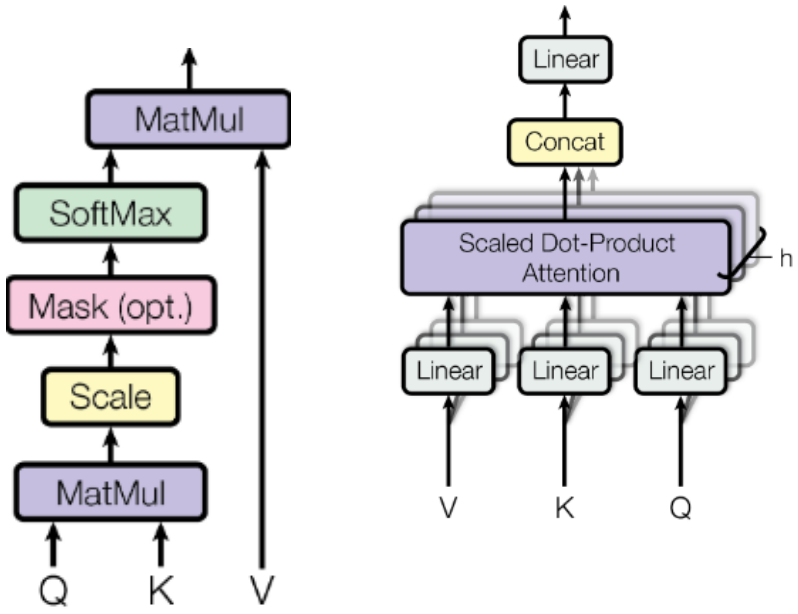
The quantity $w_i^l(t)$ in Equation (2.5) is the *attention weight* that token i places on token t at layer l; collecting these over all pairs gives a row-stochastic *attention weight matrix* $A^l \in \mathbb{R}^{n\times n}$ with $A^l_{i,t} = w_i^l(t)$. This matrix is a direct, layer-local readout of which context the model deems relevant to each token. It is computed in every layer regardless of the downstream architecture.

Naive attention has time and memory cost $O(n^2 d)$ in the sequence length n, which dominates at long context. FlashAttention [23] computes the exact same result with an IO-aware, tiled kernel that avoids materializing the full $n \times n$ matrix in high-bandwidth memory.

2.2.3 Feed-Forward Sublayer

Each attention sublayer is followed by a position-wise feed-forward network (FFN) applied independently to every token:

$$\mathrm{FFN}(\mathbf{h}) = W_2\,\sigma(W_1\mathbf{h} + b_1) + b_2, \tag{2.7}$$



(a) *Scaled Dot-Product Attention.*

(b) *Multi-Head Attention with H parallel heads.*

Figure 2.2. *Attention mechanisms from Vaswani et al. [1].* ***(a)*** *Scaled Dot-Product Attention computes query–key compatibilities, scales by $1/\sqrt{d_k}$, applies softmax, and weights the values.* ***(b)*** *Multi-Head Attention projects queries, keys, and values into H lower-dimensional subspaces, runs Scaled Dot-Product Attention in each, and concatenates the results.*

a two-layer MLP that expands to an inner dimension $d_{\text{ff}} \gg d$ and projects back. Early Transformers used ReLU or GELU [24] nonlinearities; most recent models adopt the gated SwiGLU variant [25]. Because d_{ff} is several times d, the FFN typically holds the majority of a dense Transformer's parameters. It is precisely this sublayer that Mixture-of-Experts replaces with a collection of independently parameterized experts, making the FFN the locus of conditional computation.

2.2.4 Positional Encodings

Self-attention is permutation-equivariant and therefore order-blind on its own, so position information must be injected explicitly. The original Transformer added fixed sinusoidal encodings [1], while later models used learned absolute embeddings. The models in this thesis instead use Rotary Position Embeddings (RoPE) [26], which rotate the query and key vectors by an angle proportional to their absolute position, so that the attention score in Equation (2.5) depends only on the *relative* offset $i - t$.

2.3 Mixture-of-Experts Language Models

Mixture-of-Experts (MoE) [3] is an architecture that replaces the dense feed-forward sublayer with a collection of E independent expert networks $\{F_1, \ldots, F_E\}$ and a learned router that selects a small subset of $K \ll E$ experts for each token. The model's total parameter count scales with E, but the active parameter count per forward pass is determined only by K, enabling a favorable decoupling of capacity from compute—larger models for roughly fixed inference cost.

2.3.1 The Expert Layer

Each expert F_e is itself a standard FFN of the form in Section 2.2.3, with its own parameters. Given the router gate values $\mathbf{g}_i^l$ (defined below), the output of the MoE layer for token i is the gate-weighted sum over the selected experts:

$$\mathbf{h}_i^{l+1} = \mathbf{h}_i^l + \sum_{e \in \text{Top-K}(\mathbf{g}_i^l)} g_{i,e}^l \, F_e\big(\mathbf{h}_i^l\big) \,, \tag{2.8}$$

where the residual connection is kept as in the dense case. Only the K selected experts are evaluated, so the cost of the sum is independent of E. Several modern designs add one or more *shared* experts that process every token unconditionally, alongside the routed experts, to capture common computation and stabilize training; this is the design adopted by Qwen1.5-MoE and the DeepSeek family [27].

2.3.2 Token Routing

The router maps each token's hidden state to a distribution over experts and selects the top-K by score. The standard formulation computes routing logits as a linear function of the hidden state:

$$\mathbf{g}_i^l = \text{softmax}\big(W_l \, \mathbf{h}_i^l\big) \,, \quad W_l \in \mathbb{R}^{E \times d}, \tag{2.9}$$

and selects the K experts with the highest scores. This is the baseline routing mechanism that the attention-aware router of this thesis augments. Variants differ chiefly in K and in how the selection is made: the Switch Transformer [4] routes each token to a single expert ($K = 1$) for maximal efficiency, while GShard, Mixtral, and OLMoE use top-K routing with $K \geq 2$ [28, 5, 6]. Expert-choice routing [29] inverts the assignment so that each expert selects its top tokens, guaranteeing balanced load by construction at the cost of a non-causal selection.

A structural feature of Equation (2.9) is central to this thesis: the router observes only the post-attention hidden state $\mathbf{h}_i^l$—a single vector that compresses the token together with its aggregated context—and *not* explicitly the context, or the attention weight matrix A^l from Equation (2.5) that was computed in the same layer. The contextual struc-

ture that attention makes explicit is thus available but unused at routing time, a gap that attention-aware routing is designed to close.

2.3.3 Load Balancing

Left unconstrained, routers tend to collapse onto a small set of experts: a rich-get-richer dynamic in which frequently chosen experts receive more gradient signal, improve faster, and are chosen even more often, leaving the remaining experts undertrained [3]. To counteract this, MoE models add a differentiable auxiliary load-balancing loss [28, 4]:

$$\mathcal{L}_{\text{lb}} = \alpha \, E \sum_{e=1}^{E} f_e \, P_e, \tag{2.10}$$

where f_e is the fraction of tokens in the batch routed to expert e, P_e is the mean router probability assigned to expert e, and α scales the penalty. The loss is minimized when both quantities are uniform across experts, encouraging balanced utilization. Training is often further stabilized with a router z-loss that penalizes large routing logits [30]. OLMoE [6], the primary model of this thesis, is trained dropless (no tokens are discarded under capacity limits) with both the load-balancing loss and the router z-loss during pretraining, and no auxiliary losses during SFT.

2.3.4 Key MoE Models

We briefly summarize the MoE models referenced or used in this thesis.

Switch Transformer. Fedus, Zoph, and Shazeer [4] gave the first large-scale demonstration that sparsely activated experts scale efficiently, simplifying routing to top-1 and introducing capacity factors and training tricks that made trillion-parameter sparse models trainable.

GShard and GLaM. GShard [28] scaled sparse MoE for multilingual translation with automatic sharding and the now-standard top-2 auxiliary-loss recipe, and GLaM [31] showed that an MoE language model can match a much larger dense model's quality at a fraction of the inference compute.

OLMoE. Muennighoff et al. [6] released a fully open MoE—data, code, logs, and checkpoints—with 64 experts per layer and top-8 routing, dropless training, and load-balancing. It being open-source and having a fine-grained expert pool make it the primary model for the routing analyses in this thesis.

Qwen1.5-MoE. The Qwen1.5-MoE model [7] combines fine-grained routed experts with always-on shared experts and is upcycled from a dense checkpoint; it is the second model used in our experiments.

Chapter 3

Attention-Aware Routing

3.1 Introduction

In Mixture-of-Experts (MoE) architectures [3, 4, 28], language models scale capacity while keeping inference cost manageable by activating only a subset of experts per token. The router—the mechanism that decides which experts process each token—is a load-bearing component [32]: it not only ensures efficient allocation of computation, but determines which computational pathways shape each token's representation at every layer.

Standard routers make this decision from a narrow vantage point. Given a token's hidden state $\mathbf{h}_i^l$, the router computes a linear projection and selects the top-k experts [3]. The hidden state carries contextual information, but only implicitly; it does not explicitly represent how a token relates to others in the sequence.

We propose to augment the router with a sliding window of recent attention weights. While hidden states summarize what each token represents, attention weights encode compressed relational intra-sequence structure: which positions the model attends to, which it ignores, and where focus concentrates. The attention weights are not unstructured signal: as illustrated in Figure 3.1, they naturally carry task-specific information that clusters by domain at shallow layers and evolves with depth.

We introduce **Attention-Aware Routing (AAR)**, which augments the standard router with a sliding window of attention weights, incorporating both time-domain and frequency-domain features. **AAR-T** operates on the raw attention window in the time domain; **AAR-TF** extends it with a parallel frequency-domain router on the DFT magnitude of the same window. To gauge AAR's effect on performance experimentally, we isolate it as the sole variable, by training the routing parameters of the base transformers alone, while keeping all other weights frozen.

3.2 Related Work

We summarize related work across several axes, highlighting how our approach differs from and builds upon prior research. More detailed discussion of related work was

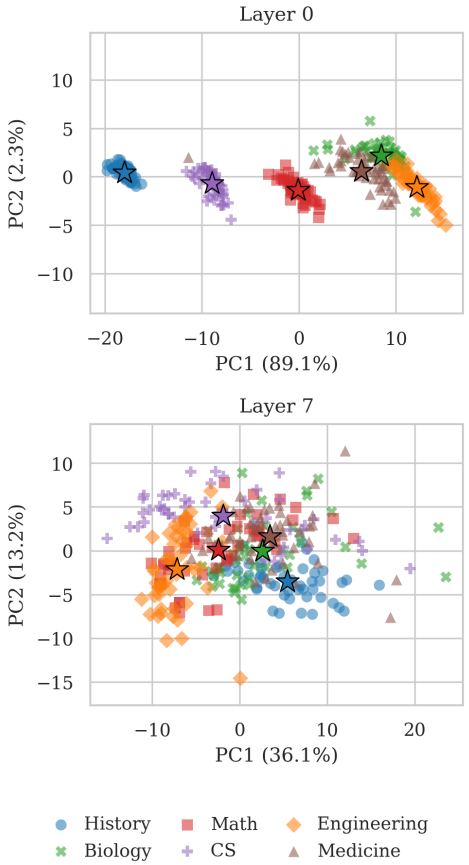


Figure 3.1. ***PCA of Attention Window Fingerprints.*** *We visualize the first two principal components of the attention windows (only including the 128-sized attention weight vectors, averaged across heads) for different MMLU subjects. Each point represents the average attention weight vector of the final question's tokens in each 5-shot prompt. At Layer 0, attention patterns exhibit strong subject-specific clustering. This clustering becomes more entangled by Layer 7, showcasing the progression of the internal processing. Notice how semantically similar subjects get relatively closer, compared to dissimilar subjects.*

provided in Chapter 2. Here we focus on the specific connections to our method and findings.

MoE routing. Standard token-choice routing computes a softmax over expert logits derived from the token representation and selects the top-k experts [3]. Load-balancing auxiliary losses encourage uniform expert utilization [4]. Expert-choice routing [29] inverts the assignment. Our work keeps token-choice routing intact and augments it with an attention-conditioned auxiliary signal.

Various routers. Qiu et al. [33] implement MoE routing with a recurrent unit, shared across layers. The input to the router at each layer comes from the hidden state at the current layer, as well as previous routing decisions (the recurrent unit's hidden state). In Do et al. [34], a fixed randomly initialized hypernetwork generates the router's parameters, conditioned on a trainable router embedding; this alleviates representation collapse and limits the number of experts used during inference. Similar to these works, our routing method does not rely on the hidden state alone to generate each layer's routing logits, but incorporates attention weights as an additional signal.

Attention sinks. Xiao et al. [35] observe that transformers concentrate disproportionate attention mass on the first token (the "attention sink"), which stabilizes representations under streaming contexts. Barbero et al. [36] show that sink presence slows information mixing and increases robustness to prompt perturbations. We find that AAR systematically strengthens attention sinks in layers following the routing modification, linking this to improved reasoning.

Layer-wise functional specialization. Yang, Gao, and Wu [37] and Song et al. [38] show that shallow LLM layers dominate likelihood and retrieval tasks while mid-to-deep layers are essential for reasoning and generation. Yang et al. [39] distinguish latent reasoning from factual shortcuts across depth.

Attention-aware interventions for reasoning. Nguyen, Huu-Tien, and Inoue [40] improve chain-of-thought reasoning through attention-aware interventions at inference time, based on empirical hard-written rules. Our work differs in that the attention signal is incorporated into the routing mechanism during training, making the effect persistent, learnable, and architecture-native.

Overthinking in chain-of-thought. Li et al. [41] and Su et al. [42] show that longer reasoning chains correlate with higher error rates. Our finding that AAR reduces output length specifically on incorrect answers connects routing-induced changes to this line of research.

3.3 Method

3.3.1 Standard MoE Routing

In a typical MoE layer of a transformer LLM at depth l, given a token representation $\mathbf{h}_i^l \in \mathbb{R}^d$ after the attention sub-layer, the router computes expert selection logits as

$$\mathbf{g}_{\text{std}} = W_r^l \mathbf{h}_i^l, \quad W_r^l \in \mathbb{R}^{E \times d}, \tag{3.1}$$

where E is the number of experts. Top-k experts are selected after applying a softmax over $\mathbf{g}_i$, and the token is processed by a weighted combination of the selected experts' outputs [3].

3.3.2 AAR-T: Time-Domain Attention Router

At each transformer layer, the attention submodule computes attention weights over the input sequence as part of its normal forward pass. These weights are never utilized again after computing the attention output, yet they encode a compact summary of how the model has distributed its focus across the context at that layer. We propose to make use of this otherwise discarded signal by including it in the routing module's input.

Notation. For MoE layer $l \in \{0, \ldots, L-1\}$, let $\mathbf{h}_i^l \in \mathbb{R}^d$ denote the hidden state of token i after the attention sub-layer, and let $\mathbf{w}_i^l \in \mathbb{R}^M$ denote the attention weight vector at token position i: averaged across heads, taken as a sliding window over the last M positions and zero-padded on the left for $i < M$. We write E for the number of experts and k for the number of selected experts (top-k). Every parameter introduced below ($W_r^l, A_l, \tilde{A}_l, G_l^{(1)}, G_l^{(2)}$) is layer-specific; no weights are shared across layers.

The auxiliary time-domain router A_l is a single linear layer that maps the M-dimensional attention weight vector $\mathbf{w}_i^l$ to E expert logits:

$$\mathbf{g}_{\text{attn}} = A_l^\top \mathbf{w}_i^l, \quad A_l \in \mathbb{R}^{M \times E}. \tag{3.2}$$

The attention weight vector $\mathbf{w}_i^l$ is constructed as follows: for the very first token of a prompt, the vector is all zeros. When processing the next token, the window shifts by one position. For tokens after position M, the window is full and the weights occupy the vector in first-in, first-out order.

3.3.3 AAR-TF: Frequency-Domain Extension

The attention window $\mathbf{w}_i^l$ can be viewed as a discrete signal in the time domain. Taking its DFT magnitude yields a complementary representation with two useful properties:

(1) *shift-invariance*, so attention patterns with the same shape but different absolute positions within the window map to similar router inputs; and (2) a *non-linear transformation*, allowing the linear frequency router to access features that would otherwise require a deeper module to extract.

We define the frequency-domain attention vector as

$$\tilde{w}_i^l[k] = \left| \sum_{t=1}^{M} w_i^l(t)\, e^{-\jmath 2\pi(k-1)(t-1)/M} \right|, \tag{3.3}$$

for $k \in \{1, \ldots, K\}$ with $K = \lfloor M/2 \rfloor + 1$. We retain only the first K bins, since $\mathbf{w}_i^l$ is real-valued and its DFT magnitude is symmetric about the Nyquist frequency. We apply $\log(1 + \tilde{\mathbf{w}}_i^l)$ to compress the dynamic range before passing it to the frequency-domain router:

$$\mathbf{g}_{\text{fft}} = \tilde{A}_l^\top \tilde{\mathbf{w}}_i^l, \quad \tilde{A}_l \in \mathbb{R}^{K \times E}. \tag{3.4}$$

Appendix C develops the shift-invariance argument in detail, and presents empirical evidence that the spectral structure of attention windows varies systematically across layers—providing post-hoc justification for why the frequency-domain router benefits from being applied selectively to deeper layers.

3.3.4 Router Interpolation

Each context router is associated with a confidence gate conditioned on the hidden state and its respective context signal:

$$\boldsymbol{\alpha}_1 = \sigma\left(G_l^{(1)}\left[\mathbf{h}_i^l \,\|\, \mathbf{w}_i^l\right]\right), \quad G_l^{(1)} \in \mathbb{R}^{E \times (d+M)}, \tag{3.5}$$

$$\boldsymbol{\alpha}_2 = \sigma\left(G_l^{(2)}\left[\mathbf{h}_i^l \,\|\, \tilde{\mathbf{w}}_i^l\right]\right), \quad G_l^{(2)} \in \mathbb{R}^{E \times (d+K)}. \tag{3.6}$$

The gates determine per-expert interpolation weights, with the standard router retaining the largest share of the total weight:

$$\beta_1 = \frac{\boldsymbol{\alpha}_1}{2}, \quad \beta_2 = \frac{\boldsymbol{\alpha}_2}{2}, \quad \beta_0 = \mathbf{1} - \beta_1 - \beta_2. \tag{3.7}$$

The final routing logits are then:

$$\mathbf{g}_i = \beta_0 \odot \mathbf{g}_{\text{std}} + \beta_1 \odot \mathbf{g}_{\text{attn}} + \beta_2 \odot \mathbf{g}_{\text{fft}}. \tag{3.8}$$

This formulation recovers AAR-T when $\boldsymbol{\alpha}_2 \equiv \mathbf{0}$, and standard MoE routing when $\boldsymbol{\alpha}_1 \equiv \boldsymbol{\alpha}_2 \equiv \mathbf{0}$.

3.3.5 Layer-Selective AAR

Both AAR-T and AAR-TF can be applied selectively across depth. We introduce the *layer-selective* variant: AAR_S is applied only on a subset S of layers $\{0, \ldots, L-1\}$, while standard routing is used elsewhere. This is motivated by prior work showing that shallow and deep layers play functionally distinct roles in LLMs [37, 38, 39]: rather than applying AAR uniformly, we let the choice of S reflect where in the network the attention signal carries task-relevant structure. Section 4 shows that this layer-selective design also serves as a depth-wise attention probe, letting us read off where different capabilities are encoded in attention by varying only where routing is modified.

3.4 Experiments

3.4.1 OLMoE

For our main experiments, we use OLMoE [6], an open-source sparse Mixture-of-Experts language model with 1B active parameters and 7B total parameters. The model consists of 16 decoder blocks, each containing a feedforward layer with 64 experts, of which $k = 8$ are active per token, and has a sequence length equal to 4096.

Specifically, we use the `OLMoE-1B-7B-0125-SFT` checkpoint[1], which was obtained by supervised fine-tuning of the pretrained `OLMoE-1B-7B-0125`[2]. The pretrained model was trained on a mixture of Dolmino-mix-1124[3] and OLMoE-mix-0924[4]. For post-training, following [16], the model was fine-tuned on the Tulu-3 SFT mixture[5].

3.4.2 Training Configuration

To isolate the effect of the routing mechanism, we freeze all model parameters except for the original routers and the newly introduced attention routers. We fine-tune using the AdamW optimizer [10] with the hyperparameters listed in Table 3.1.

The attention routers are initialized by sampling from a normal distribution with $\sigma = 0.01$. Experiments are repeated across 5 random seeds. Training is performed on 4×A100 (64GB) GPUs; a single OLMoE run takes approximately 4 GPU hours.

Unless stated otherwise, we train on a 25% sample (∼220K examples) of Tulu-3 [16], the SFT dataset used for OLMoE, including high-quality math, code, and instruction-following samples.

[1] `https://huggingface.co/allenai/OLMoE-1B-7B-0125-SFT`
[2] `https://huggingface.co/allenai/OLMoE-1B-7B-0125`
[3] `https://huggingface.co/datasets/allenai/dolmino-mix-1124`
[4] `https://huggingface.co/datasets/allenai/OLMoE-mix-0924`
[5] `https://huggingface.co/datasets/allenai/tulu-3-sft-mixture`

Hyperparameter	Value
Learning rate	1×10^{-4}
LR scheduler	Linear
Warmup ratio	0.03
Weight decay	0.0
Batch size	1024
Max sequence length	512
Precision	BF16

Table 3.1. *Training hyperparameters.*

3.4.3 Baselines

We compare AAR against the original OLMoE-0125-SFT model and identical routing-only SFT runs without the auxiliary routers, isolating the effect of the additional attention signal from that of continued training. We verify our results with the same comparisons on Qwen1.5 [7] (24 layers, 2.7B active parameters, 14.3B in total)[6] with its corresponding chat template.

3.4.4 Evaluation

We evaluate on GSM8K [18], MMLU [43], HumanEval [44], IFEval [45], BBH [46] and ARC-Challenge and ARC-Easy [47]. These tasks cover a wide spectrum of linguistic capabilities, including multi-step reasoning, code generation, instruction following, and general knowledge across various domains. Evaluation configurations and hyperparameters are detailed in Appendix A.

3.4.5 Main Results

Table 3.2 consolidates results across reasoning and factual benchmarks. AAR-TF, applied to layers $S = \{9, \ldots, 15\}$ with $M = 40$, achieves the best GSM8K performance at 55.36 ± 0.74, a gain of $+3.37$ pp over routing-only SFT. AAR-T at $M = 20$ already improves GSM8K from 51.99 to 54.57 ($+2.58$ pp), with AAR-TF providing further gain. Performance on BBH and IFEval is maintained within noise; HumanEval ticks down slightly at the best configuration. GSM8K gains replicate on Qwen1.5 as shown in Table 3.2, though MMLU-Math does not improve for that model.

3.4.6 Window-Size Ablation: AAR-T Applied to All Layers

We additionally report the full ablation over context window sizes $M \in \{10, 20, 40, 64, 128, 512\}$, with AAR-T applied to all layers (no layer selection), in Tables 3.3 and 3.4.

[6] `Qwen/Qwen1.5-MoE-A2.7B-Chat`

Configuration	GSM8K	BBH	MMLU	MMLU-Math	HumanEval	IFEval
OLMoE-0125-SFT	51.33	36.80	53.96	31.01	41.46	56.19
+ routing-only SFT	$51.99_{\pm 0.76}$	$36.17_{\pm 0.45}$	$53.91_{\pm 0.01}$	$30.45_{\pm 0.56}$	$41.46_{\pm 0.80}$	$56.56_{\pm 1.09}$
+ AAR-T$_S$, $M = 20$	$54.57_{\pm 1.10}$	$36.33_{\pm 0.61}$	$53.78_{\pm 0.14}$	$32.29_{\pm 0.64}$	$40.85_{\pm 1.14}$	$56.34_{\pm 1.88}$
+ AAR-TF$_S$, $M = 20$	$54.89_{\pm 1.02}$	$36.30_{\pm 0.42}$	$53.93_{\pm 0.10}$	$\mathbf{33.48}_{\pm 0.49}$	$40.98_{\pm 1.80}$	$56.67_{\pm 1.36}$
+ AAR-TF$_S$, $M = 40$	$\mathbf{55.36}_{\pm 0.74}$	$36.71_{\pm 0.46}$	$\mathbf{53.96}_{\pm 0.20}$	$32.91_{\pm 0.54}$	$40.00_{\pm 1.26}$	$55.64_{\pm 0.41}$
Qwen1.5-MoE-A2.7B-Chat						
+ routing-only SFT	$53.48_{\pm 1.45}$	$\mathbf{40.96}_{\pm 0.43}$	$\mathbf{59.22}_{\pm 0.04}$	$\mathbf{42.06}_{\pm 0.43}$	$39.26_{\pm 0.83}$	$\mathbf{36.45}_{\pm 1.26}$
+ AAR-TF$_{S_Q}$, $M = 40$	$\mathbf{55.36}_{\pm 1.41}$	$40.51_{\pm 0.36}$	$59.02_{\pm 0.13}$	$41.40_{\pm 0.41}$	$\mathbf{40.48}_{\pm 2.27}$	$35.82_{\pm 0.97}$

Table 3.2. *AAR compared against trained baselines. GSM8K consistently improves. AAR-TF achieves best results. $S = \{9, \ldots, 15\}$ for OLMoE; $S_Q = \{14, \ldots, 17\}$ for Qwen1.5. Best in bold.*

Configuration	GSM8K	ARC-C (acc)	ARC-C (norm)	ARC-E (acc)	ARC-E (norm)
Original SFT checkpoint	51.33	**46.59**	49.40	75.72	75.25
Baseline (SFT on Tulu-3 25%)	$51.99_{\pm 0.76}$	$46.08_{\pm 0.48}$	$49.30_{\pm 0.38}$	$75.71_{\pm 0.25}$	$75.60_{\pm 0.18}$
AAR-T, $M = 10$	$51.99_{\pm 1.13}$	$46.50_{\pm 0.76}$	$49.61_{\pm 0.76}$	$76.14_{\pm 0.26}$	$76.52_{\pm 0.48}$
AAR-T, $M = 20$	$\mathbf{54.10}_{\pm 0.52}$	$46.19_{\pm 0.69}$	$49.20_{\pm 0.98}$	$\mathbf{76.58}_{\pm 0.24}$	$\mathbf{76.69}_{\pm 0.35}$
AAR-T, $M = 40$	$53.18_{\pm 0.55}$	$45.85_{\pm 0.59}$	$49.22_{\pm 0.76}$	$76.30_{\pm 0.47}$	$76.46_{\pm 0.65}$
AAR-T, $M = 64$	$52.87_{\pm 1.75}$	$45.85_{\pm 0.76}$	$48.77_{\pm 0.44}$	$76.30_{\pm 0.63}$	$76.65_{\pm 0.54}$
AAR-T, $M = 128$	$53.74_{\pm 0.91}$	$45.63_{\pm 0.46}$	$49.37_{\pm 0.88}$	$76.30_{\pm 0.38}$	$76.59_{\pm 0.47}$
AAR-T, $M = 512$	$52.96_{\pm 0.76}$	$46.14_{\pm 0.56}$	$\mathbf{49.64}_{\pm 0.66}$	$76.42_{\pm 0.49}$	$76.58_{\pm 0.47}$

Table 3.3. *AAR-T (all layers) results (% accuracy) on reasoning benchmarks (averaged over 5 seeds, mean ± std). ARC-C = ARC-Challenge, ARC-E = ARC-Easy (0-shot, without a chat template).*

Configuration	MMLU	IFEval	HumanEval
Original SFT checkpoint	**53.96**	56.19	**41.46**
Baseline (SFT on Tulu-3 25%)	$53.91_{\pm 0.01}$	$\mathbf{56.56}_{\pm 1.09}$	$\mathbf{41.46}_{\pm 0.80}$
AAR-T, $M = 10$	$52.06_{\pm 0.12}$	$56.04_{\pm 1.84}$	$38.27_{\pm 0.87}$
AAR-T, $M = 20$	$52.21_{\pm 0.05}$	$55.97_{\pm 1.90}$	$40.15_{\pm 1.16}$
AAR-T, $M = 40$	$52.35_{\pm 0.15}$	$55.27_{\pm 1.32}$	$38.29_{\pm 1.33}$
AAR-T, $M = 64$	$52.13_{\pm 0.20}$	$54.71_{\pm 1.30}$	$39.41_{\pm 2.10}$
AAR-T, $M = 128$	$52.31_{\pm 0.23}$	$56.12_{\pm 0.96}$	$39.00_{\pm 1.30}$
AAR-T, $M = 512$	$52.22_{\pm 0.24}$	$56.30_{\pm 0.77}$	$39.68_{\pm 1.71}$

Table 3.4. *AAR-T (all layers) results (% accuracy) on coding, general knowledge and instruction-following benchmarks (averaged over 5 seeds, mean ± std). For HumanEval we report the pass@10 % accuracy.*

3.5 Window Size and Signal Structure

3.5.1 Intrinsic Correlation Length of the Attention Signal

We analyze the temporal structure of $\mathbf{w}_i^l$ via its unbiased normalized autocorrelation function (ACF), computed for GSM8K prompts and averaged over query tokens. For a signal $w_i^l(t)$, $t \in \{1, \ldots, M\}$, the normalized ACF at lag τ is defined as

$$R(\tau) = \frac{\dfrac{1}{M-\tau}\displaystyle\sum_{t=1}^{M-\tau}\big(w_i^l(t)-\bar{w}\big)\big(w_i^l(t+\tau)-\bar{w}\big)}{\dfrac{1}{M}\displaystyle\sum_{t=1}^{M}\big(w_i^l(t)-\bar{w}\big)^2}, \qquad \tau = 0, 1, \ldots, M-1, \tag{3.9}$$

where $\bar{w} = \frac{1}{M}\sum_{t=1}^{M} w_i^l(t)$ is the sample mean. By construction $R(0) = 1$, and $R(\tau) \to 0$ as τ grows when successive weights are uncorrelated.

3.5.2 Window Size

AAR-T peaks at $M = 20$ in the all-layers configuration (Table 3.3), consistent with the intrinsic correlation length identified above. AAR-TF marginally shifts this optimum to $M = 40$ (Table 3.2): with $K = \lfloor M/2 \rfloor + 1$ frequency bins growing with M, the frequency-domain router benefits from finer spectral resolution.

3.6 From Routing to Generation: Internal and External Effects

Routing decisions are not confined to the router itself; they determine which experts process the token, thus shaping the representations that are passed to subsequent modules, including attention. This raises the question of how AAR-induced routing changes propagate through the model and manifest in its behavior. We analyze the effects of AAR on model behavior at two levels: internally, by examining how routing changes propagate to the attention mechanism, and externally, by examining how they manifest in the model's generation. For clarity, both analyses use AAR-T, which isolates the effect of the time-domain attention signal alone.

3.6.1 Internal: AAR Reshapes Attention

To understand the low-level changes underlying the reasoning improvements, we examine whether and how attention weights differ between AAR-T and baseline models. We

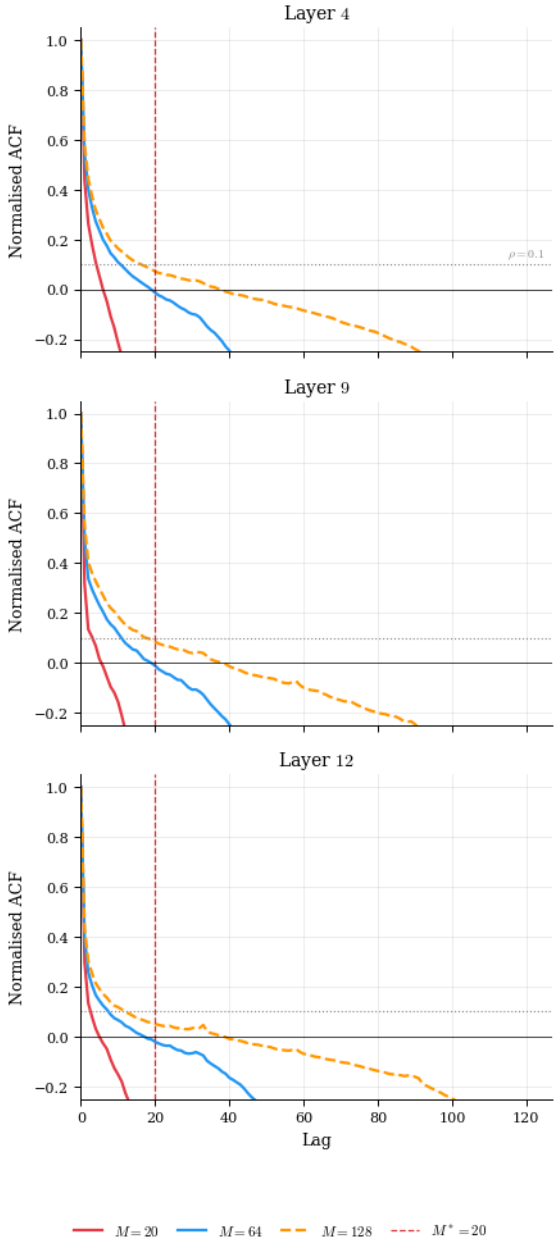


Figure 3.2. *Normalized autocorrelation function (ACF) of the attention window signal $w_i^l(t)$ at layers 4, 9, and 12, for example window sizes $\in \{20, 64, 128\}$. The ACF is averaged over query tokens and prompts from GSM8K (5-shot). The dashed vertical line marks lag $= 20$. Across all layers and window sizes, the ACF has decayed below 0.1 by lag ~ 20, revealing an intrinsic correlation length of 20 tokens in the attention signal. A window of $M = 20$ is thus sufficient to capture most structured information.*

collect attention weights generated when processing GSM8K prompts, using the AAR-T models with $M = 20$ and their corresponding baselines.

We find that AAR-T amplifies *attention sinks*, i.e., the concentration of attention mass on the first token of the sequence [35], and verify that this effect is associated with AAR via the layer-selective variant: when AAR is applied from layer l onwards, attention sinks are amplified starting precisely in layer $l + 1$ (Figure 3.3). Routing changes at layer l alter expert selection and weighting, which in turn reshapes the representations passed to layer $l + 1$'s attention submodule, **without any direct update to the attention mechanism itself**.

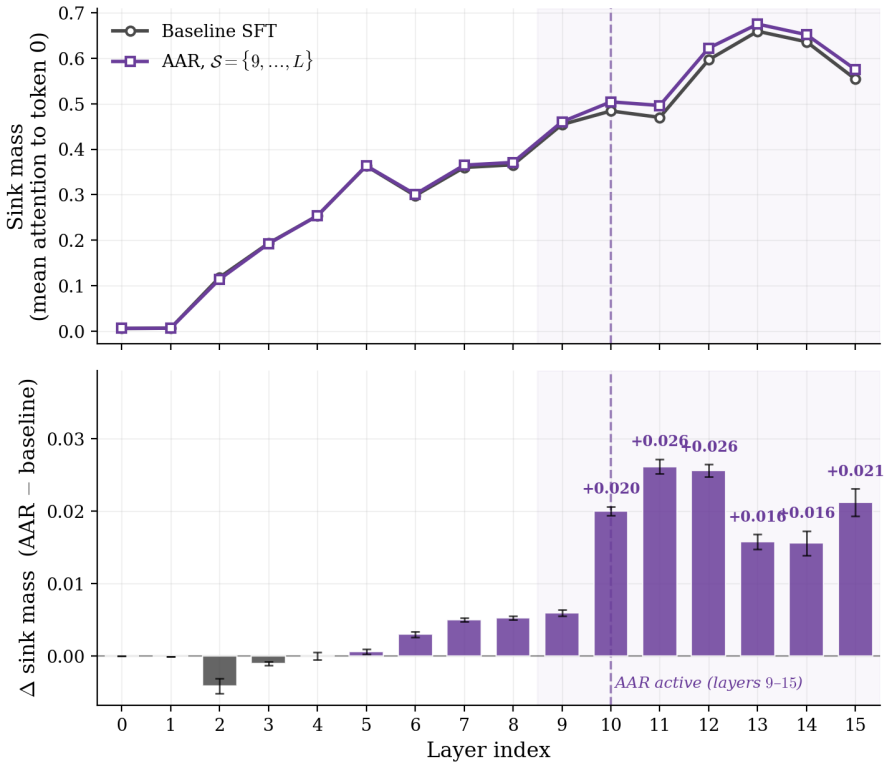


Figure 3.3. *Top: Absolute weight to the attention sink, per layer (averaged across prompt's tokens). Bottom: the absolute difference, isolated. The different routing decisions of layer 9, because of AAR, lead to different attention weights produced in the attention submodule, at layer 10.*

This suggests a mechanism consistent with how AAR may improve mathematical reasoning. Prior work has established that attention sink formation stabilizes decoding by anchoring computation to a fixed reference token [35, 48]. Under AAR, the model reinforces the pattern of concentrating attention on the first token, through routing alone. Routing thus acts not only as a selector of expert capabilities, but as an indirect regulator of attention's processing downstream.

3.6.2 External: AAR Reduces Long Diverging Generation

We compare answer lengths on GSM8K between matched pairs of AAR-T and baseline models (Table 3.5). When both models answer incorrectly, AAR-T produces substantially shorter answers (mean -11.8%, P99 -14.9%). When both models are correct, lengths are indistinguishable. The effect is concentrated in the wrong-answer regime and consistent across the full output length distribution, suggesting that AAR reduces *diverging generation length* [41, 42]: the model commits to an answer sooner when wrong, rather than generating long rambling reasoning chains. Correct answers are mostly unaffected, indicating the model has not simply learned to be uniformly brief.

Condition	Metric	Base	AAR-T	Rel. Δ
Both Correct	Mean	68.2	67.6	-0.8%
	Median	54.0	54.0	0.0%
	P99 (chars)	251.0	248.0	-1.2%
Both Wrong	Mean	163.7	144.4	-11.8%
	Median	78.0	73.0	-6.4%
	P99 (chars)	8513.0	7242.0	-14.9%
	Ramble Rate	5.63%	4.29%	-23.9%
Wrong $\rightarrow$ Correct	Mean	140.4	89.5	-36.3%
	Median	83.0	80.0	-3.6%
	P99 (chars)	7306.0	1036.0	-85.8%

Table 3.5. *Length analysis (characters) on GSM8K averaged across seeds and context sizes M. Ramble Rate denotes responses exceeding 1,000 characters. AAR-T leaves correct responses nearly unchanged while reducing the length of incorrect reasoning, especially in the tail (P99). The answer length reduction also persists for answers that get corrected by AAR-T (Wrong $\rightarrow$ Correct row).*

3.7 Complementary Experiments and Ablations

3.7.1 AAR and Load Balancing

In all previous experiments, following OLMoE's SFT training regime, we do not apply an auxiliary load-balancing loss during the routers' fine-tuning. However, load balancing is a common component of MoE training, and it is important to verify that AAR's improvements are not contingent on its absence. We thus conduct an additional experiment with the auxiliary load-balancing loss applied at a standard weight of $\lambda = 0.01$. We find that AAR-T with load balancing still improves GSM8K across window sizes when applied to all layers (Table 3.6). Thus, AAR's performance gains are not a mere artifact of greedy expert collapse, but reflect a genuine improvement in the routing mechanism's ability to leverage the attention signal for mathematical reasoning.

Configuration		GSM8K	HumanEval	MMLU
Baseline (routing-only SFT)		$51.99_{\pm 0.76}$	$41.46_{\pm 0.80}$	$53.91_{\pm 0.01}$
AAR-T, $M = 20$	w/o LB	$\mathbf{54.10}_{\pm 0.52}$	$\mathbf{40.15}_{\pm 1.16}$	$52.21_{\pm 0.05}$
	w/ LB	$53.75_{\pm 0.33}$	$39.78_{\pm 1.13}$	$52.23_{\pm 0.14}$
AAR-T, $M = 64$	w/o LB	$52.87_{\pm 1.75}$	$39.41_{\pm 2.10}$	$52.13_{\pm 0.20}$
	w/ LB	$52.98_{\pm 1.68}$	$38.80_{\pm 1.50}$	$52.11_{\pm 0.18}$
AAR-T, $M = 128$	w/o LB	$53.74_{\pm 0.91}$	$39.00_{\pm 1.30}$	$52.31_{\pm 0.23}$
	w/ LB	$53.95_{\pm 0.83}$	$38.59_{\pm 1.54}$	$\mathbf{52.34}_{\pm 0.29}$
AAR-T, $M = 512$	w/o LB	$52.96_{\pm 0.76}$	$39.68_{\pm 1.71}$	$52.22_{\pm 0.24}$
	w/ LB	$53.40_{\pm 1.78}$	$39.24_{\pm 1.38}$	$52.24_{\pm 0.24}$

Table 3.6. *Effect of auxiliary load-balancing loss (weight $\lambda = 0.01$) on AAR-T applied to all OLMoE layers. GSM8K (8-shot CoT), HumanEval (pass@10), and MMLU (5-shot). Bold marks the best AAR variant per column.*

3.7.2 Routing with Attention Statistics

While testing AAR at inference-time, we saw that simple interventions in the structure of the attention signal that kept its distributional characteristics intact, such as shuffling the order of the attention weights, did not necessarily yield complete performance degradation to baseline levels. This suggests that the routing mechanism indeed leverages distributional characteristics of the attention signal, along with the positional characteristics. To isolate the distributional effect and aggregated positional effect, we conduct an additional experiment, focused on interpretability: we replace the attention signal window with a feature vector carrying distributional characteristics: mean, 2nd moment, 3rd moment, Gini index, KL divergence from uniform, recency mass, and average centroid position. Results are reported in Table 3.7.

Configuration		GSM8K	HumanEval	MMLU
Baseline (routing-only SFT)		$51.99_{\pm 0.76}$	$41.46_{\pm 0.80}$	$53.91_{\pm 0.01}$
$M = 20$	AAR-T (window)	$\mathbf{54.10}_{\pm 0.52}$	$\mathbf{40.15}_{\pm 1.16}$	$52.21_{\pm 0.05}$
	Statistics	$53.69_{\pm 0.88}$	$39.24_{\pm 1.28}$	$\mathbf{52.32}_{\pm 0.24}$
$M = 64$	AAR-T (window)	$52.87_{\pm 1.75}$	$39.41_{\pm 2.10}$	$52.13_{\pm 0.20}$
	Statistics	$52.72_{\pm 0.84}$	$38.80_{\pm 0.82}$	$52.29_{\pm 0.14}$
$M = 128$	AAR-T (window)	$53.74_{\pm 0.91}$	$39.00_{\pm 1.30}$	$52.31_{\pm 0.23}$
	Statistics	$53.39_{\pm 1.37}$	$38.51_{\pm 1.18}$	$52.16_{\pm 0.22}$
$M = 512$	AAR-T (window)	$52.96_{\pm 0.76}$	$39.68_{\pm 1.71}$	$52.22_{\pm 0.24}$
	Statistics	$53.69_{\pm 0.60}$	$38.73_{\pm 2.06}$	$52.23_{\pm 0.18}$

Table 3.7. *AAR-T with raw attention window vs. routing conditioned on attention statistics (mean, 2nd/3rd moment, Gini index, KL divergence from uniform, recency mass, centroid position), applied to all OLMoE layers. GSM8K (8-shot CoT), HumanEval (pass@10), MMLU (5-shot). Bold marks the best AAR variant per column.*

We see that the statistics-based router captures some of the AAR-T gain on GSM8K,

improving over the baseline by $+1.70$ pp at $M = 20$, but does not reach the full window-based AAR-T performance. This shows that while distributional characteristics of the attention signal are indeed leveraged by the routing mechanism, the fine-grained positional structure of the attention weights also carries important information that is not fully captured by global statistics. At the same time, the statistics-based router consistently degrades HumanEval performance more than AAR-T, suggesting further inflexibility at generation time.

3.8 Summary of Results

AAR demonstrates that augmenting MoE routing with attention weights—a signal that is computed but then discarded in standard models—improves GSM8K performance, with minor regressions on coding and instruction-following at the best configuration. The frequency-domain extension (AAR-TF) provides further gains through spectral features of the attention window. Both improvements are achieved by training the routing parameters alone, with the base transformer remaining frozen. The underlying mechanism is a routing-attention coupled circuit: routing changes at layer l propagate through the residual stream to reshape attention at layer $l + 1$, without any direct update to the attention mechanism itself.

Chapter 4

Layer-Selective AAR as a Depth-wise Attention Probe

4.1 Motivation

The results of Chapter 3 reveal a consistent tension: attention routing improves reasoning-heavy benchmarks (GSM8K, MMLU-Math) while degrading performance on knowledge-intensive tasks, most notably MMLU overall (-1.5% or more across all context window configurations). We hypothesize that this degradation stems from interference with the *factual retrieval* mechanisms that reside in the earlier layers of the network.

This hypothesis is grounded in a growing body of evidence that transformer layers serve functionally distinct roles. Song et al. [38] demonstrate that shallow layers dominate likelihood estimation and factual retrieval, while mid-to-deep layers are essential for reasoning and generation. Similarly, work on latent reasoning modes in LLMs [39] distinguishes between factual shortcut pathways, which rely on early representations, and genuine reasoning pathways that engage deeper computation. If attention routing modifies expert selection in early layers—where the model retrieves stored knowledge—it may disrupt well-calibrated retrieval patterns without contributing meaningfully to reasoning.

4.2 Overview

Layer-selective AAR is a targeted refinement of the full-layer design described in Chapter 3. Rather than applying attention-weighted gating at every layer, we restrict the mechanism to layers at or above a designated start layer l_s:

$$\mathbf{g}_i^{(l)} = \begin{cases} W_l \mathbf{h}_i & \text{if } l < l_s, \\ \alpha_i^{(l)} \odot W_l \mathbf{h}_i + (1 - \alpha_i^{(l)}) \odot A_l \mathbf{a}_i & \text{if } l \geq l_s, \end{cases} \tag{4.1}$$

where all notation follows Chapter 3. Layers below l_s retain the original routing with no auxiliary parameters; layers at or above l_s use the full attention-aware mechanism.

This design is not merely an ablation variant: the best-performing OLMoE configuration in the main evaluation (Table 3.2) is AAR-T$_S$, which uses this design with $l_s = 9$. The sweep over start layers $l_s \in \{1, 3, 5, 7, 9, 11, 13, 15\}$ (zero-indexed) reported below is therefore both a performance study and a post-hoc explanation of *why* the selective variant outperforms full-layer AAR. All experiments use a fixed context window $M = 20$ and the training configuration of Table 3.1, repeated over 5 random seeds.

4.3 Experiments

Our experimentation involves training AAR-T$_S$ models with different start layers l_s and evaluating them on key benchmarks of interest, as in Chapter 3. The key questions we seek to answer are:

- How does the choice of start layer l_s affect performance on reasoning-heavy benchmarks versus knowledge-intensive benchmarks?
- Where should we apply attention routing to maximize reasoning gains while minimizing retrieval disruption?
- Broadly: What does the layer-wise performance pattern reveal about the functional specialization of attention in transformer layers? Namely, under AAR, which layers' attention weights carry the most useful information for routing, and how does this relate to the retrieval-reasoning distinction?

4.3.1 MMLU

Motivated by the performance degradation on MMLU observed in Chapter 3 when AAR is applied to all layers, we first do a per-subject analysis. Table 4.1 reports the absolute accuracy difference from baseline when AAR-T is applied to *all* layers, across representative MMLU subjects and window sizes M. Reasoning-oriented subjects (Math, Computer Science) are preserved or improved, while retrieval-heavy subjects (Biology, History, Philosophy) suffer consistent degradation.

We train AAR-T$_S$ models with different start layers l_s and evaluate on MMLU. The results show a clear pattern: as l_s increases, MMLU performance recovers and plateaus near baseline for $l_s \geq 7$, with the most significant gains occurring when l_s is in the mid-to-deep layer range (7–11).

The recovery of MMLU performance is not uniform across subjects. Figure 4.2 shows per-subject accuracy as a function of start layer. The subjects exhibiting the strongest recovery are those that require both factual retrieval and reasoning: Chemistry, Biology, Economics, and Physics. These are precisely the domains where early-layer disruption could be most damaging: under an oversimplified view, the model must first retrieve relevant knowledge (e.g., chemical properties, biological mechanisms) and then reason

MMLU Subject	$\Delta M{=}10$	$\Delta M{=}20$	$\Delta M{=}64$
Math	$+0.7$	$+0.6$	$+0.5$
Computer Sci.	$+0.5$	$+0.4$	-0.1
Biology	-4.5	-3.4	-3.8
History	-2.5	-2.4	-2.2
Health	-1.4	-1.8	-1.9
Psychology	-1.4	-1.7	-1.2
Geography	-1.0	$+0.2$	$+0.0$
Philosophy	-3.2	-3.1	-3.4

Table 4.1. *MMLU subcategory Δ (absolute % accuracy difference from baseline, 5 seeds) when AAR-T is applied to all layers of OLMoE. Reasoning-oriented subjects (top) are improved; degradation is concentrated in retrieval-heavy subjects (bottom).*

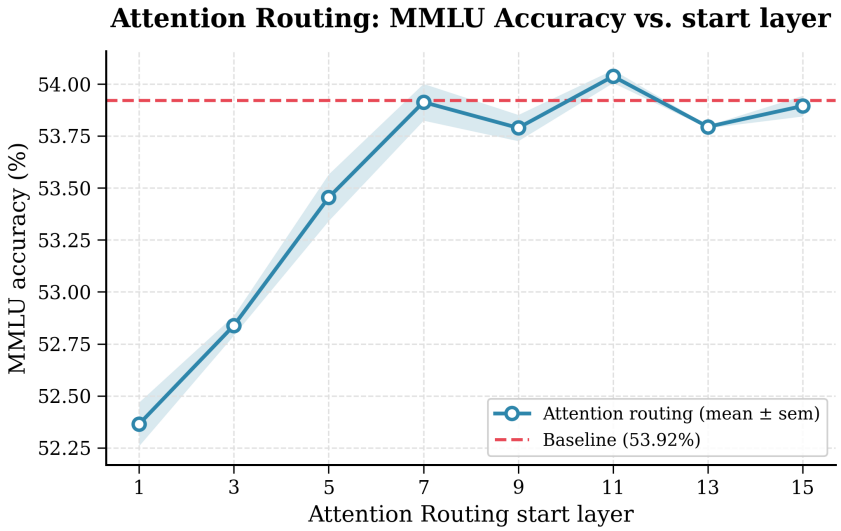


Figure 4.1. *MMLU accuracy (%) versus attention routing start layer (mean $\pm$ SEM over 5 seeds). The dashed red line indicates the routing-only SFT baseline (53.92%). Performance recovers and plateaus near baseline for $l_s \geq 7$, with a slight peak around $l_s = 9$–11.*

over it. By exempting the early layers from attention routing, we preserve the retrieval step while enhancing the reasoning step.

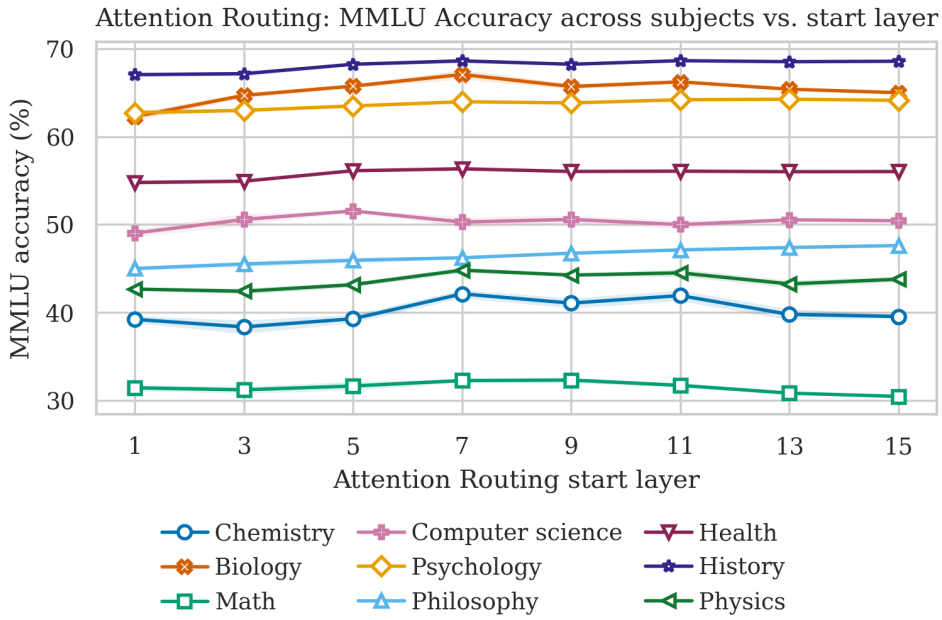


Figure 4.2. *AAR-T ($M = 20$) accuracy across MMLU subjects as a function of starting layer. Most subjects' accuracy curves follow a bell shape. Retrieval-heavy subjects (Biology, History, Philosophy) are worst when AAR starts early and recover as the starting layer increases; reasoning-heavy subjects improve regardless of starting depth; subjects combining both (Economics, Physics) peak around layers 7–11.*

Math follows a different pattern: performance peaks when routing starts from layers 7–9, then *declines* when routing is applied only to the very deepest layers ($l_s \geq 13$). This suggests that mathematical reasoning engages a broader range of layers, and restricting attention routing to only the last few layers removes useful signal.

Philosophy is a notable outlier, showing smooth, monotonic improvement as l_s increases. This can be attributed to a variety of factors, including the subject's potentially heavy reliance on retrieval of specific concepts and the robustness of AAR training, in general.

4.3.2 ARC-Challenge and ARC-Easy

We evaluate on ARC –a benchmark that combines reasoning and retrieval capabilities– to assess layer-wise performance of AAR, as well as robustness. In this section, we consider 0-shot prompts with a chat template.

The ARC results present a more nuanced picture than MMLU. For ARC-Easy, AAR-trained models surpass the baseline, when AAR is applied early. Performance decreases as the AAR start layer gets deeper. For ARC-Challenge, the pattern is monotonic: early startlayers 5–11 yield severe performance degradation, which then recovers at deeper

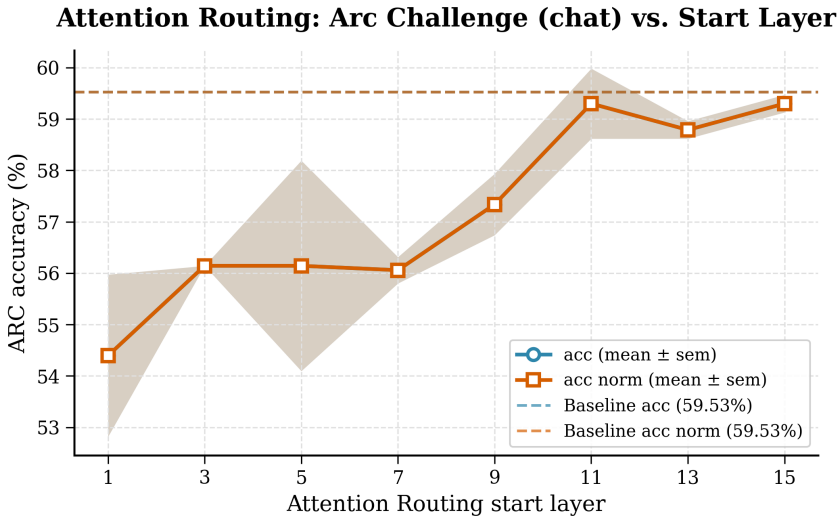


Figure 4.3. *ARC-Challenge accuracy (0-shot, with chat template, log-likelihood) versus start layer. The blue line shows raw accuracy; the orange line shows normalized accuracy.*

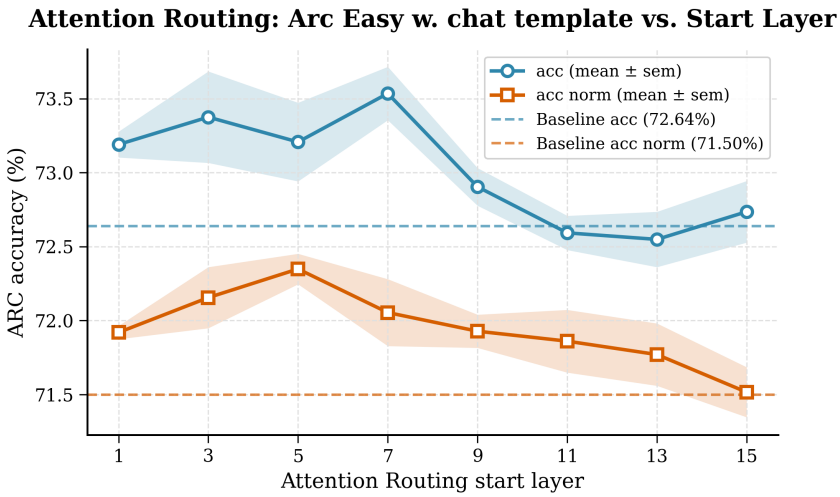


Figure 4.4. *ARC-Easy accuracy (0-shot, with chat template, log-likelihood) versus start layer.*

start layers. The conclusions we can draw with this experiment are 2-sided and task-dependent. In the case of the much harder ARC-Challenge, the results are consistent with the MMLU pattern: early-layer routing disrupts retrieval and harms performance, while deeper-layer routing preserves retrieval and recovers performance. In the case of ARC-Easy, the results suggest that the reasoning boost from AAR can actually enhance performance on easier questions, but only if applied from early layers. ARC-Easy follows a pattern more similar to GSM8K, where the reasoning boost is more important than retrieval preservation, while ARC-Challenge follows a pattern more similar to MMLU, where retrieval preservation appears crucial.

4.4 Discussion

The layer-selective experiments yield three main insights.

Functional specialization of layers. The start-layer sweep serves as an implicit ablation study on *where* in the network attention patterns carry information useful for routing. The monotonic recovery of MMLU with increasing l_s confirms that shallow layers handle retrieval and pattern matching, while mid-to-deep layers perform compositional reasoning. This is consistent with prior work [38] and extends it to the MoE routing setting.

The sweet spot. Layers 7–11 (zero-indexed) represent the optimal range to start attention routing. Within this range, MMLU is recovered to or above baseline ($53.92\% \rightarrow\approx 54.0\%$), and GSM8K retains most of its improvement ($\approx$ 53.5–53.8% vs. 54.10% with full routing and 51.99% baseline). The configuration $l_s = 9$ corresponds directly to the AAR-T$_S$ model reported as the best OLMoE variant in Table 3.2.

Task-dependent trade-offs. The optimal start layer depends on the application. For reasoning-first settings, $l_s = 7$–9 preserves the strongest GSM8K gains while recovering MMLU. For settings where factual accuracy is paramount, $l_s = 11$ or higher recovers nearly all baseline MMLU performance, at the cost of a slightly reduced reasoning boost.

Layer-selective AAR as a depth-wise probe. Beyond the performance gains, this design yields a depth-wise attention probe as a byproduct: by varying only where AAR is applied, we can read off what kind of routing-relevant information each layer's attention weights carry. Per-layer attention weights thus act as a fingerprint of each layer's information processing. The probe reveals where the retrieval-reasoning tension is concentrated, and that it is readable from the attention signal. This is consistent with and extends Song et al. [38], who show that shallow layers dominate retrieval tasks

while mid-to-deep layers are essential for reasoning. Rather than probing representations post-hoc, layer-selective AAR probes the functional role of attention weights by measuring the downstream effect of routing changes directly—without the use of auxiliary classifiers.

Chapter 5

Discussion

Self-regulation through routing. By providing the router with a fingerprint of the model's contextual processing, as encoded in the attention weights, we enable a form of self-regulation: the routers can settle into configurations under which downstream attention patterns shift in ways that co-occur with improved task performance, making this coupling directly available during optimization. The effect is reflected internally in the reshaping of attention (Section 3), and externally in the reduced length of wrong answers. It is not an explicitly supervised signal; it emerges from training with standard next-token prediction on the routing parameters alone.

From routing to generation. Our results trace a cascade: routing at layer l reshapes the representations passed to layer $l + 1$, which alters that layer's attention weights (evidenced by the one-layer-delayed sink amplification), and is accompanied by shorter generation on incorrect responses. That a change confined entirely to routing parameters can produce measurable effects at the level of generation behavior suggests that routing, attention, and generation are tightly coupled, and that routing is a consequential component of how the model processes information, beyond its role as a dispatching mechanism.

Retrieval-reasoning tension. The layer-selective experiments reveal that the attention-routing circuit carries qualitatively different information at different depths. Early layers are critical for factual lookup and sensitive to AAR-driven changes; beyond the middle layers, retrieval-relevant structure has consolidated and AAR can be applied without disrupting it. This is consistent with and extends Song et al. [38], and suggests that a single uniform intervention across all layers is sub-optimal for tasks that combine retrieval and reasoning.

Routing as a degree of freedom. Taken together, the results reinforce the view that routing is not merely a dispatching mechanism but a consequential degree of freedom in language models. Shifting routing from a local, per-token decision to one informed by the model's own attention patterns yields better reasoning performance and a more interpretable interface into the model's internal computation.

Chapter 6

Closing Remarks

6.1 Conclusion

Standard MoE routers select experts based on each token's hidden state alone, without direct access to the attention patterns that shaped it. We show that augmenting the router with a sliding window of attention weights can meaningfully reshape model behavior through routing alone, improving GSM8K performance. The frequency-domain extension provides further gains through spectral features of the same window. These improvements come with modest regressions on HumanEval and IFEval, reflecting the retrieval-reasoning trade-off discussed in Chapter 4. These routing changes also propagate through the network: they amplify attention sinks one layer downstream, and at generation time they reduce output length on incorrect answers while leaving correct ones unchanged. The retrieval-reasoning tension we observe demonstrates that the attention-routing circuit carries qualitatively different information at different depths, with layer-selective AAR serving as a natural probe into this structure. Taken together, our results reinforce the view that routing is not merely a dispatching mechanism, but a consequential degree of freedom in language models.

6.2 Limitations

Efficiency. AAR requires explicit attention weight materialization at each AAR-active layer, which precludes the use of FlashAttention [23]. The resulting eager-attention fallback introduces memory and compute overhead that scales linearly with sequence length and the number of active layers.

Training regime. Our experiments focus on post-SFT adaptation: auxiliary routers are trained jointly with already-initialized routing parameters, while the base transformer remains frozen. This asymmetric setup has two consequences. First, the optimal set of AAR-active layers is entangled with the model's prior training history, which limits turnkey cross-model generalization — the best layer ranges for OLMoE and Qwen1.5 both concentrate in middle-to-deep layers, but differ in their exact span. Second, training

only a small number of new parameters on top of a fixed model introduces performance variance that full-parameter joint training from scratch would likely reduce.

6.3 Future Work

Pretraining integration. Integrating AAR during pretraining rather than as a post-SFT adaptation would allow the model and its routers to co-evolve from the outset, potentially unlocking stronger routing-attention coupling and alleviating the layer-range dependence on prior training.

Richer spectral features. AAR-TF demonstrates that simple spectral features of the attention window carry complementary routing signal beyond what the raw time-domain window provides. Richer signal transformations — learned filterbanks, wavelet decompositions, or jointly trained feature extractors — may yield further gains, particularly for tasks where the relevant attention structure varies across layers or prompt types.

Adaptive layer selection. The layer-selective design currently relies on a manually swept layer subset S. A learned or input-conditioned mechanism for selecting S would make AAR more broadly applicable without per-model tuning. The spectral flatness heuristic of Appendix C offers a data-driven proxy for choosing S, and could serve as initialization for such a learned selector.

AAR in masked and diffusion language models. AAR was developed entirely within autoregressive, decoder-only MoE models, but its premise — that the router benefits from conditioning on the contextual state summarized by attention — is not specific to that setting. Masked and diffusion language models expose a bidirectional attention structure that, in the diffusion case, evolves across denoising steps. Porting AAR into these regimes, where a router could additionally condition on the denoising step or local masking density, is a natural and largely unexplored direction; it would also test whether the attention-routing circuit of Chapter 3, including the downstream amplification of attention sinks, has an analogue under bidirectional attention. This connects directly to concurrent work by the author on frequency-informed masked diffusion language modeling (Appendix D).

Routing as a steering mechanism. The routing-attention circuit identified in this work suggests that routing can serve as a targeted intervention point for shaping model behavior without modifying the attention mechanism itself. Exploring this perspective in the context of controlled generation, safety, or alignment is a promising direction: routing modifications are lightweight, interpretable, and localized to specific layers, making them a natural candidate for fine-grained behavioral control.

References

[1] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[2] Jason Wei et al. “Emergent Abilities of Large Language Models”. In: *Transactions on Machine Learning Research* (2022). Survey Certification. ISSN: 2835-8856. URL: https://openreview.net/forum?id=yzkSU5zdwD.

[3] Noam Shazeer et al. “Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer”. In: *ICLR*. 2017.

[4] William Fedus, Barret Zoph, and Noam Shazeer. “Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity”. In: *JMLR* (2022).

[5] Albert Q. Jiang et al. *Mixtral of Experts*. 2024. arXiv: 2401.04088 [cs.LG]. URL: https://arxiv.org/abs/2401.04088.

[6] Niklas Muennighoff et al. “OLMoE: Open Mixture-of-Experts Language Models”. In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=xXTkbTBmqq.

[7] Qwen Team. *Qwen1.5-MoE: Matching 7B Model Performance with 1/3 Activated Parameters*. Feb. 2024. URL: https://qwenlm.github.io/blog/qwen-moe/.

[8] Yoshua Bengio et al. “A neural probabilistic language model”. In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 1137–1155. ISSN: 1532-4435.

[9] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

[10] Ilya Loshchilov and Frank Hutter. “Decoupled Weight Decay Regularization”. In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[11] Alec Radford and Karthik Narasimhan. “Improving Language Understanding by Generative Pre-Training”. In: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[12] Alec Radford et al. “Language Models are Unsupervised Multitask Learners”. In: 2019. URL: https://api.semanticscholar.org/CorpusID:160025533.

[13] Jason Wei et al. “Finetuned Language Models are Zero-Shot Learners”. In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=gEZrGCozdqR.

[14] Long Ouyang et al. “Training language models to follow instructions with human feedback”. In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh et al. 2022. URL: https://openreview.net/forum?id=TG8KACxEON.

[15] Dirk Groeneveld et al. “OLMo: Accelerating the Science of Language Models”. In: *ACL (1)*. 2024, pp. 15789–15809. URL: https://doi.org/10.18653/v1/2024.acl-long.841.

[16] Nathan Lambert et al. *Tulu 3: Pushing Frontiers in Open Language Model Post-Training*. 2025. arXiv: 2411.15124 [cs.CL]. URL: https://arxiv.org/abs/2411.15124.

[17] Tom B. Brown et al. “Language Models are Few-Shot Learners”. In: *NeurIPS*. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142Abstract.html.

[18] Karl Cobbe et al. “Training Verifiers to Solve Math Word Problems”. In: *arXiv preprint arXiv:2110.14168* (2021).

[19] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: 2302.13971 [cs.CL]. URL: https://arxiv.org/abs/2302.13971.

[20] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL]. URL: https://arxiv.org/abs/2307.09288.

[21] Rico Sennrich, Barry Haddow, and Alexandra Birch. “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Katrin Erk and Noah A. Smith. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162. URL: https://aclanthology.org/P16-1162/.

[22] Taku Kudo and John Richardson. “SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Ed. by Eduardo Blanco and Wei Lu. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. DOI: 10.18653/v1/D18-2012. URL: https://aclanthology.org/D18-2012/.

[23] Tri Dao et al. “FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness”. In: *Advances in Neural Information Processing Systems*. 2022. URL: https://openreview.net/forum?id=H4DqfPSibmx.

[24] Dan Hendrycks and Kevin Gimpel. *Gaussian Error Linear Units (GELUs)*. 2023. arXiv: 1606.08415 [cs.LG]. URL: https://arxiv.org/abs/1606.08415.

[25] Noam Shazeer. *GLU Variants Improve Transformer*. 2020. arXiv: 2002.05202 [cs.LG]. URL: https://arxiv.org/abs/2002.05202.

[26] Jianlin Su et al. *RoFormer: Enhanced Transformer with Rotary Position Embedding*. 2023. arXiv: 2104.09864 [cs.CL]. URL: https://arxiv.org/abs/2104.09864.

[27] Damai Dai et al. *DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models*. 2024. arXiv: 2401.06066 [cs.CL]. URL: https://arxiv.org/abs/2401.06066.

[28] Dmitry Lepikhin et al. "{GS}hard: Scaling Giant Models with Conditional Computation and Automatic Sharding". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=qrwe7XHTmYb.

[29] Yanqi Zhou et al. "Mixture-of-Experts with Expert Choice Routing". In: *Advances in Neural Information Processing Systems*. 2022. URL: https://openreview.net/forum?id=jdJo1HIVinI.

[30] Barret Zoph et al. *ST-MoE: Designing Stable and Transferable Sparse Expert Models*. 2022. arXiv: 2202.08906 [cs.CL]. URL: https://arxiv.org/abs/2202.08906.

[31] Nan Du et al. "GLaM: Efficient Scaling of Language Models with Mixture-of-Experts". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 5547–5569. URL: https://proceedings.mlr.press/v162/du22c.html.

[32] Nishanth Dikkala et al. "On the Benefits of Learning to Route in Mixture-of-Experts Models". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 9376–9396. DOI: 10.18653/v1/2023.emnlp-main.583. URL: https://aclanthology.org/2023.emnlp-main.583/.

[33] Zihan Qiu et al. "Layerwise Recurrent Router for Mixture-of-Experts". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=eWNEqdH0vk.

[34] Truong Giang Do et al. "HyperRouter: Towards Efficient Training and Inference of Sparse Mixture of Experts". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 5754–5765. DOI: 10.18653/v1/2023.emnlp-main.351. URL: https://aclanthology.org/2023.emnlp-main.351/.

[35] Guangxuan Xiao et al. “Efficient Streaming Language Models with Attention Sinks”. In: *The Twelfth International Conference on Learning Representations*. 2024. URL: https://openreview.net/forum?id=NG7sS51zVF.

[36] Federico Barbero et al. “Why do LLMs attend to the first token?” In: *Second Conference on Language Modeling*. 2025. URL: https://openreview.net/forum?id=tu4dFUsW5z.

[37] Mutian Yang, Jiandong Gao, and Ji Wu. “Decoupling Knowledge and Reasoning in LLMs: An Exploration Using Cognitive Dual-System Theory”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 40.40 (2026), pp. 34268–34276. DOI: 10.1609/aaai.v40i40.40723. URL: https://ojs.aaai.org/index.php/AAAI/article/view/40723.

[38] Xinyuan Song et al. *Demystifying the Roles of LLM Layers in Retrieval, Knowledge, and Reasoning*. 2026. arXiv: 2510.02091 [cs.AI]. URL: https://arxiv.org/abs/2510.02091.

[39] Yiran Yang et al. “Unveiling Internal Reasoning Modes in LLMs: A Deep Dive into Latent Reasoning vs. Factual Shortcuts with Attribute Rate Ratio”. In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Ed. by Christos Christodoulopoulos et al. Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 2186–2206. ISBN: 979-8-89176-332-6. DOI: 10.18653/v1/2025.emnlp-main.111. URL: https://aclanthology.org/2025.emnlp-main.111/.

[40] Phuong Minh Nguyen, Dang Huu-Tien, and Naoya Inoue. “Improving Chain-of-Thought for Logical Reasoning via Attention-Aware Intervention”. In: *Findings of the Association for Computational Linguistics: EACL 2026*. Rabat, Morocco: Association for Computational Linguistics, Mar. 2026, pp. 2917–2941. DOI: 10.18653/v1/2026.findings-eacl.152. URL: https://aclanthology.org/2026.findings-eacl.152/.

[41] Xueyan Li et al. “Sample Smart, Not Hard: Correctness-First Decoding for Better Reasoning in LLMs”. In: *The Fourteenth International Conference on Learning Representations*. 2026. URL: https://openreview.net/forum?id=pNwCWIBHBC.

[42] Jinyan Su et al. *Between Underthinking and Overthinking: An Empirical Study of Reasoning Length and correctness in LLMs*. 2025. arXiv: 2505.00127 [cs.CL]. URL: https://arxiv.org/abs/2505.00127.

[43] Dan Hendrycks et al. “Measuring Massive Multitask Language Understanding”. In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=d7KBjmI3GmQ.

[44] Mark Chen et al. “Evaluating Large Language Models Trained on Code”. In: (2021). arXiv: 2107.03374 [cs.LG].

[45] Jeffrey Zhou et al. *Instruction-Following Evaluation for Large Language Models*. 2023. arXiv: 2311.07911 [cs.CL]. URL: https://arxiv.org/abs/2311.07911.

[46] Mirac Suzgun et al. "Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them". In: *ACL (Findings)*. 2023, pp. 13003–13051. URL: https://doi.org/10.18653/v1/2023.findings-acl.824.

[47] Peter Clark et al. "Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge". In: *arXiv:1803.05457v1* (2018).

[48] Xu Liu, Guikun Chen, and Wenguan Wang. "SinkTrack: Attention Sink based Context Anchoring for Large Language Models". In: *The Fourteenth International Conference on Learning Representations*. 2026. URL: https://openreview.net/forum?id=Gg1aPETCL6.

[49] Leo Gao et al. *The Language Model Evaluation Harness*. Version v0.4.3. July 2024. DOI: 10.5281/zenodo.12608602. URL: https://zenodo.org/records/12608602.

[50] Despoina Kosmopoulou et al. "Masked Diffusion Language Models with Frequency-Informed Training". In: *Proceedings of the First BabyLM Workshop*. Ed. by Lucas Charpentier et al. Suzhou, China: Association for Computational Linguistics, Nov. 2025, pp. 531–539. DOI: 10.18653/v1/2025.babylm-main.38. URL: https://aclanthology.org/2025.babylm-main.38/.

[51] Subham Sekhar Sahoo et al. *Simple and Effective Masked Diffusion Language Models*. 2024. arXiv: 2406.07524 [cs.CL]. URL: https://arxiv.org/abs/2406.07524.

Appendix A

Evaluation Details

A.1 GSM8K

GSM8K [18] is a benchmark of 1317 grade school math word problems (`dev` set) requiring multi-step arithmetic reasoning. We use the implementation in `open-instruct`[1], with 8-shot chain-of-thought prompting. We apply the Tulu-3 chat template, which structures each conversation as:

<s><|system|>
system message
<|user|>
user message
<|assistant|>
assistant response</s>
<|user|>
next user message
<|assistant|>

where <s> and </s> denote the beginning and end of sequence tokens, and the final <|assistant|> prompt signals the model to generate its response. We use use the native implementation in `open-instruct`. We have verified that the `lm-eval-harness` implementation (task: `gsm8k_cot`) preserves, or enhances, the relative differences between the baseline and our models.

A.2 HumanEval

HumanEval [44] is a code generation benchmark consisting of 164 hand-written Python programming problems, each with a function signature, docstring, and unit tests. We use the native implementation in `open-instruct`, with the same chat template as above. We set the unbiased sampling size to 10 and report `pass@10`. We repeat the evaluation as denoted in the main text.

[1] https://github.com/allenai/open-instruct

A.3 BBH

BIG-Bench Hard [46] is a suite of 23 challenging reasoning tasks drawn from the broader BIG-Bench benchmark, specifically selected as tasks where prior language models had failed to surpass average human performance. We use the same chat template as above.

A.4 IFEval

IFEval [45] is a benchmark that evaluates instruction-following ability through verifiable constraints, such as response length, keyword inclusion, or formatting requirements. We use the same chat template as above.

A.5 MMLU

Massive Multitask Language Understanding (MMLU) [43] evaluates a model's knowledge and reasoning across 57 academic subjects spanning STEM, humanities, social sciences, and professional domains, at difficulty levels ranging from elementary to professional. We use 5-shot evaluation without a chat template. The full list of subjects is as follows:

- Abstract Algebra
- Anatomy
- Astronomy
- Business Ethics
- Clinical Knowledge
- College Biology
- College Chemistry
- College Computer Science
- College Mathematics
- College Medicine
- College Physics
- Computer Security
- Conceptual Physics
- Econometrics
- Electrical Engineering
- Elementary Mathematics
- Formal Logic
- Global Facts
- High School Biology
- High School Chemistry
- High School Computer Science
- High School European History
- High School Geography
- High School Government and Politics
- High School Macroeconomics
- High School Mathematics
- High School Microeconomics
- High School Physics
- High School Psychology
- High School Statistics
- High School US History
- High School World History
- Human Aging
- Human Sexuality
- International Law
- Jurisprudence
- Logical Fallacies
- Machine Learning
- Management

- Marketing
- Medical Genetics
- Miscellaneous
- Moral Disputes
- Moral Scenarios
- Nutrition
- Philosophy
- Prehistory
- Professional Accounting
- Professional Law
- Professional Medicine
- Professional Psychology
- Public Relations
- Security Studies
- Sociology
- US Foreign Policy
- Virology
- World Religions

A.6 ARC

The AI2 Reasoning Challenge (ARC) [47] is a multiple-choice question-answering benchmark drawn from science exams. It is split into an Easy and a Challenge partition, where the latter contains questions that require more complex reasoning and cannot be answered by simple retrieval methods. We report both accuracy and normalized accuracy. For ARC, we use EleutherAI's evaluation harness[2] [49] with the same chat template as above.

[2]https://github.com/EleutherAI/lm-evaluation-harness

Appendix B

Training with Different Data

In all experiments, we report performance after SFT on a Tulu-3 25% sample. Our experiments suggest a preference of AAR for mathematical reasoning tasks. We now change the SFT data mix and train separately on two alternative datasets—CodeFeedBack and MetaMath—to examine whether AAR's gains persist under different data regimes and to probe which part of the training signal drives the improvement. For code, mirroring Chapter 3.7.1, we include ablation tables with and without load balancing, different attention window sizes, as well as the attention-statistics router. For math, we focus on the best-performing AAR-T with $M = 20$.

B.1 CodeFeedBack

CodeFeedBack[1] is a code-focused instruction dataset that is *already fully included* in the Tulu-3 SFT mixture used to train the OLMoE-0125-SFT checkpoint. Training AAR on CodeFeedBack alone therefore measures whether attention-aware routing can still extract a useful signal from a data distribution that the backbone has already seen during its original post-training, and where code generation is the dominant task.

Tables B.1 and B.2 report results for, respectively, the raw attention-window router (AAR-T) and the attention-statistics router, each with and without an auxiliary load-balancing loss. Because CodeFeedBack is code-centric, MMLU performance drops substantially relative to the Tulu-3 baseline for all AAR variants; GSM8K improvements are smaller but still present.

B.2 MetaMath

MetaMath[2] is a math-only dataset, part of which is included in Tulu3. Training AAR exclusively on MetaMath isolates its effect in a maximally math-focused regime, at the cost of catastrophic forgetting on general knowledge tasks. At the same time, even the baseline's performance on GSM8K substantially degrades, hinting at routing collapse.

[1] https://huggingface.co/datasets/m-a-p/CodeFeedback-Filtered-Instruction
[2] https://huggingface.co/datasets/meta-math/MetaMathQA

Configuration		GSM8K	HumanEval	MMLU
Baseline (SFT on CodeFeedBack)		$50.93_{\pm 0.24}$	$40.88_{\pm 0.84}$	$53.92_{\pm 0.11}$
Baseline (SFT on CodeFeedBack) + LB		$51.69_{\pm 0.38}$	$\mathbf{40.90}_{\pm 0.52}$	$\mathbf{53.94}_{\pm 0.05}$
AAR-T, $M = 10$	w/o LB	$51.63_{\pm 0.83}$	$37.54_{\pm 1.99}$	$50.18_{\pm 0.32}$
	w/ LB	$51.80_{\pm 0.75}$	$37.98_{\pm 1.50}$	$50.18_{\pm 0.31}$
AAR-T, $M = 20$	w/o LB	$51.55_{\pm 1.06}$	$37.80_{\pm 1.54}$	$50.16_{\pm 0.35}$
	w/ LB	$51.39_{\pm 1.08}$	$38.32_{\pm 0.72}$	$50.14_{\pm 0.23}$
AAR-T, $M = 64$	w/o LB	$52.13_{\pm 0.65}$	$38.34_{\pm 0.74}$	$50.02_{\pm 0.16}$
	w/ LB	$\mathbf{52.39}_{\pm 0.84}$	$\mathbf{38.61}_{\pm 1.37}$	$49.94_{\pm 0.16}$
AAR-T, $M = 128$	w/o LB	$51.40_{\pm 0.42}$	$37.73_{\pm 1.57}$	$\mathbf{50.22}_{\pm 0.18}$
	w/ LB	$51.86_{\pm 0.73}$	$38.41_{\pm 1.96}$	$50.13_{\pm 0.15}$
AAR-T, $M = 512$	w/o LB	$51.16_{\pm 1.54}$	$37.00_{\pm 1.15}$	$50.04_{\pm 0.18}$
	w/ LB	$51.16_{\pm 1.05}$	$36.93_{\pm 0.65}$	$50.15_{\pm 0.22}$

Table B.1. *AAR-T (raw attention window, all layers) trained on CodeFeedBack. GSM8K (8-shot CoT), HumanEval (pass@10), MMLU (5-shot). Bold marks the best value per column.*

Configuration		GSM8K	HumanEval
Baseline (SFT on CodeFeedBack)		$50.93_{\pm 0.24}$	$40.88_{\pm 0.84}$
Baseline (SFT on CodeFeedBack) + LB		$51.69_{\pm 0.38}$	$\mathbf{40.90}_{\pm 0.52}$
Statistics, $M = 10$	w/o LB	$52.62_{\pm 1.16}$	$38.29_{\pm 1.98}$
	w/ LB	$\mathbf{52.86}_{\pm 1.14}$	$\mathbf{38.61}_{\pm 1.53}$
Statistics, $M = 20$	w/o LB	$51.81_{\pm 0.76}$	$38.05_{\pm 2.11}$
	w/ LB	$51.61_{\pm 1.41}$	$37.51_{\pm 1.98}$
Statistics, $M = 64$	w/o LB	$51.37_{\pm 1.44}$	$37.27_{\pm 1.09}$
	w/ LB	$51.61_{\pm 1.10}$	$37.05_{\pm 1.29}$
Statistics, $M = 128$	w/o LB	$52.04_{\pm 1.06}$	$36.05_{\pm 3.46}$
	w/ LB	$52.51_{\pm 0.99}$	$37.93_{\pm 1.39}$
Statistics, $M = 512$	w/o LB	$52.07_{\pm 1.53}$	$37.61_{\pm 1.02}$
	w/ LB	$52.46_{\pm 0.26}$	$37.63_{\pm 1.60}$

Table B.2. *Attention-statistics router (mean, 2nd/3rd moment, Gini index, KL divergence from uniform, recency mass, centroid position; all layers) trained on CodeFeedBack. Bold marks the best value per column.*

Table B.3 reports results for the baseline (SFT on MetaMath) and AAR-T with $M = 20$ applied to all layers. We can draw two main conclusions:

- Inconsistent with the Tulu-3 SFT results, Math-only training yields no improvement on GSM8k relative to an untrained baseline. This is not an artifact of AAR, or AAR training, as the baseline SFT model severely degrades relative to the untrained baseline. This showcases that the GSM8K improvement from AAR-T in the Tulu-3 SFT setting is not simply due to better math performance, but rather to a more complex interaction between the math training signal and the model's existing capabilities, and it cannot be attributed to overfitting on math data alone.
- Code data outperform Math-only data on mathematical tasks (tables B.1 and B.2). Broadly, multi-task training data appear important for AAR to learn a routing strategy that generalizes well across tasks, and the math signal alone is insufficient to drive the learning of a routing strategy that benefits GSM8K.

Configuration	GSM8K	MMLU
Baseline (SFT on MetaMath)	$47.79_{\pm 0.71}$	$\mathbf{53.74}_{\pm 0.12}$
AAR-T, $M = 20$ (all layers)	$\mathbf{50.33}_{\pm 1.65}$	$48.98_{\pm 0.15}$

Table B.3. *AAR-T trained on MetaMath only. GSM8K (8-shot CoT), MMLU (5-shot). AAR yields a large GSM8K gain ($+2.54$ pp) but at the cost of a severe MMLU drop (-4.76 pp) relative to the MetaMath baseline, consistent with catastrophic forgetting on non-math tasks. The GSM8K gains are over the severely degraded performance of the SFT baseline on GSM8K. Bold marks the best value per column.*

Appendix C

Frequency-Domain Analysis of Attention Routing

This appendix provides the theoretical and empirical background for the frequency-domain router (AAR-TF) introduced in Section 3.3.3. Section C.1 motivates the use of DFT magnitudes over raw time-domain windows. Section C.2 presents an empirical analysis of how spectral structure varies across transformer layers.

C.1 Shift-Invariance and the Case for Spectral Features

The rolling attention window $\mathbf{w}_i^l \in \mathbb{R}^M$ has several properties that complicate routing when it is fed directly to a linear layer:

- It is *position-agnostic*: the window slides over tokens without encoding the absolute position of the current token in the sequence.
- It may contain *structural zeros*: at the beginning of a sequence, positions outside the context history are zero-padded.
- It is a *sequence*: the positional structure of attention weights within the window may carry information, so treating it as a bag of weights would discard that structure.

The shift-invariance problem. Consider the task of detecting whether the window contains an *attention sink* — a single token capturing a disproportionately large fraction of attention mass, which correlates with improved reasoning (Section 3). A localized attention sink can appear anywhere within the window depending on the current token position. A linear layer on the raw window $\mathbf{w}_i^l$ cannot reliably detect such a pattern at an arbitrary offset: it would need to draw a separating hyperplane that outputs a high activation for any single-spike pattern, regardless of its position, while suppressing flat or multi-modal distributions. This is not achievable by a single linear map.

FFT magnitude as a shift-invariant representation. The magnitude of the Discrete Fourier Transform (DFT) is inherently shift-invariant: if $\mathbf{x}' = \text{shift}(\mathbf{x}, k)$ is a cyclic shift

of $\mathbf{x}$ by k positions, then $|\text{DFT}(\mathbf{x}')| = |\text{DFT}(\mathbf{x})|$. This means that a localized attention pattern yields the same frequency signature regardless of where it falls in the window. A linear layer on $|\text{DFT}(\mathbf{w}_i^l)|$ can therefore detect structured patterns (e.g., an isolated sink, a periodic attention rhythm) anywhere in the window with a simple learned linear combination over frequency bins.

Concretely, a single attention sink approximates a discrete impulse $\delta[n - n_0]$, whose DFT magnitude is flat ($|\hat{X}[k]| = 1$ for all k) — a uniquely identifiable signature. Distributed attention produces destructive interference across bins, yielding a non-flat magnitude profile. The two cases become linearly separable in the frequency domain, but not in the time domain.

Trade-offs. The DFT magnitude discards absolute positional information. A router that needs to distinguish "the sink is at the start of the window" from "the sink is at the end" cannot do so from $|\text{DFT}(\mathbf{w}_i^l)|$ alone. AAR-TF retains both representations: the frequency-domain router $\tilde{A}_l$ operates on the DFT magnitude, while the time-domain router A_l retains positional sensitivity. The two are combined through the interpolation mechanism in Section 3.3.3, so the joint model can leverage both shift-invariant structure detection and positional context.

C.2 Layer-wise Spectral Structure

Setup. To characterize how spectral content varies across depth, we pass 50 GSM8K prompts through OLMoE and Qwen1.5-MoE and, for every generated token, collect the attention weights to the $M = 100$ most recent tokens at each layer. We then compute two statistics per layer:

- **Spectral flatness**: the ratio of the geometric mean to the arithmetic mean of the DFT power spectrum, $|\text{DFT}(\mathbf{w}_i^l)|^2$. A perfectly flat spectrum (value $= 1$) indicates a noise-like, unstructured signal; lower flatness indicates energy concentrated in a small number of frequency bins, i.e., a more structured periodic or impulsive signal.
- **Shannon entropy** of the raw attention weight sequence, as a complementary measure of how concentrated or diffuse the attention distribution is.

U-shaped spectral flatness. Figure C.1 shows both metrics across OLMoE layers. Spectral flatness exhibits a clear U-shape: it is relatively high (flatter, less structured) in the earliest and deepest layers, and dips to its minimum in the middle layers (roughly layers 4–10). Entropy does not mirror this pattern exactly, suggesting that spectral flatness and entropy carry partially complementary information about the attention signal.

The same U-shaped pattern appears in Qwen1.5-MoE (Figure C.2), confirming that this is not an artifact of OLMoE's architecture. DeepSeek-V2-Lite is a notable exception:

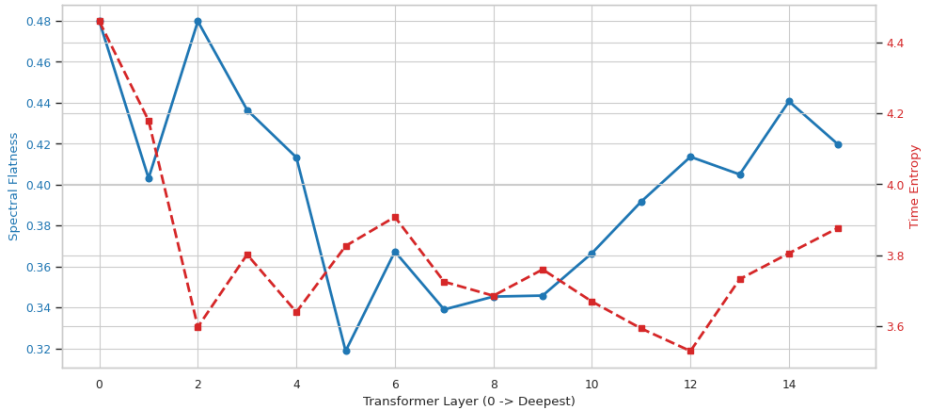


Figure C.1. *Layer-wise spectral flatness and Shannon entropy of attention windows in OLMoE. Spectral flatness follows a U-shape: middle layers (∼4–10) exhibit the least flat, most structured frequency content, while the earliest and deepest layers are flatter. The dashed red line (entropy, right axis) does not follow the exact inverse, indicating complementary information.*

its least-flat region spans roughly layers 0–15 rather than the middle third, consistent with its different depth distribution and suggesting that the optimal AAR start layer would be deeper for that model.

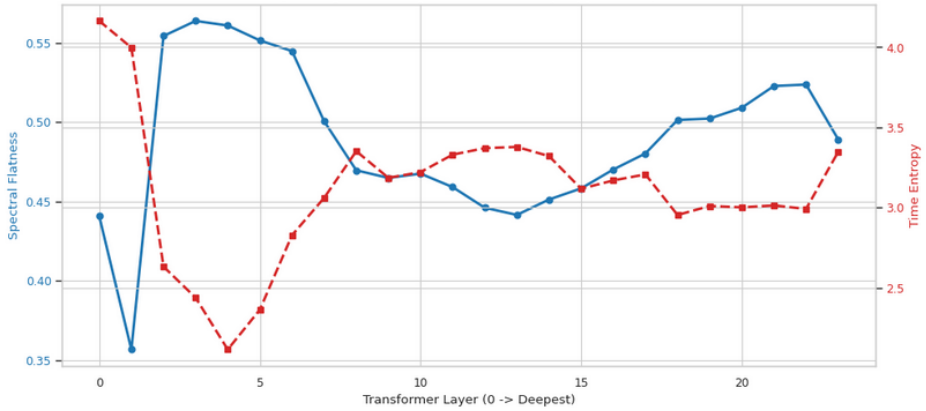


Figure C.2. *Layer-wise spectral flatness and Shannon entropy for Qwen1.5-MoE. The U-shape in spectral flatness is more pronounced than in OLMoE, reinforcing the model-agnostic nature of the finding.*

Connection to layer-selective AAR. The U-shaped spectral flatness profile offers a *model-intrinsic heuristic* for choosing the AAR start layer l_s without running the full layer-sweep of Chapter 4: apply AAR from some layer where spectral flatness has dropped, i.e., where the attention window starts to carry more structured, non-flat frequency content that a linear frequency router can exploit. For OLMoE, this corresponds roughly to layer 5 onwards, which is slightly earlier than the empirically optimal $l_s = 9$ found in Chapter 4. Similarly, Qwen1.5-MoE's flatness dip starts being substantial around layer 8, and has a minimum at layer 13: applying AAR-TF in layers $14, \ldots, 17$ yielded perfor-

mance improvements for GSM8K. The spectral flatness analysis also reveals that entropy and spectral flatness are not redundant: layers that appear similar in entropy can differ substantially in flatness, and vice versa. This is consistent with the complementary gains of AAR-TF over AAR-T reported in Table 3.2: the frequency-domain router accesses structure in the signal that the time-domain router cannot represent.

Appendix D

Future Work: AAR and Masked Diffusion Language Models

During the course of this thesis, work was conducted on data-efficient language modeling, published at the First BabyLM Workshop (2025) [50]; the submission ranked first on the NLP task suite of the BabyLM Challenge's STRICT (100M-word) track.

The BabyLM 2025 Challenge asks whether competitive language models can be trained under a strict data budget, roughly the amount of language a child is exposed to, rather than the internet-scale corpora used for frontier models. Under this constraint, every training signal must be spent efficiently, which makes the choice of learning objective especially consequential.

The work in [50] approaches this problem through **masked diffusion language modeling** [51] rather than the conventional autoregressive objective. In a masked diffusion model, a forward process progressively corrupts a sequence by masking tokens according to a noise schedule, and the model is trained to denoise, i.e., to reconstruct the original tokens, by optimizing a negative evidence lower bound (NELBO). The central contribution concerns the noise schedule: the study designs and compares several scheduling strategies, including two-mode schedules, together with different NELBO weighting schemes, while preserving the theoretical validity of the diffusion objective. Building on this, a complementary *frequency-informed* component biases the masking process by token frequency so that rare tokens, which carry disproportionate information yet appear too seldom to be learned well under a fixed data budget, are prioritized for reconstruction. Evaluated across the challenge's axes, linguistic competence, world knowledge, and human-likeness, the resulting models attain performance competitive with strong hybrid autoregressive–masked baselines, indicating that diffusion-based objectives are a viable alternative for data-restricted language learning.

This concurrent work points to a concrete direction for future research. Attention-Aware Routing was developed and evaluated entirely within *autoregressive*, decoder-only Mixture-of-Experts models, but its core premise, that a router benefits from conditioning on the contextual state summarized by attention weights, is not specific to that setting. Masked and diffusion language models expose an attention structure of their own, one that is bidirectional and, in the diffusion case, evolves across denoising steps

as the noise schedule unmasks tokens. They therefore invite a natural and largely unexplored extension of AAR: a diffusion-time router could condition not only on the current attention window but also on the denoising step or the local masking density, coupling the routing signal to the corruption process itself. Whether the attention-routing circuit identified in Chapter 3, and in particular the downstream amplification of attention sinks, has an analogue under bidirectional masked attention is an open question that would test the generality of the mechanism proposed in this thesis.