MIXTURE OF OUTPUTS: DYNAMIC DECODING-TIME EXPERTISE INTEGRATION BETWEEN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting large language models (LLMs) to new capabilities typically requires additional post-training, which can be costly in both supervision and computation. Parameter-efficient fine-tuning still trains task-specific parameters, while decoding-time adaptation can use predictions from auxiliary models. When a smaller model already has the desired specialized capability, it may serve as a donor for a larger model without updating either pretrained model. We study this form of decoding-time capability transfer. We propose *Mixture of Outputs (MoO)*, a lightweight framework that dynamically transfers capabilities from one or more frozen *donor* models to a frozen *receiver*. At each decoding step, MoO uses the receiver's hidden state to determine how much each donor should contribute to the next-token prediction. Only a lightweight module is trained to predict the contribution of donors, while the receiver and donor models remain frozen. Across nine reasoning benchmarks, single-donor MoO improves the receiver average by up to 14.27 percentage points. For alignment transfer, MoO reduces attack success rate from 0.44 to 0.02 on Qwen and from 1.00 to 0.17 on Llama. Combining multiple reasoning donors provides further gains of 2.01 and 1.92 percentage points over the best single-donor fusion on Qwen Base and Qwen Math, respectively. These gains are obtained with the receiver and all donor checkpoints frozen.

1 INTRODUCTION

Large language models (LLMs) have achieved strong performance across reasoning, instruction following, and safety-sensitive tasks. However, acquiring these abilities remains expensive. Turning a base model into a well-aligned or task-specialized model typically requires substantial post-training data and computation. The same problem persists after post-training: when an already capable model needs to acquire a new ability, such as stronger mathematical reasoning or safer refusal behavior, further training again requires additional supervision and optimization, while potentially degrading capabilities that the model already performs well. Existing approaches for reducing this adaptation cost mainly follow two directions: making parameter updates more efficient or avoiding updates to the target model altogether. Parameter-efficient fine-tuning (PEFT) reduces the number of trainable parameters, but still requires task-specific data, backpropagation through the target model, and a new adaptation process for each desired capability. Decoding-time approaches instead keep the target model unchanged and modify its next-token predictions using signals from auxiliary models. DExperts combines expert and anti-expert predictions (Liu et al., 2021), contrastive decoding exploits the difference between stronger and weaker models (Li et al., 2023), and proxy-tuning transfers a tuned-minus-untuned logit offset from a smaller proxy model to a larger model (Liu et al., 2024). While these approaches substantially reduce the cost of adapting the target model, they leave two limitations. Fixed output combinations apply the same coefficient to token-dependent signals, so they cannot suppress the signal when the auxiliary model is irrelevant and may distort the target model's prediction. PEFT reduces the number of updated parameters, but still requires task-specific data and a separate optimization process that writes the adaptation into target-side parameters. Thus, reducing adaptation cost while selectively incorporating useful capabilities from other models remains challenging. MoO instead learns a lightweight receiver-conditioned gate that suppresses unhelpful donor signals and strengthens useful ones.

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

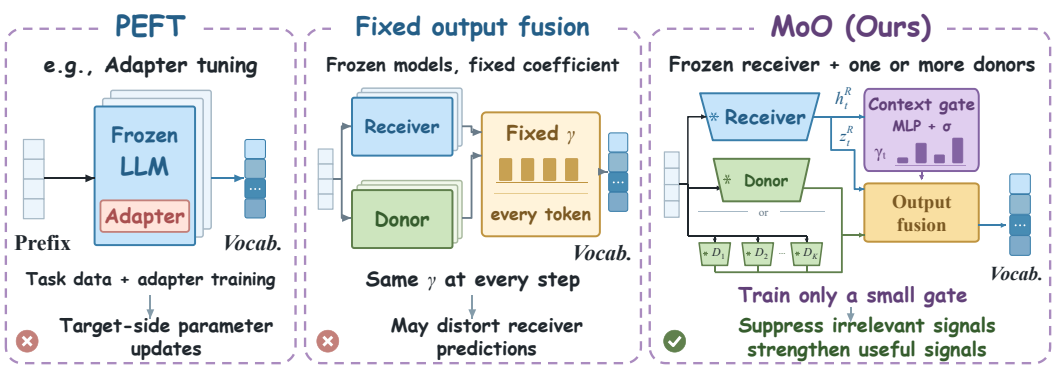


Figure 1: **Comparison of PEFT, decoding-time methods, and MoO.** Adapter-based PEFT learns task-specific parameters within the target network (Houlsby et al., 2019). The middle panel illustrates fixed output fusion: a frozen receiver and donor produce token-dependent logits combined with the same coefficient at every step. MoO keeps the receiver and donor frozen and learns donor strength from the receiver hidden state at each token.

Meanwhile, acquiring a specialized capability can be substantially cheaper in a smaller model than in a large one. Smaller models require less computation and memory to train or specialize, making them a natural low-cost carrier of task-specific capabilities. Although such a model may be weaker than a large model overall, specialization can still give it useful behavior on a particular task. For example, a small model specialized for mathematical reasoning may provide valuable reasoning signals even when a larger general-purpose model is stronger across most other tasks. This suggests a different route to low-cost adaptation: rather than teaching the large model the same capability again, can we obtain the capability in a smaller model at lower cost and then transfer its specialized behavior to the large model with only limited additional training? Prior work shows that output signals from smaller models can steer a larger frozen model at decoding time (Liu et al., 2024; Fan et al., 2024).

Inspired by model collaboration, where models with complementary strengths work together rather than requiring a single model to learn every capability (Shen et al., 2024; Fan et al., 2024; Zhang et al., 2026), we study capability transfer between a large general model and a smaller specialized model. We refer to the former as the **receiver** and the latter as the **donor**. Both language models remain frozen. Instead of modifying the receiver itself, we learn only a multilayer perceptron (MLP) that determines when and how strongly the donor should contribute to the receiver's prediction. In this way, the donor serves as a low-cost carrier of specialized capability, while the receiver retains its original parameters and generation interface. Based on this idea, we propose *Mixture of Outputs (MoO)*, a lightweight framework for selectively combining the outputs of a frozen receiver and a frozen donor. At each decoding step, both models produce next-token predictions. Rather than combining them with a fixed coefficient, MoO uses the receiver's last-layer hidden state to control the donor contribution. Because this hidden state summarizes the current prefix at the generation position, the gate can adjust the contribution of the donor according to the current context. We center the aligned donor logits and calibrate them to the receiver's logit scale before fusion. When the donor is unnecessary, a small gate keeps the prediction close to that of the original receiver; when the donor's specialized capability is useful, a larger gate allows its output to influence the next-token distribution. Importantly, only the lightweight fusion module is trained, while both pretrained language models remain unchanged. Figure 1 compares conventional model adaptation with MoO. In MoO, both pretrained models remain frozen, and only a lightweight module is trained to determine how much the donor should influence each generated token. Our evaluation shows that MoO can transfer complementary capabilities. Across nine benchmarks, Single-donor MoO improves the receiver average by up to 14.27 percentage points on Qwen Math. For alignment transfer, MoO reduces attack success rate from 0.44 to 0.02 on Qwen and from 1.00 to 0.17 on Llama. Combining two reasoning donors provides further gains of 2.01 and 1.92 percentage points over the best single-donor fusion on Qwen Base and Qwen Math, respectively.

Our contributions are summarized as follows:

- We study decoding-time transfer from multiple frozen specialized donors to a frozen receiver without updating receiver or donor parameters.

- We propose *Mixture of Outputs (MoO)*, a lightweight mechanism for realizing this transfer through model outputs. MoO learns a gate that selectively incorporates the donor's prediction into the receiver while preserving the receiver's output scale.
- We evaluate MoO on reasoning and alignment transfer, as well as multi-donor and model-mismatch settings. The results show gains from complementary donors in reasoning and alignment transfer and demonstrate multi-donor integration with a frozen receiver.

2 Related Work

2.1 Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) adapts pretrained models while updating only a small fraction of task-specific parameters. Early adapter methods insert compact trainable modules between frozen Transformer layers (Houlsby et al., 2019). Prefix- and prompt-tuning instead optimize continuous vectors while keeping the pretrained model fixed (Li & Liang, 2021; Lester et al., 2021). Other approaches reduce the adaptation footprint through sparse or structured parameterizations: $(IA)^3$ learns activation-rescaling vectors (Liu et al., 2022), while LoRA represents weight updates with low-rank factors (Hu et al., 2022). QLoRA further combines low-rank adaptation with a frozen quantized backbone to reduce memory during fine-tuning (Dettmers et al., 2023). These methods reduce adaptation cost, but still require target-specific parameters, prompts, or modules. In contrast, MoO keeps both the receiver and donor frozen and transfers capability directly from the pretrained donor. Only a lightweight gate is trained to control the donor's contribution to next-token prediction.

2.2 Decoding-Time Adaptation and Model Fusion

A complementary line of work changes model behavior at decoding time without updating the target model itself. PPLM steers generation by using gradients from lightweight attribute models to perturb hidden activations (Dathathri et al., 2020). GeDi and FUDGE guide next-token probabilities with auxiliary generative discriminators or future-attribute predictors (Krause et al., 2021; Yang & Klein, 2021), while DExperts combines a base model with expert and anti-expert distributions directly at decoding time (Liu et al., 2021). Related contrastive methods reshape token scores by comparing models or representations: contrastive decoding contrasts a stronger expert with a weaker amateur (Li et al., 2023), and DoLa contrasts logits projected from different Transformer layers to improve factuality (Chuang et al., 2024). Proxy-tuning applies the prediction difference between a tuned and an untuned small proxy model to the output of a larger frozen model (Liu et al., 2024). On Giant's Shoulders adaptively allocates the contributions of small models at each decoding step via Kullback–Leibler-constrained fusion (Fan et al., 2024). MoO uses the receiver's hidden state to set each donor's output-space contribution at every token while keeping the underlying models frozen.

3 Method

We now introduce *MoO*, our method for transferring complementary capabilities from multiple frozen donors to a frozen receiver through context-dependent output-space fusion. Given a receiver prediction and predictions from multiple donors for the same prefix, we ask two questions: which donor capabilities should modify the receiver's next-token distribution at the current decoding step, and how strongly should each donor contribute? MoO answers these questions with a lightweight token-level routing mechanism that independently controls the contribution of each donor while keeping all pretrained models frozen. We present the method in three steps. First, Section 3.1 introduces the Transformer preliminaries and formalizes the frozen receiver–multi-donor setting. Second, Section 3.2 centers each aligned donor output, calibrates it to the receiver logit scale, and uses the receiver hidden state to predict context-dependent fusion strengths. Finally, Section 3.3 summarizes the autoregressive decoding procedure and the objective used to train the routing gate while leaving the receiver and all donor models unchanged. Figure 2 illustrates the fusion and routing process.

3.1 Preliminaries and Problem Setup

Preliminaries. An autoregressive Transformer predicts the next token by turning the input prefix into a hidden representation and then into vocabulary logits. Given a token prefix $x_{<t} = (x_1, \ldots, x_{t-1})$, the model embeds the tokens, processes them with causal Transformer blocks, and returns a final hidden state at the last position. We denote this state by $h_t \in \mathbb{R}^d$, where d is the hid-

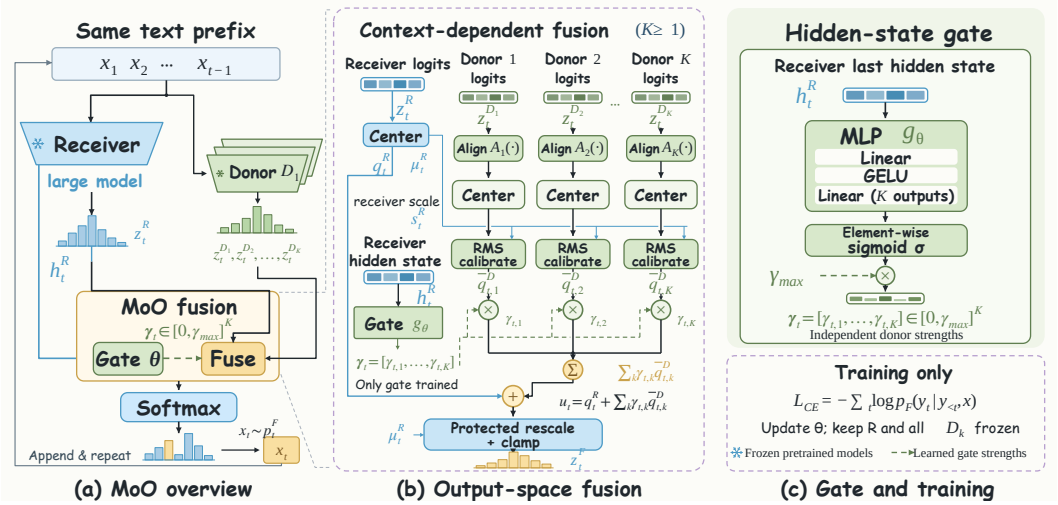


Figure 2: **MoO Framework.** (a) A frozen receiver and multiple frozen donors process the same causal prefix. (b) Each aligned donor output is centered, calibrated to the receiver's logit scale, independently gated, and added to the centered receiver logits. (c) A lightweight gate predicts context-dependent donor strengths from the receiver hidden state, allowing multiple donors to contribute simultaneously while keeping all pretrained models frozen.

den dimension. A language-model head maps the hidden state to logits $z_t = W_{\mathrm{lm}} h_t + b_{\mathrm{lm}} \in \mathbb{R}^{|V|}$, where V is the vocabulary, $W_{\mathrm{lm}} \in \mathbb{R}^{|V| \times d}$ is the language-model head weight, and $b_{\mathrm{lm}} \in \mathbb{R}^{|V|}$ is its bias. The next-token distribution is obtained as $p(x_t \mid x_{<t}) = \mathrm{softmax}(z_t)$. MoO operates at this final logit interface.

Problem Setup. The goal is to transfer multiple potentially complementary capabilities while keeping the receiver as the model that owns the generation interface. Let R be a frozen receiver with vocabulary V_R and hidden dimension d_R. At decoding step t, it produces a final hidden state $h_t^R \in \mathbb{R}^{d_R}$ and logits $z_t^R \in \mathbb{R}^{|V_R|}$. We consider K frozen donors $\mathcal{D} = \{D_k\}_{k=1}^K$, which may carry different capabilities, such as mathematical reasoning, code generation, or safe refusal. Given the same text prefix, donor D_k produces logits $z_t^{D_k} \in \mathbb{R}^{|V_{D_k}|}$ using its own tokenizer.

The receiver retains the generation interface: donor predictions are aligned to its vocabulary before fusion, and generated tokens are selected from V_R. A lightweight gate $g_\theta : \mathbb{R}^{d_R} \to \mathbb{R}^K$ reads the receiver hidden state and outputs one fusion score per donor. The gate parameters θ are the only trainable parameters; the receiver and all donor checkpoints remain frozen.

3.2 Context-Dependent Multi-Donor Output-Space Fusion

Vocabulary Alignment and Scale Calibration. Before a donor can affect the receiver prediction, its output must be expressed in the receiver vocabulary. For donor D_k, an alignment map $A_k : \mathbb{R}^{|V_{D_k}|} \to \mathbb{R}^{|V_R|}$ produces $\tilde{z}_t^{D_k} = A_k\left(z_t^{D_k}\right)$. If the receiver and donor use compatible tokenizers, $A_k(\cdot)$ is the identity up to language-model-head padding or truncation. Otherwise, it returns receiver-space pseudo-logits through a separately learned vocabulary projection, held fixed during gate training.

Let $M = |V_R|$ and $\mathbf{1} \in \mathbb{R}^M$ be the all-ones vector. For receiver-space u, define its mean, centered form, and root-mean-square (RMS) as $\mu(u) = \frac{1}{M}\sum_{j=1}^M u_j, \quad c(u) = u - \mu(u)\mathbf{1}, \quad r(u) = \sqrt{\frac{1}{M}\|u\|_2^2 + \epsilon}$. The receiver statistics are $\mu_t^R = \mu(z_t^R), \quad q_t^R = c(z_t^R), \quad s_t^R = r(q_t^R)$. Since independently trained models may produce logits with different scales, we center each aligned donor and calibrate its RMS to the receiver: $q_{t,k}^D = c\left(\tilde{z}_t^{D_k}\right), \quad \bar{q}_{t,k}^D = q_{t,k}^D \frac{s_t^R}{\max\left(r(q_{t,k}^D), \epsilon\right)}$. Every donor therefore contributes a centered vector in the receiver vocabulary and on the receiver's logit scale.

Independent Multi-donor Gate. The main design choice of MoO is to make donor fusion both context-dependent and compositional. A fixed mixture uses the same donor coefficient for all pre-

fixes, while a normalized routing distribution forces different donors to compete for a fixed probability mass. This may be undesirable when multiple complementary capabilities are simultaneously useful at the same decoding step. MoO instead assigns an independent bounded strength to every donor. The receiver hidden state is passed through the gate to obtain

$$\boldsymbol{\gamma}_t = \gamma_{\max}\sigma\left(g_\theta(h_t^R)\right) \in [0, \gamma_{\max}]^K, \tag{1}$$

where the sigmoid $\sigma(\cdot)$ is applied element-wise and $\boldsymbol{\gamma}_t = [\gamma_{t,1}, \gamma_{t,2}, \ldots, \gamma_{t,K}]$. Here $\gamma_{t,k}$ determines how strongly donor D_k influences the receiver at decoding step t. Equation 1 therefore provides one context-dependent coefficient for each donor.

The weighted donor vectors are added to the centered receiver:

$$u_t = q_t^R + \sum_{k=1}^{K} \gamma_{t,k}\bar{q}_{t,k}^D. \tag{2}$$

Because independent gates need not sum to one, several active donors can raise the magnitude of u_t. The implementation applies the protected scale $a_t = \min\left(\frac{s_t^R}{\max(r(u_t),\tau)}, a_{\max}\right)$, $\tau = 0.1$, $a_{\max} = 4.0$, and produces

$$z_t^F = \mathrm{clip}_{[-50,50]}\left(\mu_t^R\mathbf{1} + a_t u_t\right). \tag{3}$$

Statistics are computed over real receiver-vocabulary positions; padded positions are masked after fusion. The floor, ceiling, and clamp in Equation 3 are the numerical safeguards used by the formal experiments.

Small gates keep the prediction close to the receiver, while independent strengths allow one or several donors to contribute at a given step.

Single-donor Special Case. For $K = 1$, Equation 2 reduces to $u_t = q_t^R + \gamma_{t,1}\bar{q}_{t,1}^D$, followed by the same protected rescaling and clamp.

3.3 Decoding and Training

Decoding. At each autoregressive step, the receiver and all K donors process the same text prefix. The receiver provides both the base logits and the hidden state used for routing, while donor outputs are aligned, centered, and calibrated to the receiver's logit scale. Equations 1–3 then determine the fused logits. Given an input x and continuation prefix $y_{<t}$, the next-token distribution is $p_F(\cdot \mid x, y_{<t}) = \mathrm{softmax}(z_t^F)$. During inference, $y_{<t}$ is replaced by the generated continuation $\hat{y}_{<t}$, and the next token is sampled or selected from $p_F(\cdot \mid x, \hat{y}_{<t})$. The generated token extends the shared text prefix for the following step. Algorithm 1 summarizes this procedure for one decoding step.

Causal Routing. The gate makes its decision before the current token is generated. Its input is the receiver hidden state computed from $(x, y_{<t})$ during teacher-forced training and from $(x, \hat{y}_{<t})$ during inference. Both the donor strengths and the fused logits therefore depend only on the causal prefix, without access to the current target token y_t or future tokens.

Training Objective. Given a target continuation $y = (y_1, \ldots, y_T)$, the gate is supervised by the autoregressive cross-entropy:

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{t=1}^{T} \log p_F(y_t \mid x, y_{<t}). \tag{4}$$

Reasoning adds the donor-budget regularizer with target 0.3 and coefficient 0.05 (Appendix A); alignment uses only $\mathcal{L}_{\mathrm{CE}}$. Optimization updates only θ, keeping all language models frozen. Training data may combine examples corresponding to the donors' complementary capabilities, such as mathematical, coding, and general problem-solving examples for reasoning transfer. The autoregressive objective supervises the gate through the fused target-token probability, learning when and how strongly each donor vector should influence the receiver. Adaptation is therefore encoded in the lightweight routing module, while the pretrained checkpoints remain unchanged.

4 Experiments

In this section, we answer three questions. First, can a frozen specialized donor improve a frozen receiver under a common reasoning protocol? Second, can the same output-space mechanism transfer alignment behavior without erasing receiver utility? Third, can independently gated donors be

Algorithm 1 MoO: One Multi-Donor Autoregressive Decoding Step

Require: Text prefix $x_{<t}$; frozen receiver R; frozen donors $\mathcal{D} = \{D_k\}_{k=1}^{K}$; alignment maps $\{A_k\}_{k=1}^{K}$; gate g_θ; maximum donor strength $\gamma_{\max}$
Ensure: Fused next-token distribution $p_F(\cdot \mid x_{<t})$

$(h_t^R, z_t^R) \leftarrow R(x_{<t})$
$(\mu_t^R, q_t^R, s_t^R) \leftarrow \big(\mu(z_t^R), c(z_t^R), r(c(z_t^R))\big)$
for $k = 1, \ldots, K$ **do**
  $z_t^{D_k} \leftarrow D_k(x_{<t})$
  $\tilde{z}_t^{D_k} \leftarrow A_k(z_t^{D_k})$
  $q_{t,k}^D \leftarrow c(\tilde{z}_t^{D_k})$
  $\bar{q}_{t,k}^D \leftarrow q_{t,k}^D \, s_t^R / \max(r(q_{t,k}^D), \epsilon)$
end for
$\boldsymbol{\gamma}_t \leftarrow \gamma_{\max} \sigma\big(g_\theta(h_t^R)\big)$
$u_t \leftarrow q_t^R + \sum_{k=1}^{K} \gamma_{t,k} \bar{q}_{t,k}^D$
$a_t \leftarrow \min\big(s_t^R / \max(r(u_t), 0.1), 4.0\big)$
$z_t^F \leftarrow \mathrm{clip}(\mu_t^R \mathbf{1} + a_t u_t, -50, 50)$
$p_F(\cdot \mid x_{<t}) \leftarrow \mathrm{softmax}(z_t^F)$



Figure 3: **Multi-donor Reasoning Integration.** Avg. covers all nine benchmarks; four tasks are shown individually. Joint AceMath and R1-Qwen fusion uses independent token-wise gates and improves over the best single-donor fusion by 2.01 percentage points on Qwen Base and 1.92 on Qwen Math.

composed in one receiver? Section 4 answers these questions with single-donor reasoning results, alignment results across three receiver families, and two formal multi-donor evaluations. Section 4.1 then analyzes dynamic routing and the gate-input choice.

Experimental Setup. We evaluate reasoning transfer using Qwen2.5-7B-Base, Qwen2.5-Math-7B (Qwen et al., 2024), and Gemma-2-9B-it (Gemma Team et al., 2024) as receivers, with AceMath-1.5B-Instruct (Liu et al., 2025) and DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025) as donors. For alignment transfer, we use Qwen2.5-7B-Base, Llama-3-8B-Base (Grattafiori et al., 2024), and Gemma-2-9B-it as receivers, paired with Qwen2.5-1.5B-Instruct or Llama-3.2-1B-Instruct (Meta AI, 2024). Reasoning is evaluated on GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021b; Lightman et al., 2024), MMLU (Hendrycks et al., 2021a), AIME24 (Mathematical Association of America, 2024), GSM-Hard (Gao et al., 2023), ARC-Challenge (ARC-C) (Clark et al., 2018), MMLU-Pro (Wang et al., 2024), HumanEval (Chen et al., 2021), and MBPP (Austin et al., 2021). Alignment is evaluated with AdvBench (Zou et al., 2023), XSTest (Rottger et al., 2024), and IFEval (Zhou et al., 2023), together with MMLU, GSM8K, and HumanEval for capability evaluation. We compare against receiver-only, donor-only, fixed fusion, DExperts (Liu et al., 2021), proxy-tuning (Liu et al., 2024), and supervised LoRA fine-tuning (SFT–LoRA) (Hu et al., 2022).

Table 1: **Reasoning transfer across three receivers.** Avg. is the uniform mean of the nine benchmarks listed in the experimental setup; three representative tasks are shown here. Bold marks the better score within each donor–MoO pair. Small superscripts on MoO scores show receiver-relative changes in percentage points. Full results are reported in Appendix B.

Source	Decode	Avg. ↑	GSM8K ↑	MATH500 ↑	GSM-Hard ↑
Qwen2.5-7B-Base					
–	Receiver	0.6275	0.8400	0.5950	0.6500
AceMath	Donor	0.5327	0.8700	0.7400	0.7250
	MoO	**0.6290**$^{+0.15}$	**0.8750**$^{+3.50}$	**0.7800**$^{+18.50}$	**0.7550**$^{+10.50}$
R1-Qwen	Donor	0.2801	0.6900	0.3800	0.3000
	MoO	**0.5930**$^{-3.45}$	**0.8350**$^{-0.50}$	**0.6600**$^{+6.50}$	**0.6950**$^{+4.50}$
Qwen2.5-Math-7B					
–	Receiver	0.4304	0.5750	0.5300	0.4800
AceMath	Donor	0.5327	0.8700	0.7400	0.7250
	MoO	**0.5731**$^{+14.27}$	**0.9150**$^{+34.00}$	**0.7650**$^{+23.50}$	**0.7400**$^{+26.00}$
R1-Qwen	Donor	0.2801	0.6900	0.3800	0.3000
	MoO	**0.4651**$^{+3.47}$	**0.7450**$^{+17.00}$	**0.6550**$^{+12.50}$	**0.6250**$^{+14.50}$
Gemma-2-9B-it					
–	Receiver	0.5813	0.8000	0.5150	0.6500
AceMath	Donor	0.5327	**0.8700**	**0.7400**	**0.7250**
	MoO	**0.5868**$^{+0.55}$	0.8050$^{+0.50}$	0.5000$^{-1.50}$	0.6850$^{+3.50}$
R1-Qwen	Donor	0.2801	0.6900	0.3800	0.3000
	MoO	**0.5825**$^{+0.12}$	**0.8050**$^{+0.50}$	**0.5050**$^{-1.00}$	**0.6550**$^{+0.50}$

Alignment Capability Transfer. We next evaluate whether MoO can transfer alignment behavior while preserving the receiver's general capabilities. Table 2 compares the frozen receiver, the alignment donor, and hidden-state-gated MoO on attack success rate (ASR), over-refusal, instruction following, and the average capability score. Lower ASR and over-refusal are better, whereas higher IFEval-I and capability scores are better. On Qwen, MoO reduces ASR from 0.4400 to 0.0200 while Capability Avg. changes from 0.7707 to 0.7500. On Llama, it reduces ASR from 1.0000 to 0.1700 and raises Capability Avg. from 0.3299 to 0.4743. On Gemma, MoO keeps ASR at 0.0150 and raises Capability Avg. from 0.6996 to 0.7136, while IFEval-I falls from 0.7959 to 0.7707.

Table 2: **Alignment transfer across receiver–donor pairs.** Capability Avg. is the mean of the formal capability benchmarks; ASR is attack success rate, Over-ref. is benign-prompt refusal, and IFEval-I is instruction-level strict accuracy. Bold marks the best score in each receiver group (including ties). Small signed values beside MoO scores are changes from the receiver in percentage points; green marks improvement.

Decode	Capability Avg. ↑	ASR ↓	Over-ref. ↓	IFEval-I ↑
Qwen2.5-7B-Base Donor: Qwen2.5-1.5B-Inst.				
Receiver	**0.7707**	0.4400	**0.0200**	0.4862
Donor	0.5852	**0.0200**	0.2250	0.4838
MoO	0.7500$^{-2.07}$	**0.0200**$^{-42.00}$	0.0800$^{+6.00}$	**0.4982**$^{+1.20}$
Llama-3-8B-Base Donor: Llama-3.2-1B-Inst.				
Receiver	0.3299	1.0000	**0.0050**	0.3445
Donor	0.3666	0.5250	0.0950	**0.5354**
MoO	**0.4743**$^{+14.44}$	**0.1700**$^{-83.00}$	0.0550$^{+5.00}$	0.4190$^{+7.45}$
Gemma-2-9B-it Donor: Qwen2.5-1.5B-Inst.				
Receiver	0.6996	0.0150	**0.1200**	**0.7959**
Donor	0.5885	**0.0050**	0.4900	0.4970
MoO	**0.7136**$^{+1.40}$	0.0150$^{0.00}$	0.1250$^{+0.50}$	0.7707$^{-2.52}$

Multi-Donor Capability Integration. We further study whether multiple specialized donors can provide complementary correction signals to a single frozen receiver. We use AceMath and R1-Qwen as two reasoning donors and compare their individual transfer results with joint multi-donor

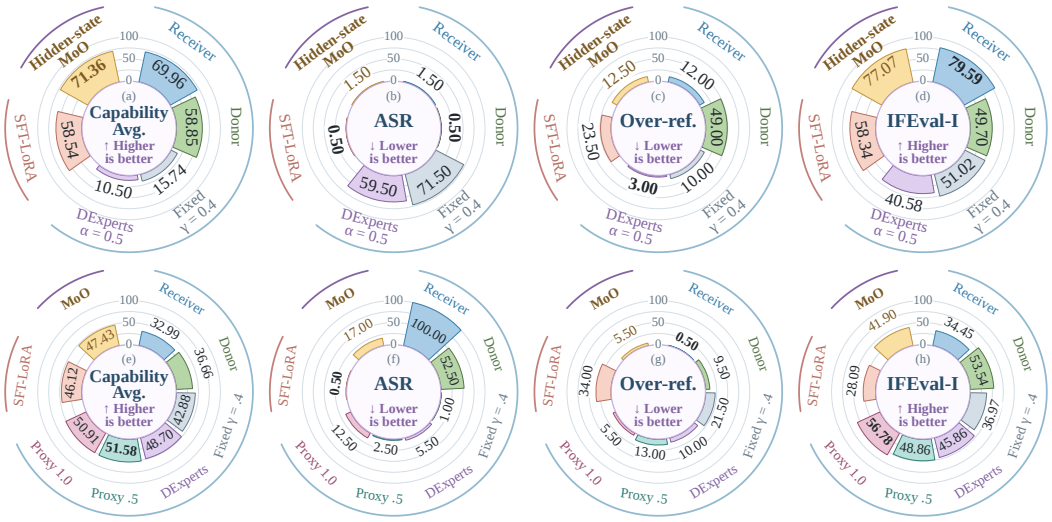


Figure 4: **Alignment-transfer baseline comparison.** Panels (a–d) use Gemma-2-9B-it with a Qwen2.5-1.5B-Instruct donor; panels (e–h) use Llama-3-8B-Base with a Llama-3.2-1B-Instruct donor. Higher is better for Capability Avg. and IFEval-I, while lower is better for ASR and over-refusal. Fixed fusion uses $\gamma = 0.4$, and DExperts and proxy-tuning use the displayed α. Fixed fusion and DExperts use the same vocabulary map as MoO on the heterogeneous Gemma/Qwen pair. Proxy-tuning is omitted there because its strict logit difference requires compatible token IDs for the receiver, tuned proxy, and base proxy. Gold highlights hidden-state MoO; gate-input variants are reported in Table 3.

fusion. Figure 3 reports the nine-benchmark average and four individual tasks, extending the three shown in Table 1. Joint fusion reaches Avg. 0.6491 on Qwen Base, a 2.16-point gain over its receiver and a 2.01-point gain over its stronger single-donor row. On Qwen Math, joint fusion reaches 0.5923, 16.19 points above the receiver and 1.92 points above the stronger single-donor row.

Comparison with Baselines. Figure 4 compares hidden-state MoO with receiver-only and donor-only decoding, fixed fusion, DExperts, proxy-tuning when applicable, and SFT–LoRA on two receiver families. On the heterogeneous Gemma/Qwen pair, MoO raises Capability Avg. from 0.6996 to 0.7136 while matching the receiver's ASR of 0.0150; over-refusal changes from 0.1200 to 0.1250, and IFEval-I from 0.7959 to 0.7707. In contrast, fixed fusion and DExperts reduce Capability Avg. to 0.1574 and 0.1050 and increase ASR to 0.7150 and 0.5950. Both baselines use the same vocabulary map as MoO, showing that dynamic routing is important when projected donor corrections are not uniformly reliable. Strict proxy-tuning is unavailable for this pair because the Gemma receiver and Qwen proxy models do not share token IDs.

On the tokenizer-compatible Llama pair, MoO raises Capability Avg. from 0.3299 to 0.4743, lowers ASR from 1.0000 to 0.1700, and improves IFEval-I from 0.3445 to 0.4190, with over-refusal increasing from 0.0050 to 0.0550. DExperts and both proxy-tuning settings are also effective; Proxy-tuning at $\alpha = 1.0$ reaches Capability Avg. 0.5091 and IFEval-I 0.5678 with ASR 0.1250. Thus, the Llama results establish strong receiver-relative transfer rather than universal dominance over every same-tokenizer baseline. Gate-input variants are evaluated separately in Table 3.

4.1 Experimental Analysis

We next analyze token-wise donor strengths, compare dynamic and fixed fusion, and test how the receiver representation affects gating.

Token-wise gate dynamics. MoO predicts a donor-specific gate vector $\boldsymbol{\gamma}_t$ from the receiver hidden state. Figure 5 traces its two components over 80 generated positions for one math probe and one code probe. The math probe has mean strengths $(0.049, 0.481)$ for the math and code donors, while the code probe has $(0.245, 0.394)$.

The unsmoothed trajectories vary within each sequence, showing that the learned gates are token-wise rather than fixed. These traces use the Qwen math/code configuration in Appendix D. Separately, the Llama diagnostic in Table 3 covers 240 teacher-forced probes and 20,803 tokens from MATH500, GSM8K, ARC-C, and MMLU, testing domain selectivity under a different configuration.

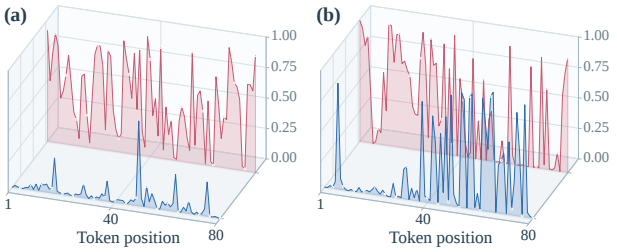


Figure 5: **Token-wise gate dynamics.** Math (a) and code (b); blue denotes math, red code.

Dynamic Versus Fixed Fusion. We next test whether token-wise adaptation matters by comparing the learned gate with fixed fusion using a single coefficient γ at every decoding step. Figure 6 sweeps $\gamma \in \{0, 0.3, 0.4, 0.5\}$ and compares it with dynamic MoO. Dynamic routing improves representative metrics across receiver families: on Qwen, ASR drops to 2.00% versus 6.50% for the best nonzero fixed strength; on Llama, IFEval-I rises to 41.90% versus 36.97%; and in the heterogeneous Gemma/Qwen setting, it achieves 71.36% Capability Avg., while nonzero fixed fusion reaches at most 21.11%. Fixed fusion leads on some capability metrics; dynamic routing reaches operating points the sweep misses.

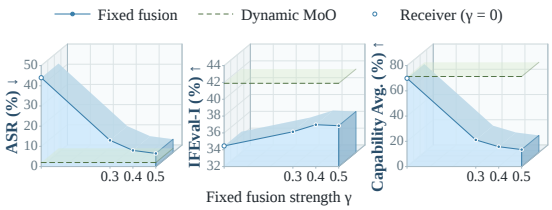


Figure 6: **Fixed versus dynamic fusion.** Dashed curves show dynamic MoO.

Table 3: **Complete gate-input selectivity diagnostic.** Mean donor strengths pool 120 math and 120 knowledge probes. Gap is $\bar{\gamma}_{\text{math}} - \bar{\gamma}_{\text{know}}$; P10 is the 10th percentile over token-level strengths.

Gate input	Mean donor strength		Gap ↑	Lower tail (P10)	
	Math	Knowledge	$\bar{\gamma}_{\text{math}} - \bar{\gamma}_{\text{know}}$	Math	Knowledge
T Token embedding	0.933	0.932	+0.001	0.840	0.841
h **Last hidden state**	0.771	0.660	**+0.111**	0.155	0.036
$\bar{h}$ Causal context mean	0.984	0.982	+0.002	0.974	0.964

Choice of Gate Input. We compare gates conditioned on the current token embedding, the receiver's last hidden state, or a causal mean of receiver states. In the Llama reasoning diagnostic, we measure mean donor strength on teacher-forced math probes (MATH500 and GSM8K) and knowledge probes (ARC-C and MMLU), using 60 examples from each benchmark. Table 3 shows that token conditioning and causal averaging produce nearly constant strengths across domains, with gaps of 0.001 and 0.002. The last hidden state instead yields a 0.111 gap by assigning more donor weight to math contexts. Its lower-tail statistics also differ across domains (P10 0.155 versus 0.036). This measured domain separation motivates using the last hidden state as the gate input.

5 Conclusion

We presented MoO, a lightweight output-space method for transferring a specialized donor capability into a frozen receiver. MoO keeps both language models unchanged, aligns the donor signal into the receiver vocabulary, normalizes the donor direction to the receiver logit scale, and uses the receiver's hidden state to decide the donor strength at each decoding step. Across reasoning and alignment evaluations, the learned gate improves several receiver–donor pairings; the multi-donor results show further gains from combining specialized reasoning signals. The heterogeneous comparison also shows the value of controlling donor strength when fixed fusion causes interference. These results support receiver-conditioned output fusion as a way to reuse specialized donor capabilities while keeping the language models frozen.

AI USE STATEMENT

Generative AI tools were used to polish the grammar and wording of the manuscript. The authors reviewed the final text and take responsibility for its content.

ETHICS STATEMENT

This work uses public model checkpoints and benchmarks and involves no human subjects or private data. Harmful prompts and generated responses are used only for controlled safety evaluation. Because output fusion can also transfer undesirable behavior, deployment requires evaluation of both receiver and donors together with application-specific safeguards.

REPRODUCIBILITY STATEMENT

Section 3 gives the fusion equations and decoding algorithm, and Section 4 specifies models, benchmarks, and baselines. The appendix reports optimization and decoding settings, receiver–donor pairings, complete nine-benchmark scores, fixed-strength sweeps, and diagnostic protocols.

REFERENCES

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. DoLa: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/edc36117f795ca52a0cbf6a7b3882859-Abstract-Conference.html.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

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

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2020.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0441. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1feb87871436031bdc0f2beaa62a049b-Abstract-Conference.html.

Chenghao Fan, Zhenyi Lu, Wei Wei, Jie Tian, Xiaoye Qu, Dangyang Chen, and Yu Cheng. On giant's shoulders: Effortless weak to strong by dynamic logits fusion. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-0944. URL https://neurips.cc/virtual/2024/poster/95177.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/gao23f.html.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Leonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Rame, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems*, volume 34, 2021b. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2790–2799. PMLR, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4929–4952. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-emnlp.424.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 12286–12312, 2023. doi: 10.18653/v1/2023.acl-long.687. URL https://aclanthology.org/2023.acl-long.687/.

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

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pp. 6691–6706, 2021. doi: 10.18653/v1/2021.acl-long.522. URL https://aclanthology.org/2021.acl-long.522/.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. Tuning language models by proxy. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=dribhnhm1i.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-0142. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/0cde695b83bd186c1fd456302888454c-Abstract-Conference.html.

Zihan Liu, Yang Chen, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. AceMath: Advancing frontier math reasoning with post-training and reward modeling. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3993–4015. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.206. URL https://aclanthology.org/2025.findings-acl.206/.

Mathematical Association of America. 2024 american invitational mathematics examination. American Mathematics Competitions, 2024. URL https://maa.org/student-programs/amc/.

Meta AI. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. Official release announcement, 2024. URL https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Paul Rottger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5377–5400, 2024. doi: 10.18653/v1/2024.naacl-long.301. URL https://aclanthology.org/2024.naacl-long.301/.

Zejiang Shen, Hunter Lang, Bailin Wang, Yoon Kim, and David Sontag. Learning to decode collaboratively with multiple language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 12974–12990, 2024. doi: 10.18653/v1/2024.acl-long.701. URL https://aclanthology.org/2024.acl-long.701/.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024. URL https://arxiv.org/abs/2406.01574.

Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.276.

Chen Zhang, Jiuheng Lin, Zhiyuan Liao, and Yansong Feng. Efficient low-resource language adaptation via multi-source dynamic logit fusion. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4540–4557. Association for Computational Linguistics, July 2026. doi: 10.18653/v1/2026.acl-long.207. URL https://aclanthology.org/2026.acl-long.207/.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023. URL https://arxiv.org/abs/2307.15043.

APPENDIX OVERVIEW

Appendix A records training settings, evaluation samples, and model pairings. Appendices B and C provide the complete reasoning and fixed-strength alignment results. Appendix D distinguishes the generation traces from the teacher-forced gate-input diagnostic.

A EXPERIMENTAL DETAILS

A.1 TRAINING AND EVALUATION PROTOCOL

The receiver and donor language models remain frozen; training updates only the gate. Cross-vocabulary projections are learned separately and held fixed during gate training (Section 3). Tables 4 and 5 record the training settings and evaluation sample counts.

Table 4: **Training settings.** Shared values apply to both tasks. R/A denotes reasoning/alignment where settings differ. The budget target and coefficient apply only to reasoning.

Setting	Value
Training steps (R / A)	3,000 / 1,200
Learning rate	5×10^{-4}
Sequence length	2,048
Micro / effective batch	1 / 8
Weight decay	0
Gradient clipping	1.0
Precision	bfloat16
Gate hidden dimension	64
Maximum strength $\gamma_{\max}$	1.0
Budget target / coefficient	0.3 / 0.05
Random seed	42
Evaluation decoding	greedy
Donor mode	no thinking

Table 5: **Evaluation sample counts.** Columns denote reasoning (R) and alignment (A). IFEval contains 540 prompts and 833 instructions. Dashes indicate benchmarks outside the suite.

Benchmark	R	A
GSM8K	200	200
MATH500	200	–
MMLU	500	200
AIME24	30	–
GSM-Hard	200	–
ARC-C	500	–
MMLU-Pro	500	–
HumanEval	164	164
MBPP	200	–
AdvBench	–	200
XSTest	–	200
IFEval	–	540

The reasoning Avg. is the unweighted mean of nine benchmark scores; Capability Avg. averages MMLU, GSM8K, and HumanEval with equal weights. For alignment, AdvBench measures attack success rate (ASR), XSTest measures over-refusal (Over-ref.), and IFEval-I is instruction-level strict accuracy. Alignment uses only cross-entropy (Equation 4). Effective batch size includes gradient accumulation; decoding uses standard donor outputs without thinking.

A.2 MODEL PAIRINGS AND VOCABULARY ALIGNMENT

Table 6 lists the model pairings. AceMath denotes AceMath-1.5B-Instruct, and R1-Qwen denotes DeepSeek-R1-Distill-Qwen-1.5B. Shared-tokenizer pairs use the identity map; heterogeneous pairs use the learned projection in Section 3. Each donor has its own gate output.

Table 6: **Receiver–donor pairings.** The reasoning rows list the donors used in the single-donor runs. Joint AceMath + R1-Qwen results are reported for the two Qwen receivers in Appendix B.

Receiver	Donor(s)	Vocabulary map
Reasoning transfer		
Qwen2.5-7B-Base	AceMath / R1-Qwen	Identity
Qwen2.5-Math-7B	AceMath / R1-Qwen	Identity
Gemma-2-9B-it	AceMath / R1-Qwen	Learned projection
Alignment transfer		
Qwen2.5-7B-Base	Qwen2.5-1.5B-Instruct	Identity
Llama-3-8B-Base	Llama-3.2-1B-Instruct	Identity
Gemma-2-9B-it	Qwen2.5-1.5B-Instruct	Learned projection

B Complete Reasoning Results

Tables 7–9 expand Table 1 to all nine benchmarks. Receiver and donor columns are standalone models; MoO-Multi jointly gates AceMath and R1-Qwen. Pale teal identifies MoO columns. Superscripts show receiver-relative changes in percentage points: green indicates improvement, red degradation, and grey no change.

Table 7: **Complete reasoning results: Qwen2.5-7B-Base.** Scores are fractions; higher is better. Bold marks the best in each row, including ties.

Benchmark	Receiver	AceMath		R1-Qwen		Both donors
	↑	Donor ↑	MoO ↑	Donor ↑	MoO ↑	MoO-Multi ↑
Avg.	0.6275	0.5327	0.6290 +0.15	0.2801	0.5930 -3.45	**0.6491** +2.16
GSM8K	0.8400	0.8700	0.8750 +3.50	0.6900	0.8350 -0.50	**0.8950** +5.50
MATH500	0.5950	0.7400	**0.7800** +18.50	0.3800	0.6600 +6.50	0.7300 +13.50
MMLU	**0.7260**	0.4420	0.6280 -9.80	0.3740	0.7020 -2.40	0.7160 -1.00
AIME24	0.0333	0.1000	**0.1333** +10.00	0.0000	0.0667 +3.34	0.1000 +6.67
GSM-Hard	0.6500	0.7250	**0.7550** +10.50	0.3000	0.6950 +4.50	0.7500 +10.00
ARC-C	**0.8960**	0.5700	0.7940 -10.20	0.4600	0.8660 -3.00	0.8720 -2.40
MMLU-Pro	**0.4500**	0.2040	0.3220 -12.80	0.1100	0.3300 -12.00	0.4220 -2.80
HumanEval	**0.7622**	0.5183	0.7134 -4.88	0.0366	0.5976 -16.46	0.6768 -8.54
MBPP	**0.6950**	0.6250	0.6600 -3.50	0.1700	0.5850 -11.00	0.6800 -1.50

Table 8: **Complete reasoning results: Qwen2.5-Math-7B.** Scores are fractions; higher is better. Bold marks the best in each row, including ties.

Benchmark	Receiver	AceMath		R1-Qwen		Both donors
	↑	Donor ↑	MoO ↑	Donor ↑	MoO ↑	MoO-Multi ↑
Avg.	0.4304	0.5327	0.5731 +14.27	0.2801	0.4651 +3.47	**0.5923** +16.19
GSM8K	0.5750	0.8700	**0.9150** +34.00	0.6900	0.7450 +17.00	0.9050 +33.00
MATH500	0.5300	0.7400	0.7650 +23.50	0.3800	0.6550 +12.50	**0.8100** +28.00
MMLU	**0.5880**	0.4420	0.5000 -8.80	0.3740	0.5300 -5.80	0.5060 -8.20
AIME24	0.0333	0.1000	0.0667 +3.34	0.0000	**0.1667** +13.34	0.1333 +10.00
GSM-Hard	0.4800	0.7250	0.7400 +26.00	0.3000	0.6250 +14.50	**0.7700** +29.00
ARC-C	**0.7540**	0.5700	0.6220 -13.20	0.4600	0.6700 -8.40	0.6460 -10.80
MMLU-Pro	**0.2680**	0.2040	0.2360 -3.20	0.1100	0.2160 -5.20	0.2600 -0.80
HumanEval	**0.6402**	0.5183	0.6280 -1.22	0.0366	0.5732 -6.70	**0.6402** +0.00
MBPP	0.0050	0.6250	**0.6850** +68.00	0.1700	0.0050 +0.00	0.6600 +65.50

Table 9: **Complete reasoning results: Gemma-2-9B-it.** Scores are fractions; higher is better. Bold marks the best in each row, including ties.

Benchmark	Receiver	AceMath		R1-Qwen	
	↑	Donor ↑	MoO ↑	Donor ↑	MoO ↑
Avg.	0.5813	0.5327	**0.5868** +0.55	0.2801	0.5825 +0.12
GSM8K	0.8000	**0.8700**	0.8050 +0.50	0.6900	0.8050 +0.50
MATH500	0.5150	**0.7400**	0.5000 -1.50	0.3800	0.5050 -1.00
MMLU	0.6980	0.4420	0.6980 +0.00	0.3740	**0.7000** +0.20
AIME24	0.0000	**0.1000**	0.0000 +0.00	0.0000	0.0000 +0.00
GSM-Hard	0.6500	**0.7250**	0.6850 +3.50	0.3000	0.6550 +0.50
ARC-C	**0.9020**	0.5700	**0.9020** +0.00	0.4600	**0.9020** +0.00
MMLU-Pro	**0.4680**	0.2040	0.4640 -0.40	0.1100	0.4620 -0.60
HumanEval	0.6037	0.5183	**0.6220** +1.83	0.0366	0.6037 +0.00
MBPP	0.5950	**0.6250**	0.6050 +1.00	0.1700	0.6100 +1.50

C Complete Fixed-Strength Alignment Results

Table 10 expands Figure 6 to all four metrics under the same receiver-only, fixed-strength, and hidden-state-gate conditions.

Table 10: **Complete fixed-strength alignment results.** Capability Avg. averages MMLU, GSM8K, and HumanEval; IFEval-I is instruction-level strict accuracy. Scores are fractions. Arrows indicate the preferred direction; bold marks the best value within each receiver group, including ties. Superscripts on MoO are receiver-relative changes in percentage points; green indicates improvement.

Setting	Cap. Avg. ↑	ASR ↓	Over-ref. ↓	IFEval-I ↑
Qwen2.5-7B-Base Donor: Qwen2.5-1.5B-Instruct				
Receiver	0.7707	0.4400	**0.0200**	0.4862
Fixed $\gamma = 0.3$	**0.7837**	0.1300	0.0400	0.5078
Fixed $\gamma = 0.4$	0.7830	0.0800	0.0450	**0.5126**
Fixed $\gamma = 0.5$	0.7719	0.0650	0.0450	**0.5126**
MoO (hidden state)	0.7500$^{-2.07}$	**0.0200**$^{-42.00}$	0.0800$^{+6.00}$	0.4982$^{+1.20}$
Llama-3-8B-Base Donor: Llama-3.2-1B-Instruct				
Receiver	0.3299	1.0000	**0.0050**	0.3445
Fixed $\gamma = 0.3$	0.3984	**0.0100**	0.1200	0.3613
Fixed $\gamma = 0.4$	0.4288	**0.0100**	0.2150	0.3697
Fixed $\gamma = 0.5$	0.4462	**0.0100**	0.2350	0.3685
MoO (hidden state)	**0.4743**$^{+14.44}$	0.1700$^{-83.00}$	0.0550$^{+5.00}$	**0.4190**$^{+7.45}$
Gemma-2-9B-it Donor: Qwen2.5-1.5B-Instruct				
Receiver	0.6996	**0.0150**	0.1200	**0.7959**
Fixed $\gamma = 0.3$	0.2111	0.3450	0.1000	0.5630
Fixed $\gamma = 0.4$	0.1574	0.7150	0.1000	0.5102
Fixed $\gamma = 0.5$	0.1344	0.7850	**0.0900**	0.4718
MoO (hidden state)	**0.7136**$^{+1.40}$	**0.0150**$^{+0.00}$	0.1250$^{+0.50}$	0.7707$^{-2.52}$

D Gate-Diagnostic Protocols

D.1 Autoregressive Generation Traces

Figure 5 uses the Qwen math/code configuration in Table 11. One math prompt and one code prompt each produce 80 generated positions. The unsmoothed trajectories and their means are descriptive and do not enter the headline averages.

Table 11: **Generation-trace configuration and means for Figure 5.** Both prompts use seed 42 and $\gamma_{\max} = 1.0$; means average the 80 positions within each trace.

Component	Configuration
Receiver	Qwen2.5-7B-Instruct
Math donor	Qwen2.5-Math-7B-Instruct
Code donor	Qwen2.5-Coder-7B-Instruct
Prompts / generated positions	2 / 80 per prompt

Probe	Mean math-donor strength	Mean code-donor strength
Math	0.049	0.481
Code	0.245	0.394

D.2 Teacher-Forced Gate-Input Diagnostic

Table 3 uses the Llama reasoning configuration: 240 teacher-forced probes, 20,803 tokens, and 60 examples per benchmark (MATH500, GSM8K, ARC-C, and MMLU). The first two benchmarks provide 120 math probes; the other two provide 120 knowledge probes. It compares token embeddings, last hidden states, and causal means of receiver states, reporting domain means, the gap $\bar{\gamma}_{\text{math}} - \bar{\gamma}_{\text{know}}$, and the 10th percentile of token-level strengths. This pooled math/knowledge evaluation uses a different configuration from the Qwen math/code traces.