COMPOSITIONAL GENERALIZATION IN AUTOREGRESSIVE MODELS VIA LOGIT COMPOSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Composing autoregressive models remains a core challenge in understanding how large language models can combine behaviors or skills learned across tasks. We introduce a new and principled composition strategy for autoregressive systems, inspired by composition methods developed for diffusion models. Under a factorized-conditionals assumption, we show that the resulting composition is projective: each component model preserves control over its own designated subspace of the output distribution avoiding interference between models. We then identify the structure a change of coordinates must have for this property to persist in an abstract feature space: the feature map must be *causal*, that is, triangular with respect to the generation order. This is what distinguishes the autoregressive setting from the diffusion one, where arbitrary reparameterizations are admissible, and we show the requirement cannot be dropped. Since exact factorization is an idealization, we then show that it need only hold approximately: we introduce a *defect* measuring how far a family of components is from factorizing, prove that the divergence between the composition and its intended law is quadratic in the defect, and show that only what a component does outside its own region contributes to it. The defect is measurable from forward passes alone, and we report it for the checkpoints we compose. Finally, we show that composition preserves length-generalizing behavior when the factorization assumptions and component guarantees hold uniformly at the target length. These results provide a principled understanding of when model composition and merging succeed in autoregressive systems and identify conditions under which their interactions remain stable.

1 INTRODUCTION

Composing specialized generative models is an appealing path toward building systems that are more modular, reusable, and controllable than a single monolithic model. If one model captures a particular concept, another captures a different one, and a third provides a shared background, a natural question is whether these components can be combined *at inference time* to produce a model that behaves as though it had learned all concepts jointly.

For Transformer-based autoregressive models, existing approaches to this problem typically operate in either *parameter space* or *architecture space*. Parameter-space methods (such as weight interpolation Utans (1996); Wortsman et al. (2022), Task Arithmetic Ilharco et al. (2023), and sparsity-based merging Yadav et al. (2023); Yu et al. (2024)) combine fine-tuned checkpoints by averaging or adding task vectors, but provide limited guarantees on the resulting output distribution. Architecture-space approaches, such as Mixture-of-Experts (MoE) Jacobs et al. (1991); Shazeer et al. (2017), Fedus et al. (2022), combine models at the level of routing and activation, typically without a distinguished background model and with input-dependent selection. Together, these approaches do not provide a static, explicit composition rule at the level of output distributions with guarantees on the resulting autoregressive model.

A related line of work in diffusion models has demonstrated that models can be composed by *linearly combining their score functions* Du et al. (2023); Liu et al. (2022). This approach enables compositional generation and can even extrapolate beyond the support of individual models. Bradley et al. (2025) provided a principled explanation of when such composition is valid, introducing the notion of *projective composition* and identifying a factorization property as a key sufficient condition.

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.

However, this perspective has not been extended to autoregressive models, where generation is inherently sequential and conditioned on an evolving prefix.

Inspired by the projective composition framework of Bradley et al. (2025), we extend their formalism to autoregressive models such as Transformers. Given a collection of diffusion models $(p_1, \ldots, p_k, p_b)$, where p_b denotes a shared background distribution, Bradley et al. (2025) studied the composition operator

$$\mathcal{C}[(p_1, p_2, \ldots, p_k, p_b)] \equiv p_b \prod_{i=1}^{k} \frac{p_i}{p_b}. \tag{1}$$

They formalized the notion of correct composition and showed that the operator in Equation 1 yields a correct composition under certain conditions. Our *selective composition* extends this probability-space composition principle to autoregressive models such as Transformers, which form the basis of large language models (LLMs), by applying an explicit arithmetic rule to conditional next-token distributions. Sequential generation, however, is not neutral with respect to the choice of coordinates. Whereas the diffusion-space operator is globally normalized and hence invariant under arbitrary reparameterizations, its autoregressive counterpart renormalizes at every step, and we show that the invariance survives precisely for feature maps that are *causal*, i.e. triangular with respect to the generation order. The factorization these results assume is an idealization that fine-tuned checkpoints meet only approximately, so we also quantify what the approximation costs: the composition is stable in a defect measuring the failure of factorization, degrading quadratically in it, and the defect can be estimated from forward passes on the components themselves.

Beyond correctness, we establish a conditional length-generalization guarantee. In particular, when the factorization assumptions and component-model guarantees hold uniformly at a target length, the composed model inherits the corresponding behavior on the combined task. This highlights a key advantage of distribution-level composition: it preserves structural properties of the underlying models under explicit assumptions. Appendix A works through a motivating example, compositional reasoning in trigonometry, that illustrates what is at stake in the sequential setting.

Our Contributions

- We introduce a principled framework for composing autoregressive models in probability space, extending projective composition to the sequential setting.
- We identify conditions (factorized and joint factorizability) under which composition is provably correct.
- We characterize composition in an abstract feature space: the admissible reparameterizations are exactly the causal (triangular) ones, and we show by explicit example that causality can be dropped neither from the hypothesis nor from the conclusion.
- We show that the guarantees do not require exact factorization: a family of components within ϵ of one that factorizes composes within $O(\sqrt{\ell}\,\epsilon)$ in total variation of the law the exact theory prescribes, only the components' deviations outside their own regions enter this bound, and the rate is attained.
- We show that composition preserves length-generalizing behavior under uniform factorization and component-model guarantees.
- We connect our framework to model merging methods, providing insight into when and why such techniques succeed.
- We empirically validate our approach on both synthetic and large language model benchmarks, showing exact behavior in the controlled setting and, in LLM evaluations, coding gains with retention of mathematical performance, on a par with standard weight-space merging methods while requiring no access to weights.
- We introduce two diagnostics that measure directly how far real fine-tuned checkpoints are from the factorization assumptions, using only forward passes and no labels, and report what they show for the checkpoints we compose; the finer of the two measures exactly the quantity that enters the stability bound.

Paper organization: The rest of the paper is organized as follows. Section 2 introduces our compositional framework and presents our main theory. Section 3 summarizes empirical results on both controlled toy tasks and large language model benchmarks, and Section 4 concludes. Appendix B reviews related work on model merging and mixture-of-experts methods, Appendices D and E contain all proofs, Appendix F connects our framework to prior model merging approaches, and Appendix G reports the experiments in full.

2 Theory: Composing Transformers to Generalize

In this section, we describe our setup. We begin with a toy example that is closely related to the framework of Bradley et al. (2025). We then formalize the example introduced in Section 1. Finally, we present a result analogous to that of Bradley et al. (2025) in an abstract feature space. We denote by $p(y_t \mid x, y_{1:t-1})$ the token-wise distribution of a Transformer, which is a conditional distribution over a vocabulary $\mathcal{X} \subseteq \mathbb{R}^n$, where n is the embedding dimension. Consider a collection of transformer models $(p_b, p_1, \ldots, p_k)$, where each p_i represents a distinct concept, and p_b represents a common background model. We assume common support, so the ratios below are well-defined whenever a component assigns positive probability. We define a composed transformer $\hat{p}$, said to be a *selective composition* of $(p_b, p_1, \ldots, p_k)$, by

$$\hat{p}(y_t|x, y_{1:t-1}) := \frac{1}{Z_t(x, y_{<t})} p_b(y_t|x, y_{1:t-1}) \prod_{i=1}^{k} \frac{p_i(y_t|x, y_{1:t-1})}{p_b(y_t|x, y_{1:t-1})}, \tag{2}$$

where $Z_t(x, y_{<t})$ is the normalizing constant over $\mathcal{X}$. We study the compositional properties of $\hat{p}$. We begin with a simple illustrative example.

2.1 Composing logits under factorized conditionals

We illustrate the idea with a very simple example of letter-replacement Transformers. Consider three Transformer models p_1, p_2, and p_b, where the tokenization maps each character to a single token. Suppose p_b is trained to return the same string it was given as an input, i.e., an identity function. Suppose p_1 is trained to replace the character "A" with "B" in a given string, while leaving all other tokens unchanged. p_1 can be thought of as a fine-tuned version of the background p_b that can perform one extra task. Similarly, p_2 is trained to replace "C" with "D" while leaving all other tokens unchanged.

$$\begin{aligned} p_1 &: \text{A} \to \text{B} & T_1(\text{SETAJKHSDJH}) &= \text{SETBJKHSDJH}, \\ p_2 &: \text{C} \to \text{D} & T_2(\text{SETCJKHSDJH}) &= \text{SETDJKHSDJH}. \end{aligned}$$

If we compose the above Transformers according to Equation 2, **will the composed model be able to perform both transformations simultaneously?** In other words, will the composed model replace each "A" with "B" and each "C" with "D" in a given string? As we will see, this is indeed the case. We now formalize this idea to prove a result similar to Bradley et al. (2025) for diffusion models.

The idea is to define formally the property that the models and task are such that each model has its own predefined **range of action** depending on the input, where the model acts independently of the previous tokens. This is captured by the following definition. Here $\mathcal{X}^\ell$ is the space of length-ℓ generations.

Definition 1 (Factorized conditionals)**.** *The family of conditional distributions $(p_1, \ldots, p_k, p_b)$ on $\mathcal{X}^\ell$ (conditioned on some input x) satisfies the* factorized conditionals *property if there exists a partition of token indices of $\mathcal{X}^\ell$*

$$[\ell] = M_1 \uplus M_2 \uplus \cdots \uplus M_k \uplus M_b$$

such that the following conditions hold. For a generation $y = (y_1, \ldots, y_\ell) \in \mathcal{X}^\ell$, write y_M for the subsequence indexed by M. Then, for every $i \in \{1, \ldots, k\}$:

1. *y_{M_i} and $y_{M_i^c}$ are independent under $p_i(\cdot|x)$.*
2. *$y_{M_b}, y_{M_1}, \ldots, y_{M_k}$ are mutually independent under $p_b(\cdot|x)$.*
3. *$p_i(y_{M_i^c}|x) = p_b(y_{M_i^c}|x)$.*

Equivalently, the factorization may be written as

$$p_i(y|x) = p_i(y_{M_i}|x)\; p_b(y_{M_i^c}|x), \tag{3}$$

$$p_b(y|x) = p_b(y_{M_b}|x) \prod_{i=1}^{k} p_b(y_{M_i}|x). \tag{4}$$

The idea of Definition 1 is illustrated in Figure 2. p_1 and p_2 have their regions of action, and the composition $\hat{p}$ performs both tasks together. An important question is when to call a composition correct. Bradley et al. (2025) introduced the notion of *projective composition*, which we adapt for this particular setting.

Definition 2 (Projective Composition)**.** *Given a collection of distributions $\{p_i\}$ along with associated "projection" functions $\{\Pi_i : \mathcal{X}^\ell \to \mathbb{R}^m\}$, we call a distribution $\hat{p}$ a projective composition if*

$$\forall i: \qquad \Pi_i \# \hat{p} = \Pi_i \# p_i.$$

That is, when $\hat{p}$ is pushed forward by each Π_i, it yields marginals identical to those of p_i.

Note that here, Π_i's act on the entire generation in $\mathcal{X}^\ell$ and return some projection in some space $\mathbb{R}^m$. Having defined all the tools, we now state our result illustrating that Equation 2 indeed produces a composed model that exhibits the expected behavior.

Theorem 1 (Correctness of Composition)**.** *Suppose a set of autoregressive distributions $(p_1, \ldots, p_k, p_b)$ conditioned on some input variable x, satisfy Definition 1, with corresponding masks $\{M_i\}_{i=1}^k$ and M_b. Consider a sample $y \in \mathbb{R}^{n \times \ell}$ generated by the composed model $\hat{p}(y|x)$ as defined in Equation 2. Then the distribution of sample y generated by $\hat{p}$ is given by*

$$\hat{p}(y|x) = p_b(y_{M_b}|x) \prod_{i=1}^{k} p_i(y_{M_i}|x).$$

In particular, the marginal of $\hat{p}(\cdot|x)$ on M_i equals the marginal of $p_i(\cdot|x)$ on M_i for all i, and so $\hat{p}$ is a projective composition with respect to projections $\{\Pi_i(y) = y_{M_i}\}_{i=1}^k$, as per Definition 2.

Proof sketch. By Definition 1, each p_i coincides with p_b off its mask M_i, so at any timestep t exactly one mask (either some M_j or M_b) is active. In the product defining $\hat{p}$, all likelihood ratios cancel except for the active mask, yielding $\hat{p}(y_t|x, y_{<t}) = p_j(y_t|x, y_{<t})$ (or p_b if $t \in M_b$). Hence sampling proceeds by drawing each token from the unique distribution assigned to its mask, implying independence across disjoint masks. Therefore $\hat{p}(y|x) = p_b(y_{M_b}|x) \prod_{i=1}^k p_i(y_{M_i}|x)$ and the marginals on each M_i match p_i, establishing projective composition. For detailed proof, see Appendix D. □

In other words, Theorem 1 illustrates that whenever this independence among the tasks (Definition 1) holds, we indeed compose correctly.

2.2 Composing with dependence on the past

Having illustrated the compositional construction in the simple case of letter substitution in Subsection 2.1, we now turn to the motivating example of Appendix A. Consider a collection of trigonometric problems of the form Prove $X = Y$, where X and Y denote trigonometric expressions, and the objective is to establish their equivalence.

Assume that such problems can be partitioned into the following categories:

1. Problems solvable via elementary algebraic manipulations.
2. Problems solvable using the identity $\sin^2(x) + \cos^2(x) = 1$, in conjunction with elementary algebraic manipulations.
3. Problems solvable using the identity $\sec^2(x) - \tan^2(x) = 1$, in conjunction with elementary algebraic manipulations.

Suppose that a model p_b performs well on problems of type (1), while models p_1 and p_2 perform well on problems of types (2) and (3), respectively. The question of interest is to determine the conditions under which the composed model

$$\hat{p} = \frac{p_1 p_2}{p_b}$$

is effective at solving problems that require the simultaneous application of both identities $\sin^2(x) + \cos^2(x) = 1$ and $\sec^2(x) - \tan^2(x) = 1$. More generally, for k distinct concepts, we seek to characterize when a model $\hat{p}$, defined as in Equation 2, is capable of jointly leveraging all relevant components. It is important to note that the theoretical framework developed in Subsection 2.1 is not directly applicable in this setting. In particular, the behavior of p_1 is history-dependent: the applicability of the identity $\sin^2(x) + \cos^2(x) = 1$ at any given step depends on the sequence of transformations that have already been carried out in the evolving proof. To formalize this notion in the present setting, we introduce a property that we term *skill factorizability*. This notion is analogous to the property of factorized conditionals (Definition 1) introduced in Subsection 2.1. From here on, a model p on a task space T is understood to be a model whose inputs lie in the set T.

Definition 3 (Skill factorizability)**.** *For an autoregressive model p, a prompt $x \in T$ and continuation $y_{1:\ell} \in \mathcal{X}^\ell$, define*

$$\mathcal{L}_p(y_{1:\ell} \mid x) := \prod_{t=1}^{\ell} p(y_t \mid x, y_{<t}), \quad \textit{where } y_{<t} = (y_1, \ldots, y_{t-1}).$$

We say that a model p (on some task T), is skill factorizable *with respect to a base model p_b if for every prompt $x \in T$, there exist stopping times $\tau_1 \leq \tau_2$, such that, for all ℓ and all $y_{1:\ell} \in \mathcal{X}^\ell$,*

$$\mathcal{L}_p(y_{1:\ell} \mid x) = \left(\prod_{t=1}^{\tau_1} p_b(y_t \mid x, y_{<t}) \right) \left(\prod_{t=\tau_1+1}^{\tau_2} p(y_t \mid x, y_{<t}) \right) \left(\prod_{t=\tau_2+1}^{\ell} p_b(y_t \mid x, y_{<t}) \right),$$

where τ_1, τ_2 may depend on the realized prefix $y_{<t}$.

We also impose the following assumption: at any generation step, the task to be performed is uniquely determined by the generated prefix. In particular, there are no prefixes for which multiple distinct skills are simultaneously applicable. For example, in solving a trigonometric problem, if an identity is to be applied at a given step, then it is uniquely determined by the current expression; there are no steps at which multiple distinct identities could be applied. We formalize this in Definition 4.

Definition 4 (Joint factorizability)**.** *Let T be a task space and $(p_b, p_1, \ldots, p_k)$ be a collection of autoregressive models. We say that they are* jointly factorizable *on T if, for every prompt $x \in T$, the following hold:*

- *For each $i \in \{1, \ldots, k\}$, the model p_i is skill factorizable with respect to p_b on task T, with associated stopping times (τ_1^i, τ_2^i).*
- *The active index sets $I_i = \{\tau_1^i + 1, \ldots, \tau_2^i\}$ are pairwise disjoint, i.e.,*

$$I_i \cap I_j = \emptyset \quad \textit{for all } i \neq j.$$

We now show that if the models $(p_b, p_1, \ldots, p_k)$ satisfy the joint factorizability property (Definition 4) on some task T, then the composed model $\hat{p}$ defined by Equation 2 indeed composes the concepts correctly.

Theorem 2 (Composition under Joint factorizability)**.** *If models $(p_b, p_1, \ldots, p_k)$ satisfy joint factorizability on task T with active index sets $I_i = \{\tau_1^i + 1, \ldots, \tau_2^i\}$, then the composed model $\hat{p}$ defined by Equation (2) satisfies*

$$\mathcal{L}_{\hat{p}}(y_{1:\ell} \mid x) = \left(\prod_{t \notin \bigcup_{i=1}^k I_i} p_b(y_t \mid x, y_{<t}) \right) \left(\prod_{i=1}^{k} \prod_{t \in I_i} p_i(y_t \mid x, y_{<t}) \right).$$

In particular, $\hat{p}$ applies each skill-specific model p_i on its corresponding interval and agrees with the base model p_b elsewhere.

Proof sketch. At each timestep t, the unnormalized composed score is

$$\tilde{p}(y_t \mid x, y_{<t}) = p_b(y_t \mid x, y_{<t}) \prod_{i=1}^{k} \frac{p_i(y_t \mid x, y_{<t})}{p_b(y_t \mid x, y_{<t})}.$$

By joint factorizability, for any prefix $y_{<t}$ there is at most one index i such that $t \in I_i$. For all other $j \neq i$, we have

$$p_j(y_t \mid x, y_{<t}) = p_b(y_t \mid x, y_{<t}),$$

so all likelihood ratios cancel except possibly one. The resulting score is already normalized, and hence

$$\hat{p}(y_t \mid x, y_{<t}) = \begin{cases} p_i(y_t \mid x, y_{<t}), & t \in I_i, \\ p_b(y_t \mid x, y_{<t}), & \text{otherwise.} \end{cases}$$

Multiplying over t yields the desired factorization of $\mathcal{L}_{\hat{p}}$, showing that each model p_i is applied exactly on its interval and p_b elsewhere. See Appendix D for details. □

2.3 Composing logits under causal reparameterizations

Inspired by the result of Bradley et al. (2025) on diffusion models in feature space, we now present a version of Theorem 1 in which the factorized conditionals property (Definition 1) is required to hold only after a change of coordinates. This frees the component models from owning disjoint sets of token positions in the original space. Sequential generation, however, is not neutral with respect to the choice of coordinates: the feature map must respect the order in which tokens are produced. We therefore work with *causal* (equivalently, triangular) reparameterizations, the discrete counterpart of the Knothe–Rosenblatt rearrangement (Rosenblatt, 1952; Knothe, 1957; Villani, 2009) and of autoregressive normalizing flows.

Definition 5 (Causal reparameterization)**.** *A map $\mathcal{D} : \mathcal{X}^\ell \to \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_\ell$ is a* causal reparameterization *if there exist maps $\mathcal{D}_t : \mathcal{X}^t \to \mathcal{Z}_t$ such that*

1. *(Causality) $\mathcal{D}(y)_t = \mathcal{D}_t(y_{1:t})$ for every t and every y, and*
2. *(Step-wise invertibility) for every t and every prefix $y_{<t}$, the map $v \mapsto \mathcal{D}_t(y_{<t}, v)$ is a bijection from $\mathcal{X}$ onto $\mathcal{Z}_t$.*

When $\mathcal{X}$ is continuous we additionally require each $\mathcal{D}_t$ to be C^1 and each $v \mapsto \mathcal{D}_t(y_{<t}, v)$ to be a C^1 diffeomorphism, i.e. $\det \partial_v \mathcal{D}_t(y_{<t}, v) \neq 0$ everywhere.

Condition 1 says that the t-th feature coordinate may look at everything written so far but never at the future; condition 2 says that the re-encoding of the current token is reversible once the prefix is fixed. Together they say that $\mathcal{D}$ relabels each token by a bijection chosen as a function of the prefix, $\mathcal{D}_t(y_{1:t}) = \sigma_{y_{<t}}(y_t)$, and consequently that $\mathcal{D}$ is a global bijection whose Jacobian, in the continuous case, is block lower-triangular (Lemma 2 in Appendix D). For $\ell = 2$ and $\mathcal{X} = \{0, 1\}$, the map $\mathcal{D}(y_1, y_2) = (y_1, y_1 \oplus y_2)$ is causal, whereas the bijection $\mathcal{D}(y_1, y_2) = (y_1 \oplus y_2, y_2)$ is not, since its first coordinate reads the second token.

Theorem 3 (Feature-space Composition)**.** *Let $(p_1, \ldots, p_k, p_b)$ be autoregressive models of length ℓ conditioned on an input variable x, with common support, and let $\hat{p}$ be their composition as defined in Equation 2. Suppose there exist a causal reparameterization $\mathcal{D}$ (Definition 5) and a partition $[\ell] = M_1 \uplus \cdots \uplus M_k \uplus M_b$ of the feature coordinates such that the pushforward distributions*

$$\{\mathcal{D}\#\mathcal{L}_{p_1}, \ldots, \mathcal{D}\#\mathcal{L}_{p_k}, \mathcal{D}\#\mathcal{L}_{p_b}\}$$

satisfy Definition 1 with respect to that partition. Then $Z_t(x, y_{<t}) = 1$ for every t and every prefix, and

$$\mathcal{D}\#\mathcal{L}_{\hat{p}} = (\mathcal{D}\#\mathcal{L}_{p_b})_{M_b} \prod_{i=1}^{k} (\mathcal{D}\#\mathcal{L}_{p_i})_{M_i}.$$

Therefore, $\hat{p}$ is a projective composition of $(p_1, \ldots, p_k, p_b)$ with respect to the projection functions $\Pi_i(y) = \mathcal{D}(y)_{M_i}$ for $i = 1, \ldots, k$.

Proof sketch. Causality makes $\mathcal{D}$ commute with the left-to-right structure of generation: for every t, the first t feature coordinates and the first t tokens determine each other, so $\mathcal{D}$ carries every next-token conditional over unchanged, up to the relabelling of the token and, in the continuous case, up to a Jacobian factor that is the same for all models. Likelihood ratios in feature space therefore coincide with likelihood ratios in token space. Off its own block M_i, Definition 1 forces $\mathcal{D}\#\mathcal{L}_{p_i}$ and $\mathcal{D}\#\mathcal{L}_{p_b}$ to have identical conditionals; transporting this back gives $p_i(\cdot \mid x, y_{<t}) = p_b(\cdot \mid x, y_{<t})$ for every $t \notin M_i$ and every prefix. The token-level cancellation of Theorem 1 then applies verbatim, so $Z_t \equiv 1$ and $\hat{p}(\cdot \mid x, y_{<t}) = p_{m(t)}(\cdot \mid x, y_{<t})$, where $m(t)$ is the block containing t; the chain rule in feature space assembles these conditionals into the block product. See Appendix D for details. □

Two remarks delimit the scope of Theorem 3, both stated and substantiated in Appendix D.3: causality cannot be dropped, since a non-causal bijection whose pushforwards satisfy Definition 1 exactly makes both conclusions fail; and requiring it does not reduce Theorem 3 to Theorem 1, since the components may be arbitrarily entangled with the past in token coordinates as long as the entanglement is causally invertible.

The contrast with the diffusion setting is instructive. The operator of Equation 1 is globally normalized and 1-homogeneous, hence invariant under arbitrary reparameterizations (Bradley et al., 2025); Equation 2, by contrast, renormalizes at every generation step. Theorem 3 identifies the structure that a feature map must have for the two to agree, namely triangularity with respect to the generation order, and this is also exactly the structure under which $\hat{p}$ requires no global renormalization and remains implementable as per-step logit arithmetic.

2.4 Composing when factorization holds only approximately

Definitions 1 and 4 ask that, off its designated region, a component coincide exactly with the background model. Fine-tuned checkpoints do not: Appendix G.3 measures the discrepancy and finds it small but plainly nonzero. The question the three theorems above leave open is therefore not whether their hypotheses hold exactly, which they do not, but whether their conclusions degrade continuously when they fail. They do, at a rate quadratic in the discrepancy, and the only discrepancy that enters is the one off the designated regions.

Multiplying out Equation 2, the unnormalized score of the composition is $\prod_{i=1}^{k} p_i / p_b^{k-1}$. Two families of components therefore produce compositions that differ only through the log-ratio of these scores, and it is this combination, rather than the individual deviations, that measures how far apart the two settings are.

Definition 6 (Composition defect)**.** *Let $(p_1, \dots, p_k, p_b)$ and $(q_1, \dots, q_k, q_b)$ be two families of autoregressive models on $\mathcal{X}^\ell$ with common support, conditioned on a prompt x. For a step t, a prefix $y_{<t}$ and a token $v \in \mathcal{X}$, write*

$$\Delta_i(v) = \log q_i(v \mid x, y_{<t}) - \log p_i(v \mid x, y_{<t}), \qquad i \in \{1, \dots, k, b\},$$

and let

$$f_t(v \mid x, y_{<t}) = \sum_{i=1}^{k} \Delta_i(v) - (k-1)\Delta_b(v). \tag{5}$$

The defect *of the second family from the first at $(t, y_{<t})$ is the oscillation of this field over the vocabulary,*

$$E_t(x, y_{<t}) = \operatorname*{osc}_{v \in \mathcal{X}} f_t(v \mid x, y_{<t}) = \max_{v \in \mathcal{X}} f_t(v \mid x, y_{<t}) - \min_{v \in \mathcal{X}} f_t(v \mid x, y_{<t}),$$

and the uniform defect *is $E = \sup_{t \le \ell} \sup_{y_{<t}} E_t(x, y_{<t})$.*

If ϵ_i bounds $|\Delta_i|$ uniformly, then $E \le 2\big(\sum_{i=1}^{k} \epsilon_i + (k-1)\epsilon_b\big)$, so the defect is controlled by the individual deviations; taking the oscillation of the combined field rather than a sum of suprema keeps deviations that cancel between components from being charged twice, and makes the defect insensitive to how each conditional is normalized.

Theorem 4 (Stability of selective composition)**.** *Let $\hat{p}$ and $\hat{q}$ be the selective compositions of two families as in Definition 6. Then for every prompt x, every step t and every prefix $y_{<t}$,*

$$\max\Big\{\mathrm{KL}\big(\hat{p}(\cdot \mid x, y_{<t}) \,\big\|\, \hat{q}(\cdot \mid x, y_{<t})\big),\ \mathrm{KL}\big(\hat{q}(\cdot \mid x, y_{<t}) \,\big\|\, \hat{p}(\cdot \mid x, y_{<t})\big)\Big\} \ \le\ \frac{E_t(x, y_{<t})^2}{8},$$

and at the level of sequences,

$$\mathrm{KL}\big(\mathcal{L}_{\hat{p}}(\cdot \mid x) \,\big\|\, \mathcal{L}_{\hat{q}}(\cdot \mid x)\big) \;\le\; \frac{1}{8}\sum_{t=1}^{\ell} \mathbb{E}_{y_{<t}\sim\mathcal{L}_{\hat{p}}}\big[E_t(x, y_{<t})^2\big] \;\le\; \frac{\ell E^2}{8}, \qquad \mathrm{TV}\big(\mathcal{L}_{\hat{p}}, \mathcal{L}_{\hat{q}}\big) \le \frac{\sqrt{\ell}\, E}{4}.$$

Proof sketch. Because the two unnormalized scores differ by the factor e^{f_t}, the composed conditional $\hat{q}(\cdot \mid x, y_{<t})$ is the exponential tilt of $\hat{p}(\cdot \mid x, y_{<t})$ by f_t. The divergence between a distribution and its tilt is the gap between the logarithmic moment generating function of f_t at 1 and its derivative at 0, which Taylor's theorem writes as an average of the variances of f_t under the intermediate tilts; each such variance is at most $E_t^2/4$ since f_t ranges over an interval of that length. Summing over steps with the chain rule for the Kullback–Leibler divergence gives the sequence-level bound, and Pinsker's inequality the total variation one. See Appendix E for details. □

Taking the comparison family to be one that satisfies Definition 1, Definition 4 or the hypothesis of Theorem 3 turns Theorem 4 into an approximate version of each of the three results: a family within ϵ of one that factorizes composes into a distribution within $\sqrt{\ell}\,\epsilon/4$ in total variation of the law that the corresponding theorem prescribes, and is projective up to that error rather than exactly (Corollaries 1–3). The comparison family need not be guessed. Given a candidate assignment of regions to components, replacing each component by the background model off its own region produces one, and its defect involves only the off-region deviations $\log p_i - \log p_b$, which are exactly what the diagnostic of Equation 6 measures from forward passes; in the stopping-time setting of Theorem 2 this construction always satisfies the hypothesis, so no assumption on the components is left over. Appendix E develops this, sharpens the per-step bound to one governed by the variance of f_t rather than its range, exhibits a family for which the constant $1/8$ and the linear growth in ℓ are both attained, and instantiates the bound at our measured deviations, where it is informative per token and, at the tail of the deviation distribution, not at full generation length.

3 Experiments

The contribution of this work is primarily theoretical, and we evaluate it in three parts: a controlled setting in which the factorization assumptions hold by construction, a composition of two domain-specialized large language models against standard weight-space merging baselines, and a direct measurement of how far real checkpoints are from the assumptions. We summarize the three below; Appendix G reports each in full, with protocols, tables and per-benchmark numbers.

Composition is exact where the assumptions hold. We train three small decoder-only Transformers from scratch on the letter-replacement task of Subsection 2.1: a background model p_b implementing the identity map, an expert p_1 replacing every `A` with `K`, and an expert p_2 replacing every `M` with `B`. The designated regions are disjoint by construction, so Definition 1 holds and Theorem 1 predicts that $\hat{p}$ applies both substitutions and leaves every other character untouched. This is what we observe, with a 100% success rate on random unseen strings (Appendix G.1).

Composing specialized language models. We compose `Gemma 2 2B` (DeepMind, 2024) with two fine-tuned versions from `MergeBench` (He et al., 2026), one specialized for mathematics and one for coding, and evaluate on `GSM8K`, `MATH`, `HumanEval+` and `MBPP+` against the three component checkpoints, three weight-space merging methods (task arithmetic, TIES-merging and model soup) and the plain product of experts obtained by dropping the base-model division from Equation 2. Selective composition combines complementary experts, reaching 30.5 on `HumanEval+` and 24.2 on `MATH` where each expert is strong on one side and weak on the other. Its mean over the four benchmarks, 33.8, places it on a par with weight-space merging (33.6, 34.9 and 31.5) rather than ahead of it, and on `GSM8K` task arithmetic and TIES lead by roughly 13 points; what the operator of Equation 2 buys is not accuracy but that it works purely at the level of next-token distributions, requiring no access to weights, no shared parameterization and no merge coefficient to tune, at the cost of $k+1$ forward passes per step. Removing the base-model division costs 10.4 points on `HumanEval+` and 2.4 on `MBPP+`, so dividing out the $k-1$ redundant copies of the background model is not a cosmetic variant of a product of experts. Appendix G.2 reports the full table with binomial standard errors, the evaluation protocol, an asymmetry between the two domains that we

are not able to explain, and the sharp degradation we observe beyond two experts, which is the failure mode Remark 6 predicts.

How far real checkpoints are from the assumptions. Definitions 1 and 4 require that, off its designated region, an expert coincide with the background model, and by Corollary 4 that off-region deviation is the whole of what the defect of Definition 6 is made of. We therefore measure it directly, per token: for a prompt x, a continuation y and each expert i,

$$\epsilon_i(t) = \left|\log p_i(y_t \mid x, y_{<t}) - \log p_{\text{base}}(y_t \mid x, y_{<t})\right|, \tag{6}$$

read from one teacher-forced forward pass of each of the three models over 200 prompts and 37,532 generated positions, with no labels and no decoding. On 199 of the 200 prompts the in-domain expert deviates from the background model by more than the off-domain one ($p \approx 10^{-58}$, exact binomial against $1/2$), so the structure Definition 1 posits is present and detectable well above chance. The off-domain deviations are nonetheless small rather than zero, with means of 0.109 and 0.073 nats, so the assumption holds approximately and not exactly, which is the regime Theorem 4 covers and at which Appendix E.3 instantiates its bound. The two experts differ sharply in how far fine-tuning stayed inside its own domain, and a second, coarser diagnostic at the level of decoded text reaches the same conclusion; Appendix G.3 reports both, together with the magnitudes and quantiles that enter the bound.

4 Conclusion

We introduced a probability-space framework for composing autoregressive Transformer models, extending the notion of projective composition beyond diffusion models to sequential generation. Our approach provides a principled way to combine specialized models with a shared background model via a simple multiplicative rule over conditional distributions. We established conditions under which this composition is correct in token space, showed that these guarantees extend to settings with prefix-dependent behavior, and identified the causal (triangular) feature maps as exactly the changes of coordinates under which they transfer to an abstract feature space. None of these guarantees requires the factorization to be exact: the composition is stable in a defect that measures how far a family of components is from factorizing, with an error quadratic in the defect and driven only by what each component does outside its own region. Importantly, we showed that composition preserves length-generalizing behavior when the factorization assumptions and component-model guarantees hold uniformly at the target length. Further, we connected our framework to existing model merging methods and showed that several empirical heuristics can be explained in terms of our composition principle. Finally, experiments on synthetic tasks and large language model benchmarks show exact behavior in the controlled setting and evidence of complementary skill transfer, with coding gains and retention of mathematical performance, on a par with standard weight-space merging methods; and two diagnostics measure how far the composed checkpoints are from the factorization assumptions, finding the assumed structure present but only approximately satisfied. Overall, this work suggests that autoregressive model composition can be understood as a structured probabilistic operation with strong theoretical guarantees, opening the door to more modular and controllable construction of large language models.

Limitations. The factorization assumptions on which our analysis rests do not hold exactly for real checkpoints, and Theorem 4 is what removes the need for them to hold exactly: all it asks is that the defect be small, and the defect is something we can measure. Two gaps remain on that side. The defect is a worst case over the vocabulary and over prefixes, and the deviations we measure are heavy-tailed, so at a high quantile of the deviation and a realistic generation length the sequence-level form of the bound carries no information even where the per-token form does; the variance form of Proposition 2 is the natural remedy and we have not estimated it. And the theory still does not explain why pretraining, a process that may introduce interference, would leave the defect small in the first place. In addition, the empirical evaluation is limited in scope: it rests on a single base model and a single pair of experts, all three checkpoints share a parameterization, so the fact that our rule needs none is argued rather than demonstrated, and the two-expert regime is the scope of the empirical claims, since composition degrades sharply beyond it.

ETHICS STATEMENT

This work studies an inference-time rule for combining publicly released language models, evaluated on public benchmarks. It involves no human subjects, no new data collection and no personally identifiable information. Selective composition inherits the behavior of the models it is given: the mechanism that combines two useful skills combines undesirable ones just as readily, and dividing by the background model removes no harmful behavior that the components share. The guarantees we prove are about agreement with the component distributions, not about the content of those distributions, and should not be read as a safety property. The diagnostics of Section 3 require only forward passes, so a practitioner can inspect how far a candidate set of components is from the assumed structure before composing them, but they likewise say nothing about what the components generate.

REPRODUCIBILITY STATEMENT

All theoretical claims are stated with their full hypotheses in Section 2 and proved in Appendices D and E; Appendix C collects the notation used throughout, and the examples establishing that the hypotheses cannot be weakened are given explicitly in Appendix D.3. The experiments are summarized in Section 3 and specified in full in Appendix G, which gives the model identifiers, the training configuration for the controlled setting, the decoding settings and evaluation protocol, the merge configuration of every baseline, the prompt sets and token counts behind the two diagnostics, and the hardware used. An anonymized copy of the notebooks and scripts that produce every number reported here, together with a README listing dependencies and commands, is included in the supplementary material.

AI USE STATEMENT

Generative AI tools were used as a writing and editing aid for the manuscript and as an assistant for some of the experiments, where they helped write and debug parts of the experimental code. They were not used to generate research ideas, and every definition, theorem statement and proof was derived and verified by the authors. The code for all experiments is included in the supplementary material. We have reviewed all AI-assisted work and take responsibility for the final content of this paper, including its text, claims, code and results.

REFERENCES

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7(2):195–204, 2025.

Jacob Austin, Augustus Odena, Maxwell Nye, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

Joshua Belofsky. Token-level adaptation of lora adapters for downstream task generalization. In *Proceedings of the 2023 6th Artificial Intelligence and Cloud Computing Conference*, pp. 168–172, 2023.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013. doi: 10.1093/acprof:oso/9780199535255.001.0001.

Arwen Bradley, Preetum Nakkiran, David Berthelot, James Thornton, and Joshua M. Susskind. Mechanisms of projective composition of diffusion models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=QV0PcBbfTd.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Alexandra Chronopoulou, Matthew E Peters, Alexander Fraser, and Jesse Dodge. Adaptersoup: Weight averaging to improve generalization of pretrained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pp. 2054–2063, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Donato Crisostomi, Alessandro Zirilli, Antonio Andrea Gargiulo, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, Iacopo Masi, and Emanuele Rodolà. MASS: Merging through adaptive subspace selection. *arXiv preprint arXiv:2504.05342*, 2025.

MohammadReza Davari and Eugene Belilovsky. Model breadcrumbs: Scaling multi-task model merging with sparse masks. In *European Conference on Computer Vision*, pp. 270–287. Springer, 2024.

Google DeepMind. Gemma 2: Improving open language models at a practical size, 2024. URL https://ai.google.dev/gemma/docs/gemma-2. Technical report.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International conference on machine learning*, pp. 5547–5569. PMLR, 2022.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International conference on machine learning*, pp. 8489–8510. PMLR, 2023.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 2024. URL https://zenodo.org/records/12608602.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18695–18705. IEEE, June 2025. doi: 10.1109/cvpr52734.2025.01742. URL http://dx.doi.org/10.1109/CVPR52734.2025.01742.

Shwai He, Run-Ze Fan, Liang Ding, Li Shen, Tianyi Zhou, and Dacheng Tao. Merging experts into one: Improving computational efficiency of mixture of experts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14685–14691, 2023.

Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang, Tong Zhang, and Han Zhao. Mergebench: A benchmark for merging domain-specialized LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=rw50iUoyLu.

Dan Hendrycks, Collin Burns, Saurav Kadavath, et al. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021. URL https://arxiv.org/abs/2103.03874.

Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

Herbert Knothe. Contributions to the theory of convex bodies. *Michigan Mathematical Journal*, 4(1): 39–52, 1957. doi: 10.1307/mmj/1028990175.

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 Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12286–12312, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.687. URL https://aclanthology.org/2023.acl-long.687/.

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 Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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. 6691–6706, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.522. URL https://aclanthology.org/2021.acl-long.522/.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=1qvx610Cu7.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *European conference on computer vision*, pp. 423–439. Springer, 2022.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. *Advances in Neural Information Processing Systems*, 37:78905–78935, 2024.

Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Andrew D Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. In *Forty-second International Conference on Machine Learning*, 2025.

Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

Mohammed Muqeeth, Haokun Liu, Yufan Liu, and Colin Raffel. Learning to route among specialized experts for zero-shot generalization. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 36829–36846, 2024.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=0A9f2jZDGW.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 3982–3992. Association for Computational Linguistics, 2019. URL https://aclanthology.org/D19-1410/.

Murray Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952. doi: 10.1214/aoms/1177729394.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. *arXiv preprint arXiv:2410.19735*, 2024.

Qi Sun, Edoardo Cetin, and Yujin Tang. Transformer-squared: Self-adaptive llms. In *The Thirteenth International Conference on Learning Representations*, 2025.

Anke Tang, Li Shen, Yong Luo, Nan Yin, Lefei Zhang, and Dacheng Tao. Merging multi-task models via weight-ensembling mixture of experts. In *International Conference on Machine Learning*, pp. 47778–47799. PMLR, 2024.

Joachim Utans. Weight averaging for neural networks and local resampling schemes. In *Proc. AAAI-96 Workshop on Integrating Multiple Learned Models. AAAI Press*, pp. 133–138. Citeseer, 1996.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2009. doi: 10.1007/978-3-540-71050-9.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clément Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020. URL https://aclanthology.org/2020.emnlp-demos.6/.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Joshua M. Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? a study in length generalization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=AssIuHnmHX.

A A Motivating Example: Compositional Reasoning in Trigonometry

Consider a collection of trigonometric problems of the form Prove $X = Y$, where X and Y are trigonometric expressions. Suppose these problems can be decomposed into distinct skills (see Figure 1):

- algebraic manipulation,
- application of the identity $\sin^2(x) + \cos^2(x) = 1$,
- application of the identity $\sec^2(x) - \tan^2(x) = 1$.

Let p_b be a background Transformer that performs algebraic manipulations, while p_1 and p_2 are specialized models that apply the first and second identities, respectively. Each model is trained independently and excels only at its designated skill.

The key question is whether we can compose these models to get a model capable of solving problems that require *multiple identities in sequence*. For example, can the composed model solve a problem that requires first rewriting an expression using $\sin^2(x) + \cos^2(x) = 1$ and later applying $\sec^2(x) - \tan^2(x) = 1$, interleaved with algebraic steps?

This setting highlights the central challenge of autoregressive composition: the applicability of each skill depends on the *current prefix* of the generated solution. As we show in this work, under suitable conditions ensuring that these skills act on disjoint segments of the generation, the composed model successfully integrates all components and produces correct multi-step solutions.

B Related Work

Model Merging Recent research has explored how to combine pretrained models without joint retraining, a paradigm known as model merging or parameter arithmetic. Early approaches focused on simple weight-space interpolation between models fine-tuned on different tasks, showing that linear interpolation can sometimes yield models capable of retaining knowledge from both sources Utans (1996); Wortsman et al. (2022). A key insight from this line is Task Arithmetic Ilharco et al. (2023), which introduced the concept of *task vectors* $\tau_t = \theta(t) - \theta(0)$, where $\theta(t)$ denotes parameters fine-tuned on task t and $\theta(0)$ is the pretrained model. Task vectors are interpreted as directions in parameter space that meaningfully steer model behavior, enabling arithmetic operations across fine-tuned models.

Subsequent work has focused on reducing interference between tasks when performing such parameter arithmetic, for example by choosing merging coefficients more effectively Akiba et al. (2025); Yang et al. (2024); Matena & Raffel (2022); Jin et al. (2023). Another line of research uses sparsification to mitigate task interference Yadav et al. (2023); Yang et al. (2024); Yu et al. (2024); Davari & Belilovsky (2024). Other work uses low-rank subspaces and matrix factorization to extract informative components for merging Gargiulo et al. (2025); Marczak et al. (2025); Stoica et al. (2024). More recent geometric approaches perform fine-tuning and merging in the tangent space of the pretrained model manifold, linearizing updates to better capture local structure, amplify weight disentanglement, and mitigate task interference Ortiz-Jimenez et al. (2023).

Our work connects to the model merging literature because we also compose a pretrained base (background) model with task-specific fine-tuned models, similar to Task Arithmetic Ilharco et al. (2023) and subsequent merging methods. However, unlike parameter-space or task-space techniques, our *selective composition* operates in *probability space*: it explicitly combines output distributions of component models while preserving their designated subspaces. This makes the comparison to mixture-of-experts methods useful, although the mechanism is different.

Mixture of Experts in Large Language Models Classical Mixture-of-Experts (MoE) architectures augment neural networks with multiple experts and a routing mechanism that selects a subset of experts per input, improving performance and efficiency by allocating capacity adaptively rather than relying on a single monolithic model Jacobs et al. (1991); Jordan & Jacobs (1994); Shazeer et al. (2017). In these settings there is typically no distinguished base model; instead, all experts are trained

jointly and symmetrically, and a gating network learns to route inputs to suitable experts to reduce error and computational cost Fedus et al. (2022); Du et al. (2022).

In the context of model merging, this idea has inspired *routing-based merging* methods, which replace static, input-agnostic aggregation with adaptive, sample-dependent combinations of models or modules. Instead of using a fixed merged model for all inputs, these approaches dynamically decide how to combine task-specific components at inference time He et al. (2023); Tang et al. (2024); Lu et al. (2024); Crisostomi et al. (2025); Sun et al. (2025); Belofsky (2023); Chronopoulou et al. (2023); Muqeeth et al. (2024).

Our *selective composition* differs from both classical MoE and routing-based merging in several key aspects. Unlike MoE's learned gating network or routing-based merging's input-conditioned coefficients, our composition is entirely static: all component models contribute simultaneously without dynamic selection. Moreover, classical MoE selects entire expert subnetworks via routing; routing-based merging dynamically weights parameter subspaces. We instead perform explicit arithmetic combination of output probability distributions. While MoE relies on routing to avoid interference, our framework gives conditions under which each component preserves control over its designated output subspace.

Finally, our *selective composition* shares conceptual similarity with diffusion model composition methods, where models are composed by linearly combining their scores Du et al. (2023); Liu et al. (2022). The literature on composing Transformers through their logits is more limited. Liu et al. (2021) propose DExperts, which steers a base model by adding and subtracting expert and anti-expert logits in a product-of-experts formulation, enabling controlled generation without retraining. Similarly, Li et al. (2023) introduce contrastive decoding, which selects tokens by maximizing the difference between an expert and a weaker "amateur" model's log-probabilities under plausibility constraints. These approaches demonstrate that simple logit-level composition can effectively modulate generation behavior, motivating our formulation.

C Notation

Table 1 summarizes the main notation used in Section 2 and in the proofs of Appendix D and Appendix E.

D Main Theory

In this section, we provide rigorous proof of all the results in the paper. We start with Theorem 1.

Proof of Theorem 1. Fix the conditioning input x. For each position $t \in [\ell]$, let $m(t) \in \{b, 1, \dots, k\}$ denote the unique index such that $t \in M_{m(t)}$. Since $\{M_1, \dots, M_k, M_b\}$ is a partition of $[\ell]$, such an index is well defined.

By Definition 1, for every $i \in \{1, \dots, k\}$, the distribution p_i agrees with p_b on all coordinates outside M_i. Equivalently, for every $t \notin M_i$,

$$p_i(y_t \mid x, y_{1:t-1}) = p_b(y_t \mid x, y_{1:t-1}).$$

Therefore, if $t \in M_j$ for some $j \in \{1, \dots, k\}$, then

$$\frac{p_i(y_t \mid x, y_{1:t-1})}{p_b(y_t \mid x, y_{1:t-1})} = \begin{cases} 1, & i \neq j, \\ \dfrac{p_j(y_t \mid x, y_{1:t-1})}{p_b(y_t \mid x, y_{1:t-1})}, & i = j. \end{cases}$$

Hence the unnormalized composition score simplifies to

$$\begin{aligned} \tilde{p}(y_t \mid x, y_{1:t-1}) &= p_b(y_t \mid x, y_{1:t-1}) \prod_{i=1}^{k} \frac{p_i(y_t \mid x, y_{1:t-1})}{p_b(y_t \mid x, y_{1:t-1})} \\ &= \begin{cases} p_j(y_t \mid x, y_{1:t-1}), & t \in M_j,\ j \in \{1, \dots, k\}, \\ p_b(y_t \mid x, y_{1:t-1}), & t \in M_b. \end{cases} \end{aligned}$$

Symbol	Meaning
$\mathcal{X}$	Token vocabulary; $\mathcal{X}^\ell$ is the space of length-ℓ generations.
x	Conditioning input or prompt.
$y_{1:\ell}$, y_t, $y_{<t}$	A generated sequence, its token at position t, and the prefix before t.
p_b	Background or base autoregressive model.
p_i	The ith specialized autoregressive model, for $i \in \{1, \dots, k\}$.
$\hat{p}$	Composed autoregressive model defined by Equation 2.
$Z_t(x, y_{<t})$	Normalizing constant for the composed conditional distribution at step t.
$\mathcal{L}_p(y_{1:\ell} \mid x)$	Sequence-level likelihood induced by an autoregressive model p.
M_i, M_b	Token-index masks partitioning $[\ell]$ into expert and background coordinates.
y_{M_i}	Subsequence of y restricted to indices in M_i.
Π_i	Projection map used to define projective composition.
$\Pi_i \# p$	Pushforward of distribution p through projection Π_i.
τ_1^i, τ_2^i	Stopping times delimiting the active region of skill i.
I_i	Active interval $\{\tau_1^i + 1, \dots, \tau_2^i\}$ for skill i.
$\mathcal{D}$, $\mathcal{Z}_t$	Causal (triangular) feature map of Definition 5 and its tth feature alphabet.
$m(t)$	Index of the unique block containing position t.
$\mathcal{C}[\cdot]$	Distribution-level composition operator used in Appendix D.
Δ_i, f_t	Log-ratio of two families at component i, and the combined field of Equation 5.
osc	Oscillation of a function on $\mathcal{X}$, that is, its maximum minus its minimum.
E_t, E	Composition defect at step t (Definition 6) and its supremum over steps and prefixes.
λ_i	Deviation $\log p_i - \log p_b$ of component i from the background model, used in Appendix E.3.
KL, TV	Kullback–Leibler divergence and total variation distance.

Table 1: Notation used in the theoretical results and proofs.

The right-hand side is already a normalized conditional distribution over the token vocabulary, so the normalizing constant in Equation 2 equals one at this prefix.

Thus, at each position t, the composed model uses exactly the conditional distribution associated with the unique block containing t.

Now apply the chain rule:

$$\hat{p}(y \mid x) = \prod_{t=1}^{\ell} \hat{p}(y_t \mid x, y_{1:t-1}).$$

Substituting the expression above yields

$$\hat{p}(y \mid x) = \left(\prod_{t \in M_b} p_b(y_t \mid x, y_{1:t-1}) \right) \prod_{i=1}^{k} \left(\prod_{t \in M_i} p_i(y_t \mid x, y_{1:t-1}) \right).$$

By the factorized conditionals property, this equals

$$\hat{p}(y \mid x) = p_b(y_{M_b} \mid x) \prod_{i=1}^{k} p_i(y_{M_i} \mid x).$$

In particular, for each i, the marginal of $\hat{p}$ on M_i coincides with $p_i(\cdot \mid x)$. Therefore $\hat{p}$ is a projective composition with respect to the projections $\Pi_i(y) = y_{M_i}$, as per Definition 2. □

We now come to Theorem 2.

Proof of Theorem 2. Fix an arbitrary prompt $x \in T$ and an arbitrary continuation $y_{1:\ell} \in \mathcal{X}^\ell$. For notational simplicity, write

$$p_{i,t} := p_i(y_t \mid x, y_{<t}), \qquad p_{b,t} := p_b(y_t \mid x, y_{<t}), \qquad \hat{p}_t := \hat{p}(y_t \mid x, y_{<t}).$$

Assume that the ratios $p_{i,t}/p_{b,t}$ are well-defined on the common support.

By definition of the composed model, before normalization,

$$\tilde{p}_t = p_{b,t} \prod_{i=1}^{k} \frac{p_{i,t}}{p_{b,t}}.$$

Let

$$I_i := \{\tau_1^i + 1, \tau_1^i + 2, \ldots, \tau_2^i\}$$

denote the active interval of skill i. Since the intervals $\{I_i\}_{i=1}^k$ are pairwise disjoint, for every t there is at most one index i such that $t \in I_i$.

If $t \in I_i$ for some i, then for every $j \neq i$, we are outside the active interval of p_j, hence

$$p_{j,t} = p_{b,t}.$$

Therefore,

$$\tilde{p}_t = p_{b,t} \left(\frac{p_{i,t}}{p_{b,t}} \right) \prod_{j \neq i} \frac{p_{j,t}}{p_{b,t}} = p_{b,t} \left(\frac{p_{i,t}}{p_{b,t}} \right) = p_{i,t}.$$

If instead $t \notin \bigcup_{i=1}^k I_i$, then for all i we have $p_{i,t} = p_{b,t}$, and hence

$$\tilde{p}_t = p_{b,t} \prod_{i=1}^{k} \frac{p_{b,t}}{p_{b,t}} = p_{b,t}.$$

In both cases the unnormalized score is already a normalized conditional distribution, so the normalizing constant in Equation 2 equals one. Thus, for every $t \in \{1, \ldots, \ell\}$,

$$\hat{p}(y_t \mid x, y_{<t}) = \begin{cases} p_i(y_t \mid x, y_{<t}), & t \in I_i \text{ for some } i, \\ p_b(y_t \mid x, y_{<t}), & t \notin \bigcup_{i=1}^k I_i. \end{cases}$$

Multiplying over $t = 1, \ldots, \ell$ yields

$$\mathcal{L}_{\hat{p}}(y_{1:\ell} \mid x) = \prod_{t=1}^{\ell} \hat{p}(y_t \mid x, y_{<t}) = \left(\prod_{t \notin \bigcup_{i=1}^k I_i} p_b(y_t \mid x, y_{<t}) \right) \left(\prod_{i=1}^{k} \prod_{t \in I_i} p_i(y_t \mid x, y_{<t}) \right).$$

This is exactly the claimed factorization, and shows that $\hat{p}$ applies each skill-specific model p_i on its interval and p_b elsewhere. □

We now come to the results in feature space. Throughout this part we fix the prompt x and suppress it from the notation, and we write $\mathcal{L}_p(y) = \prod_{t=1}^{\ell} p(y_t \mid y_{<t})$ for the sequence-level law of an autoregressive model p, so that the conditionals of $\mathcal{L}_p$ recover p. Since Theorem 3 compares $\mathcal{L}_{\hat{p}}$ with the globally normalized operator of Equation 1, we begin by recording the exact relation between the two.

Proposition 1. *Let $\hat{p}$ be defined by Equation 2 and set $W(y) := \prod_{t=1}^{\ell} Z_t(y_{<t})$. Then*

$$\mathcal{L}_{\hat{p}}(y) = \frac{1}{W(y)} \mathcal{L}_{p_b}(y) \prod_{i=1}^{k} \frac{\mathcal{L}_{p_i}(y)}{\mathcal{L}_{p_b}(y)},$$

and $\mathcal{L}_{\hat{p}} = \mathcal{C}[\mathcal{L}_{p_1}, \ldots, \mathcal{L}_{p_k}, \mathcal{L}_{p_b}]$ if and only if W is constant on $\mathcal{X}^\ell$.

Proof. By the chain rule and Equation 2,

$$\mathcal{L}_{\hat{p}}(y) = \prod_{t=1}^{\ell} \hat{p}(y_t \mid y_{<t}) = \prod_{t=1}^{\ell} \frac{1}{Z_t(y_{<t})} p_b(y_t \mid y_{<t}) \prod_{i=1}^{k} \frac{p_i(y_t \mid y_{<t})}{p_b(y_t \mid y_{<t})}.$$

Regrouping the product over t by model index, the model-indexed factors assemble into $\mathcal{L}_{p_b}(y)\prod_i \mathcal{L}_{p_i}(y)/\mathcal{L}_{p_b}(y)$ and the normalizers into $1/W(y)$, which is the displayed identity. Write $G(y)$ for the numerator, so $\mathcal{L}_{\hat{p}} = G/W$ while $\mathcal{C}[\mathcal{L}_{p_1}, \ldots, \mathcal{L}_{p_b}] = G/Z$ with $Z = \sum_y G(y)$ a single number. If $W \equiv c$, both are distributions proportional to the same positive function and hence equal. Conversely, if they are equal then $G/W = G/Z$ pointwise and $G > 0$ by common support, so $W \equiv Z$. □

Note that W is constant in the settings of Theorems 1 and 2: both proofs establish $Z_t \equiv 1$ at every prefix, so $W \equiv 1$ there. This is a statement about individual conditionals, and it is what the causality requirement of Definition 5 recovers in the feature-space setting. We record this first as a standalone step.

Lemma 1. *Suppose $\{M_1, \ldots, M_k, M_b\}$ partitions $[\ell]$ and that for every $i \in \{1, \ldots, k\}$, every $t \notin M_i$ and every prefix $y_{<t}$ we have $p_i(\cdot \mid y_{<t}) = p_b(\cdot \mid y_{<t})$. Then, for every t and every prefix,*

$$Z_t(y_{<t}) = 1 \qquad \text{and} \qquad \hat{p}(\cdot \mid y_{<t}) = p_{m(t)}(\cdot \mid y_{<t}),$$

where $m(t)$ is the unique index with $t \in M_{m(t)}$, and $p_{m(t)}$ means p_b when $m(t) = b$.

Proof. Fix t, a prefix $y_{<t}$, and set $c = m(t)$. For every $i \neq c$ in $\{1, \ldots, k\}$ we have $t \notin M_i$, so $p_i(v \mid y_{<t})/p_b(v \mid y_{<t}) = 1$ for all $v \in \mathcal{X}$. Hence the unnormalized score equals $p_c(v \mid y_{<t})$ if $c \in \{1, \ldots, k\}$, and $p_b(v \mid y_{<t})$ if $c = b$. In both cases it is already a distribution on $\mathcal{X}$, so $Z_t(y_{<t}) = 1$ and $\hat{p} = p_c$. This is the token-level cancellation used in the proof of Theorem 1. □

D.1 Causal reparameterizations preserve the autoregressive structure

Let $\mathcal{D}$ be a causal reparameterization in the sense of Definition 5, with components $\mathcal{D}_t$, and write $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_\ell$. Throughout, y and z denote corresponding points, $z = \mathcal{D}(y)$. We state the discrete case; Remark 1 treats the continuous one.

Lemma 2 (Causal maps are bijections, with causal inverse)**.** *For $t \in [\ell]$ define $\mathcal{D}_{1:t}(y_{1:t}) := (\mathcal{D}_1(y_1), \ldots, \mathcal{D}_t(y_{1:t}))$. Then each $\mathcal{D}_{1:t}$ is a bijection onto $\mathcal{Z}_1 \times \cdots \times \mathcal{Z}_t$. In particular $\mathcal{D} = \mathcal{D}_{1:\ell}$ is a bijection, and $(\mathcal{D}^{-1}(z))_{1:t}$ depends only on $z_{1:t}$.*

Proof. Induction on t. For $t = 1$, $\mathcal{D}_{1:1} = \mathcal{D}_1$ is a bijection by Definition 5(2) with the empty prefix. Assume $\mathcal{D}_{1:t-1}$ is a bijection. If $\mathcal{D}_{1:t}(y_{1:t}) = \mathcal{D}_{1:t}(y'_{1:t})$, comparing the first $t-1$ coordinates gives $y_{<t} = y'_{<t}$ by the inductive hypothesis, and comparing the t-th coordinate gives $y_t = y'_t$ by Definition 5(2); so $\mathcal{D}_{1:t}$ is injective. Given $z_{1:t}$, the inductive hypothesis produces the unique $y_{<t}$ with $\mathcal{D}_{1:t-1}(y_{<t}) = z_{<t}$, and $y_t := \mathcal{D}_t(y_{<t}, \cdot)^{-1}(z_t)$ completes the preimage; so $\mathcal{D}_{1:t}$ is surjective. The construction computes $y_{1:t}$ from $z_{1:t}$ alone, which is the stated causality of $\mathcal{D}^{-1}$. □

Lemma 3 (Conditional invariance)**.** *Let P be a distribution on $\mathcal{X}^\ell$ with full support and let $q := \mathcal{D}\#P$. Then for every t and every pair of corresponding points $z = \mathcal{D}(y)$,*

$$q(z_{1:t}) = P(y_{1:t}) \qquad \text{and hence} \qquad q(z_t \mid z_{<t}) = P(y_t \mid y_{<t}).$$

Proof. By definition of marginal and pushforward, $q(z_{1:t}) = \Pr_{y' \sim P}[\mathcal{D}(y')_{1:t} = z_{1:t}]$. By causality, $\mathcal{D}(y')_{1:t} = \mathcal{D}_{1:t}(y'_{1:t})$ depends on y' only through $y'_{1:t}$, and by Lemma 2 the map $\mathcal{D}_{1:t}$ is injective, so the event is exactly $\{y'_{1:t} = y_{1:t}\}$ and the probability equals $P(y_{1:t})$. Dividing the identities at t and $t-1$ gives the second claim. □

This is the only place where causality is used, and it is essential: without it, $\mathcal{D}(y')_{1:t}$ may depend on tokens generated after step t, and then $\{\mathcal{D}(y')_{1:t} = z_{1:t}\}$ is not an event about the first t tokens at all. An immediate consequence is that likelihood ratios are the same in both spaces: for autoregressive models p, p' with common support and $q = \mathcal{D}\#\mathcal{L}_p$, $q' = \mathcal{D}\#\mathcal{L}_{p'}$,

$$\frac{q(z_t \mid z_{<t})}{q'(z_t \mid z_{<t})} = \frac{p(y_t \mid y_{<t})}{p'(y_t \mid y_{<t})}. \tag{7}$$

Finally, we extract from Definition 1, read in feature space, the two facts about conditionals that the proof needs. Below, $q_i := \mathcal{D}\#\mathcal{L}_{p_i}$ and $q_b := \mathcal{D}\#\mathcal{L}_{p_b}$ satisfy Definition 1 on $\mathcal{Z}$ with partition $\{M_1, \ldots, M_k, M_b\}$.

Lemma 4 (Block-local conditionals). *For every $t \in [\ell]$ and every $z \in \mathcal{Z}$:*

1. *if $t \in M_i$ with $i \in \{1, \dots, k\}$, then $q_i(z_t \mid z_{<t}) = q_i(z_t \mid z_{M_i \cap [t-1]})$;*
2. *if $t \notin M_i$ with $i \in \{1, \dots, k\}$, then $q_i(z_t \mid z_{<t}) = q_b(z_t \mid z_{<t})$;*
3. *$q_b(z_t \mid z_{<t}) = q_b(z_t \mid z_{M_{m(t)} \cap [t-1]})$.*

Proof. Write $S = [t]$ and $S' = [t-1]$. Marginalizing Equation 3 to S and to S' (the product structure passes to marginals, by the independence in Definition 1) gives

$$q_i(z_S) = q_i(z_{S \cap M_i})\, q_b(z_{S \cap M_i^c}), \qquad q_b(z_S) = \prod_{B \in \{M_b, M_1, \dots, M_k\}} q_b(z_{S \cap B}),$$

and likewise with S' in place of S.

(1) If $t \in M_i$, then $S \cap M_i^c = S' \cap M_i^c$, so dividing the two identities for q_i cancels the q_b factors and leaves $q_i(z_t \mid z_{S' \cap M_i})$, which is the claim.

(3) With $c = m(t)$, we have $S \cap B = S' \cap B$ for every block $B \neq M_c$, so dividing the two identities for q_b cancels all factors with $B \neq M_c$ and leaves $q_b(z_t \mid z_{M_c \cap [t-1]})$.

(2) If $t \notin M_i$, then $S \cap M_i = S' \cap M_i$, so dividing the two identities for q_i cancels the q_i factors and leaves $q_b(z_t \mid z_{M_i^c \cap [t-1]})$. Since t lies in the block $M_c \subseteq M_i^c$ and M_i^c is a union of whole blocks, marginalizing the identity for q_b to $M_i^c \cap [t]$ and dividing as in (3) gives $q_b(z_t \mid z_{M_i^c \cap [t-1]}) = q_b(z_t \mid z_{M_c \cap [t-1]}) = q_b(z_t \mid z_{<t})$. □

D.2 Proof of Theorem 3

Proof of Theorem 3. **Step 1: transport the off-block agreement back to token space.** Fix $i \in \{1, \dots, k\}$, a position $t \notin M_i$, a prefix $y_{<t}$ with corresponding feature prefix $z_{<t}$ (Lemma 2), and an arbitrary $v \in \mathcal{X}$ with $w \coloneqq \mathcal{D}_t(y_{<t}, v)$. Lemma 4(2) gives $q_i(w \mid z_{<t}) = q_b(w \mid z_{<t})$, and Equation 7 turns this into

$$\frac{p_i(v \mid y_{<t})}{p_b(v \mid y_{<t})} = \frac{q_i(w \mid z_{<t})}{q_b(w \mid z_{<t})} = 1.$$

As v was arbitrary, $p_i(\cdot \mid y_{<t}) = p_b(\cdot \mid y_{<t})$ whenever $t \notin M_i$.

Step 2: step-wise selection. Step 1 is exactly the hypothesis of Lemma 1, so $Z_t(y_{<t}) = 1$ for every t and every prefix, and $\hat{p}(\cdot \mid y_{<t}) = p_{m(t)}(\cdot \mid y_{<t})$. In particular $W \equiv 1$ in Proposition 1, so $\mathcal{L}_{\hat{p}} = \mathcal{C}[\mathcal{L}_{p_1}, \dots, \mathcal{L}_{p_k}, \mathcal{L}_{p_b}]$.

Step 3: the block product in feature space. Let $\hat{q} \coloneqq \mathcal{D} \# \mathcal{L}_{\hat{p}}$. Applying Lemma 3 to $\mathcal{L}_{\hat{p}}$, then Step 2, then Lemma 3 to $\mathcal{L}_{p_{m(t)}}$ in the other direction,

$$\hat{q}(z_t \mid z_{<t}) = \hat{p}(y_t \mid y_{<t}) = p_{m(t)}(y_t \mid y_{<t}) = q_{m(t)}(z_t \mid z_{<t}).$$

The chain rule on $\mathcal{Z}$, grouping positions by their block, then gives

$$\hat{q}(z) = \prod_{t=1}^{\ell} q_{m(t)}(z_t \mid z_{<t}) = \left(\prod_{t \in M_b} q_b(z_t \mid z_{<t}) \right) \prod_{i=1}^{k} \left(\prod_{t \in M_i} q_i(z_t \mid z_{<t}) \right).$$

Replacing each conditional by its block-local version using Lemma 4(3) for $t \in M_b$ and Lemma 4(1) for $t \in M_i$, each inner product becomes a chain-rule expansion of the marginal on that block and telescopes, yielding

$$\hat{q}(z) = q_b(z_{M_b}) \prod_{i=1}^{k} q_i(z_{M_i}),$$

which is the claimed feature-space factorization.

Step 4: projectivity. By Step 3 the blocks of $\hat{q}$ are mutually independent and its marginal on M_i is $(q_i)_{M_i}$. Moreover $\Pi_i = \rho_{M_i} \circ \mathcal{D}$ with $\rho_{M_i}(z) = z_{M_i}$ coordinate restriction, and pushing forward by a coordinate restriction is taking the marginal, so $\Pi_i \# P = (\mathcal{D} \# P)_{M_i}$ for any P. Applying this to $P = \mathcal{L}_{\hat{p}}$ and to $P = \mathcal{L}_{p_i}$ gives $\Pi_i \# \mathcal{L}_{\hat{p}} = \hat{q}_{M_i} = (q_i)_{M_i} = \Pi_i \# \mathcal{L}_{p_i}$, which is Definition 2. □

Remark 1 (Continuous tokens). *When $\mathcal{X} = \mathbb{R}^n$ and $\mathcal{D}$ is as in the continuous part of Definition 5, causality makes the Jacobian $J_{\mathcal{D}}(y)$ block lower-triangular, so $\det J_{\mathcal{D}}(y) = \prod_t \det \partial_{y_t} \mathcal{D}_t(y_{1:t}) \neq 0$ and $\mathcal{D}$ is a global diffeomorphism by the induction of Lemma 2. The change-of-variables formula then turns Lemma 3 into*

$$q(z_t \mid z_{<t}) = \frac{P(y_t \mid y_{<t})}{|\det \partial_{y_t} \mathcal{D}_t(y_{1:t})|},$$

where the Jacobian factor depends on $\mathcal{D}$ alone and not on P. It therefore cancels in Equation 7, and the proof above goes through verbatim with sums over $\mathcal{X}$ replaced by integrals in the definition of Z_t. The step-wise diffeomorphism requirement is what makes this available: an invertible C^1 map need not admit a change-of-variables formula for densities, as $v \mapsto v^3$ on $\mathbb{R}$ shows.

D.3 Causality is necessary, and the resulting theorem is not a corollary of Theorem 1

Remark 2 (Causality cannot be dropped). *Appendix D.3 exhibits a family with $\ell = 2$ and $k = 2$ and the non-causal bijection $\mathcal{D}(y_1, y_2) = (y_1 \oplus y_2, y_2)$ whose pushforwards satisfy Definition 1 exactly, and for which both conclusions fail: Z_2 depends on the prefix, and the marginal of $\hat{p}$ along Π_1 assigns mass 0.744 where p_1 assigns 0.9. What causality excludes is precisely the mixing of positions by $\mathcal{D}$.*

Remark 3 (Strictly more general than Theorem 1). *Requiring $\mathcal{D}$ to be causal does not reduce Theorem 3 to Theorem 1. The component models may be arbitrarily entangled with the past in token coordinates, as long as the entanglement is causally invertible. Appendix D.3 gives models satisfying the hypothesis of Theorem 3 with $\mathcal{D}(y_1, y_2) = (y_1, y_1 \oplus y_2)$ for which both the hypothesis and the conclusion of Theorem 1 fail.*

We substantiate Remarks 2 and 3 with a single pair of examples. Let $\mathcal{X} = \{0, 1\}$, $\ell = 2$, $k = 2$, let a be the distribution on $\{0, 1\}$ with $a(0) = 0.9$ and $a(1) = 0.1$, and take masks $M_1 = \{1\}$, $M_2 = \{2\}$, $M_b = \varnothing$ and feature-space laws

$$q_b(z) = \tfrac{1}{4}, \qquad q_1(z) = a(z_1) \cdot \tfrac{1}{2}, \qquad q_2(z) = \tfrac{1}{2} \cdot a(z_2),$$

which satisfy Definition 1 exactly: each q_i is a product across $\{1\}$ and $\{2\}$, each agrees with q_b on the complement of its own block, and q_b is a product.

A non-causal map breaks the conclusion. Take $\mathcal{D}(y_1, y_2) = (y_1 \oplus y_2, y_2)$, a bijection of $\{0, 1\}^2$ that is not causal since its first coordinate reads y_2. As $\mathcal{D}$ is an involution, the induced token-space laws are $\mathcal{L}_{p_b}(y) = \frac{1}{4}$, $\mathcal{L}_{p_1}(y) = \frac{1}{2} a(y_1 \oplus y_2)$ and $\mathcal{L}_{p_2}(y) = \frac{1}{2} a(y_2)$, with conditionals $p_1(y_2 \mid y_1) = a(y_1 \oplus y_2)$, $p_2(y_2 \mid y_1) = a(y_2)$ and $p_b(\cdot \mid y_1)$ uniform. At $t = 1$ all ratios are 1, so $Z_1 = 1$; at $t = 2$ the unnormalized score is $2\, a(y_1 \oplus y_2)\, a(y_2)$, whence

$$Z_2(y_1 = 0) = 2(0.81 + 0.01) = 1.64, \qquad Z_2(y_1 = 1) = 2(0.09 + 0.09) = 0.36.$$

The normalizer depends on the prefix, so W is not constant and Proposition 1 already rules out the feature-space factorization. Concretely, $\mathcal{L}_{\hat{p}}$ assigns $(0.4939, 0.0061, 0.2500, 0.2500)$ to $(0,0), (0,1), (1,0), (1,1)$, against the values $(0.81, 0.01, 0.09, 0.09)$ that the factorization would prescribe, and projectivity fails as well:

$$\Pr_{\mathcal{L}_{\hat{p}}}[\Pi_1 = 0] = 0.4939 + 0.25 = 0.7439, \qquad \Pr_{\mathcal{L}_{p_1}}[\Pi_1 = 0] = 0.9.$$

Since any permutation of finitely many points of $\mathbb{R}^d$ with $d \geq 2$ is realized by a compactly supported diffeomorphism, embedding the four sequences as the corners of $\{0, 1\}^2 \subset \mathbb{R}^2$ turns this into a C^1 example.

The causal map works, and Theorem 1 does not cover it. Keep q_b, q_1, q_2 and the masks, and replace $\mathcal{D}$ by the causal map $\mathcal{D}'(y_1, y_2) = (y_1, y_1 \oplus y_2)$. Now $\mathcal{L}_{p_1}(y) = \frac{1}{2} a(y_1)$ and $\mathcal{L}_{p_2}(y) = \frac{1}{2} a(y_1 \oplus y_2)$, so

$$p_1(y_1) = a(y_1), \qquad p_1(y_2 \mid y_1) = \tfrac{1}{2} = p_b(y_2 \mid y_1),$$
$$p_2(y_1) = \tfrac{1}{2} = p_b(y_1), \qquad p_2(y_2 \mid y_1) = a(y_1 \oplus y_2),$$

which is exactly the off-block agreement produced by Step 1 of the proof. Hence $Z_1 = Z_2 = 1$, $\mathcal{L}_{\hat{p}}(y) = a(y_1)a(y_1 \oplus y_2)$, and $\mathcal{D}'\#\mathcal{L}_{\hat{p}}(z) = a(z_1)a(z_2)$, as Theorem 3 asserts.

Theorem 1 does not apply to this family, and its conclusion is false for it. Its hypothesis requires y_2 to be independent of y_1 under p_2, whereas $\mathcal{L}_{p_2}(0,0) = \frac{1}{2}a(0) = 0.45$ while $\mathcal{L}_{p_2}(y_1 = 0)\,\mathcal{L}_{p_2}(y_2 = 0) = \frac{1}{2} \cdot \frac{1}{2} = 0.25$. Its conclusion would give $\Pr_{\mathcal{L}_{\hat{p}}}[y_2 = 0] = \Pr_{\mathcal{L}_{p_2}}[y_2 = 0] = 0.5$, whereas in fact $\Pr_{\mathcal{L}_{\hat{p}}}[y_2 = 0] = a(0)^2 + a(1)^2 = 0.82$. The projections supplied by Theorem 3, namely $\Pi_1(y) = y_1$ and $\Pi_2(y) = y_1 \oplus y_2$, do match:

$$\Pr_{\mathcal{L}_{\hat{p}}}[\Pi_2 = 0] = a(0)a(0) + a(1)a(0) = 0.9 = \Pr_{\mathcal{L}_{p_2}}[\Pi_2 = 0].$$

So the component models may be entangled with the past in token coordinates, provided the entanglement is causally invertible; what causality forbids is only the mixing of positions by $\mathcal{D}$.

Remark 4 (Relation to the diffusion setting)**.** *Bradley et al. (2025) exploit the fact that the operator of Equation 1 is 1-homogeneous and therefore independent of parameterization, i.e.*

$$\mathcal{C}[\mathcal{D}\#p_1, \ldots, \mathcal{D}\#p_k, \mathcal{D}\#p_b] = \mathcal{D}\#\mathcal{C}[p_1, \ldots, p_k, p_b]$$

for any invertible $\mathcal{D}$. That invariance is a statement about a single, global normalizer. Equation 2 instead normalizes at every step, and by Proposition 1 the two agree only when W is constant. The proof above establishes $\mathcal{L}_{\hat{p}} = \mathcal{C}[\mathcal{L}_{p_1}, \ldots, \mathcal{L}_{p_k}, \mathcal{L}_{p_b}]$ as a conclusion rather than assuming it, and the hypothesis that delivers it is causality of $\mathcal{D}$.

D.4 Length Generalization

Length generalization refers to the ability of Transformer models to successfully perform a given task on input sequences that are longer than those encountered during training. Zhou et al. (2024) conjectured that a Transformer can achieve length generalization for a particular algorithmic task if there exists a sufficiently simple implementation of that algorithm within the RASP (Restricted Access Sequence Processing) framework. Our results are stated for arbitrary ℓ. Consequently, if the hypotheses of Theorems 1, 2, or 3 hold uniformly at a target length and each component model performs its corresponding task at that length, then the composed model $\hat{p}$ inherits the corresponding behavior on the composed task. This is a conditional transfer statement: the theory does not by itself prove that arbitrary pretrained or fine-tuned Transformers length-generalize, nor that approximate factorization must persist outside the lengths on which the components were trained. Under a defect that is uniform in the length, Theorem 4 makes the transfer quantitative and shows what it costs: the divergence per generated token stays bounded by $E^2/8$, so the guarantee survives at the level of individual steps, while the sequence-level guarantee weakens as $\sqrt{\ell}$ and is informative at a target length only if the defect is small compared with $1/\sqrt{\ell}$.

E Approximate Factorization

Definitions 1 and 4 are exact requirements: off its designated region, an expert must coincide with the background model. Fine-tuned checkpoints do not, as Appendix G.3 measures. This appendix develops the approximate theory announced in Subsection 2.4 and shows that none of the three theorems depends on exactness.

The development turns on a single quantity, the defect of one family of components from another (Definition 6), and a single estimate, Theorem 4, which bounds the divergence between the two compositions by that defect. Subsection E.1 proves the theorem, sharpens it, and shows that its rate is attained. Subsection E.2 obtains approximate forms of Theorems 1, 2 and 3 by choosing the comparison family to be one that satisfies the corresponding hypothesis. Subsection E.3 gives a canonical choice of that family, for which the defect is a function of the off-region deviations alone and can be estimated from forward passes, and instantiates the bound at the values measured in Appendix G.3.

Throughout, $(p_1, \ldots, p_k, p_b)$ and $(q_1, \ldots, q_k, q_b)$ are two families of autoregressive models on $\mathcal{X}^\ell$ with common support, conditioned on a prompt x; $\hat{p}$ and $\hat{q}$ denote their selective compositions in the sense of Equation 2, and $\mathcal{L}_{\hat{p}}, \mathcal{L}_{\hat{q}}$ the induced sequence laws. We write KL for the Kullback–Leibler

divergence and TV for the total variation distance, and abbreviate

$$\hat{p}_t \coloneqq \hat{p}(\cdot \mid x, y_{<t}), \qquad \hat{q}_t \coloneqq \hat{q}(\cdot \mid x, y_{<t}), \qquad f_t \coloneqq f_t(\cdot \mid x, y_{<t}),$$

with f_t the log-ratio field of Definition 6.

E.1 Stability of Selective Composition

Multiplying out Equation 2, the unnormalized score of the composition is $\prod_{i=1}^{k} p_i / p_b^{k-1}$. The two families therefore enter the comparison only through the log-ratio of their scores, which is the field f_t of Definition 6, and the passage from $\hat{p}$ to $\hat{q}$ is an exponential tilt by it.

Lemma 5 (Composition as tilting)**.** *For every t, every prefix $y_{<t}$ and every $v \in \mathcal{X}$,*

$$\hat{q}_t(v) = \frac{\hat{p}_t(v)\, e^{f_t(v)}}{\sum_{v' \in \mathcal{X}} \hat{p}_t(v')\, e^{f_t(v')}}.$$

Proof. By Equation 2, $\hat{p}_t(v) = \big(\prod_{i=1}^{k} p_i(v \mid x, y_{<t})\big)\, p_b(v \mid x, y_{<t})^{-(k-1)} / Z_t(x, y_{<t})$, and likewise for $\hat{q}_t$ with the second family. Substituting $q_i = p_i e^{\Delta_i}$ and $q_b = p_b e^{\Delta_b}$ gives

$$\Big(\prod_{i=1}^{k} q_i(v \mid x, y_{<t})\Big)\, q_b(v \mid x, y_{<t})^{-(k-1)} = \Big(\prod_{i=1}^{k} p_i(v \mid x, y_{<t})\Big)\, p_b(v \mid x, y_{<t})^{-(k-1)} e^{f_t(v)},$$

so the two unnormalized scores differ by the factor $e^{f_t(v)}$. Both sides are then divided by their own sums over $\mathcal{X}$, which is the displayed identity. □

Proof of Theorem 4. Fix x, a step t and a prefix $y_{<t}$, and write E_t for the defect there. For $s \in [0,1]$ let

$$\hat{p}_t^{(s)}(v) \coloneqq \frac{\hat{p}_t(v) e^{s f_t(v)}}{\sum_{v'} \hat{p}_t(v') e^{s f_t(v')}}, \qquad \psi(s) \coloneqq \log \sum_{v \in \mathcal{X}} \hat{p}_t(v)\, e^{s f_t(v)},$$

so that ψ is smooth on $[0,1]$ with $\psi(0) = 0$, $\psi'(s) = \mathbb{E}_{\hat{p}_t^{(s)}}[f_t]$ and $\psi''(s) = \mathrm{Var}_{\hat{p}_t^{(s)}}(f_t)$. By Lemma 5, $\hat{q}_t = \hat{p}_t^{(1)}$, hence

$$\mathrm{KL}(\hat{p}_t \,\|\, \hat{q}_t) = \mathbb{E}_{\hat{p}_t}\big[\log \hat{p}_t - \log \hat{q}_t\big] = \psi(1) - \mathbb{E}_{\hat{p}_t}[f_t] = \psi(1) - \psi(0) - \psi'(0). \tag{8}$$

Taylor's theorem with integral remainder turns the right-hand side into $\int_0^1 (1-s)\, \psi''(s)\, ds$. Each $\psi''(s)$ is the variance of a random variable taking values in an interval of length E_t, so $\psi''(s) \le E_t^2/4$ by Popoviciu's inequality, and $\int_0^1 (1-s)\, ds = \frac{1}{2}$; therefore $\mathrm{KL}(\hat{p}_t \| \hat{q}_t) \le E_t^2/8$. Exchanging the roles of the two families replaces f_t by $-f_t$, which has the same oscillation, so $\mathrm{KL}(\hat{q}_t \| \hat{p}_t) \le E_t^2/8$ as well.

For the sequence-level statement, the chain rule for the Kullback–Leibler divergence between the laws of $y_{1:\ell}$ gives

$$\mathrm{KL}\big(\mathcal{L}_{\hat{p}}(\cdot \mid x) \,\big\|\, \mathcal{L}_{\hat{q}}(\cdot \mid x)\big) = \sum_{t=1}^{\ell} \mathbb{E}_{y_{<t} \sim \mathcal{L}_{\hat{p}}}\Big[\mathrm{KL}\big(\hat{p}_t \,\|\, \hat{q}_t\big)\Big] \le \frac{1}{8} \sum_{t=1}^{\ell} \mathbb{E}_{y_{<t} \sim \mathcal{L}_{\hat{p}}}\big[E_t(x, y_{<t})^2\big] \le \frac{\ell E^2}{8},$$

and Pinsker's inequality gives $\mathrm{TV} \le \sqrt{\mathrm{KL}/2} \le \sqrt{\ell}\, E/4$. □

Two features of this argument are worth isolating, because they are what make the bound usable at the deviations real checkpoints exhibit. The first is that the sequence-level bound is an average over prefixes before it is a supremum: what the chain rule delivers is $\sum_t \mathbb{E}_{\mathcal{L}_{\hat{p}}}[E_t^2]$, along the composition's own trajectory, and ℓE^2 is only the crudest way of bounding it. The second is that Equation 8 is an identity, so the passage to $E_t^2/8$ can be sharpened whenever more is known about f_t than its range.

Proposition 2 (Variance form of the per-step bound)**.** *With the notation of the proof above,*

$$\mathrm{KL}(\hat{p}_t \,\|\, \hat{q}_t) = \int_0^1 (1-s)\, \mathrm{Var}_{\hat{p}_t^{(s)}}(f_t)\, ds \;\le\; \frac{1}{2} \max_{s \in [0,1]} \mathrm{Var}_{\hat{p}_t^{(s)}}(f_t).$$

If moreover $f_t - \mathbb{E}_{\hat{p}_t}[f_t] \le M$ *pointwise on* $\mathcal{X}$*, then*

$$\mathrm{KL}(\hat{p}_t \,\|\, \hat{q}_t) \;\le\; \frac{e^M - M - 1}{M^2}\; \mathrm{Var}_{\hat{p}_t}(f_t),$$

where the prefactor decreases to $1/2$ *as* $M \to 0$*.*

Proof. The identity is the integral remainder computed in the proof of Theorem 4, and the inequality follows from $\int_0^1 (1-s)ds = \frac{1}{2}$. The second display is Bennett's bound on the logarithmic moment generating function of a random variable bounded above (Boucheron et al., 2013, Chapter 2), applied to $f_t - \mathbb{E}_{\hat{p}_t}[f_t]$ under $\hat{p}_t$ at $s = 1$, combined with Equation 8. □

Remark 5 (Which tokens pay)**.** *Proposition 2 weights the deviation at each token by the probability the composition itself assigns to that token, whereas the defect of Definition 6 is a spread over the whole vocabulary. The per-token deviations we measure in Appendix G.3 are heavy-tailed: the off-domain rows of Table 3 have means of* 0.109 *and* 0.073 *nats against maxima of* 5.64 *and* 4.39*. The uniform form of Theorem 4 is set by that tail, and is correspondingly pessimistic; the variance form is not, provided the tail sits on tokens the composition rarely emits.*

The quadratic dependence on the defect, the constant $1/8$ and the linear growth in ℓ are all attained, and already between two families each of which satisfies Definition 1 exactly. So the length dependence of Theorem 4 is a property of composition under perturbation, not an artifact of the proof.

Proposition 3 (The rate is attained)**.** *Let* $\mathcal{X} = \{0, 1\}$*,* $k = 1$*,* $M_1 = [\ell]$ *and* $M_b = \varnothing$*. Fix* $a > 0$*, let* $p_b = q_b$ *be uniform on* $\mathcal{X}$ *at every step and every prefix, let* $p_1 = p_b$*, and let* $q_1(1 \mid x, y_{<t}) = e^a/(1+e^a)$ *at every step and every prefix. Both families satisfy Definition 1 with respect to* $\{M_1, M_b\}$*, the defect equals* a *at every step, and*

$$\mathrm{KL}\big(\mathcal{L}_{\hat{p}} \,\|\, \mathcal{L}_{\hat{q}}\big) = \ell \log\cosh(a/2) = \frac{\ell a^2}{8}\Big(1 - \frac{a^2}{24} + O(a^4)\Big).$$

Proof. Definition 1 holds for both families: its conditions 1 and 3 are statements about the empty index set M_1^c, and its condition 2 is a statement about the single block M_1. With $k = 1$ the composition of Equation 2 is $p_b \cdot (p_1/p_b) = p_1$, so $\hat{p} = p_1$ and $\hat{q} = q_1$. Also with $k = 1$ we have $f_t = \Delta_1$, whose oscillation is $\log\big(q_1(1)/q_1(0)\big) = a$ since p_1 is uniform. Both sequence laws are products of ℓ identical factors, so

$$\begin{aligned}\mathrm{KL}\big(\mathcal{L}_{\hat{p}} \,\|\, \mathcal{L}_{\hat{q}}\big) &= \ell\, \mathrm{KL}\big(\mathrm{Bernoulli}(\tfrac{1}{2}) \,\big\|\, \mathrm{Bernoulli}(\sigma(a))\big) \\ &= -\frac{\ell}{2}\log\big(4\sigma(a)(1-\sigma(a))\big) = \ell \log\cosh(a/2),\end{aligned}$$

where $\sigma(a) = e^a/(1+e^a)$ and the last equality uses $4\sigma(a)(1-\sigma(a)) = \mathrm{sech}^2(a/2)$. The expansion follows from $\log\cosh u = u^2/2 - u^4/12 + O(u^6)$. □

Remark 6 (Dependence on the number of components)**.** *The field* f_t *sums a contribution from each component, so a family of* k *components whose individual deviations do not cancel has a defect growing linearly in* k*, and a composition error growing quadratically. Theorem 4 therefore predicts that composition degrades as components are added even when each is individually well localized, which is what we observe: beyond two experts the generation quality collapses (Appendix G.2.3), and the mechanism there, disagreement about end-of-sequence behavior, is a large off-region deviation concentrated at the positions where it is most costly.*

E.2 Approximate Forms of the Three Theorems

Theorem 4 compares two arbitrary families. It becomes a statement about a single family once the comparison family is required to satisfy one of the hypotheses of Section 2, in which case the conclusion of the corresponding theorem holds for it exactly and transfers with an error controlled by the defect.

Definition 7 (ϵ-approximate factorization)**.** *Let* $P = (p_1, \ldots, p_k, p_b)$ *be a family of autoregressive models on* $\mathcal{X}^\ell$ *with common support and let* $\epsilon \ge 0$*. We say that* P *is* ϵ-approximately factorized *with respect to a partition* $[\ell] = M_1 \uplus \cdots \uplus M_k \uplus M_b$ *if there exists a family* $Q = (q_1, \ldots, q_k, q_b)$

satisfying Definition 1 with respect to that partition whose defect from P is at most ϵ. Replacing Definition 1 by Definition 4, we say P is jointly ϵ-approximately factorizable *on a task T; replacing it by the hypothesis of Theorem 3, that P is* ϵ-approximately factorized in the feature space of $\mathcal{D}$. *In each case Q is called a* witness, *and $\epsilon = 0$ returns the exact property.*

Definition 8 (δ-projective composition)**.** *Given distributions $\{p_i\}$ with projections $\{\Pi_i : \mathcal{X}^\ell \to \mathbb{R}^m\}$ and $\delta \geq 0$, we call $\hat{p}$ a* δ-projective composition *if*

$$\forall i: \qquad \mathrm{TV}\big(\Pi_i \# \hat{p},\ \Pi_i \# p_i\big) \leq \delta.$$

Definition 2 is the case $\delta = 0$.

Corollary 1 (Theorem 1, approximate form)**.** *Suppose P is ϵ-approximately factorized with respect to $\{M_i\}_{i=1}^k$ and M_b, with witness Q. Then*

$$\mathrm{TV}\Big(\mathcal{L}_{\hat{p}}(\cdot \mid x),\ q_b(\cdot_{M_b} \mid x) \prod_{i=1}^{k} q_i(\cdot_{M_i} \mid x)\Big) \ \leq\ \frac{\sqrt{\ell}\,\epsilon}{4},$$

and $\mathcal{L}_{\hat{p}}$ is a $\frac{\sqrt{\ell}\,\epsilon}{4}$-projective composition of $\{\mathcal{L}_{q_i}\}$ with respect to $\Pi_i(y) = y_{M_i}$. If in addition $\mathrm{osc}_{v \in \mathcal{X}}\, \Delta_i(v) \leq e_i$ at every step and prefix, then the same holds against the components themselves:

$$\mathrm{TV}\big(\Pi_i \# \mathcal{L}_{\hat{p}},\ \Pi_i \# \mathcal{L}_{p_i}\big) \ \leq\ \frac{\sqrt{\ell}}{4}\,(\epsilon + e_i) \qquad \textit{for every } i.$$

Proof. Theorem 1 applies to Q, so $\mathcal{L}_{\hat{q}} = q_b(\cdot_{M_b} \mid x) \prod_i q_i(\cdot_{M_i} \mid x)$ and $\Pi_i \# \mathcal{L}_{\hat{q}} = \Pi_i \# \mathcal{L}_{q_i}$ for every i. Theorem 4 bounds $\mathrm{TV}(\mathcal{L}_{\hat{p}}, \mathcal{L}_{\hat{q}})$ by $\sqrt{\ell}\,\epsilon/4$, and pushforwards contract total variation, which gives the first two claims. For the last one, apply Theorem 4 to the pair of one-component families $(p_i; p_b)$ and $(q_i; p_b)$: for $k = 1$ the composition is the component itself and the field f_t is Δ_i, so $\mathrm{TV}(\mathcal{L}_{p_i}, \mathcal{L}_{q_i}) \leq \sqrt{\ell}\, e_i/4$. The triangle inequality applied to $\Pi_i \# \mathcal{L}_{\hat{p}}$, $\Pi_i \# \mathcal{L}_{\hat{q}} = \Pi_i \# \mathcal{L}_{q_i}$ and $\Pi_i \# \mathcal{L}_{p_i}$ concludes. □

Corollary 2 (Theorem 2, approximate form)**.** *Suppose P is jointly ϵ-approximately factorizable on T, with witness Q and active index sets $I_i = \{\tau_1^i + 1, \ldots, \tau_2^i\}$. Then for every prompt $x \in T$,*

$$\mathrm{TV}\left(\mathcal{L}_{\hat{p}}(\cdot \mid x),\ \left(\prod_{t \notin \bigcup_i I_i} q_b(\cdot \mid x, y_{<t})\right)\left(\prod_{i=1}^{k} \prod_{t \in I_i} q_i(\cdot \mid x, y_{<t})\right)\right) \ \leq\ \frac{\sqrt{\ell}\,\epsilon}{4}.$$

Proof. Theorem 2 identifies $\mathcal{L}_{\hat{q}}$ with the displayed product; Theorem 4 bounds the distance to it. □

Corollary 3 (Theorem 3, approximate form)**.** *Suppose P is ϵ-approximately factorized in the feature space of a causal reparameterization $\mathcal{D}$, with witness Q and partition $\{M_i\}_{i=1}^k, M_b$. Then*

$$\mathrm{TV}\Big(\mathcal{D}\#\mathcal{L}_{\hat{p}},\ (\mathcal{D}\#\mathcal{L}_{q_b})_{M_b} \prod_{i=1}^{k} (\mathcal{D}\#\mathcal{L}_{q_i})_{M_i}\Big) \ \leq\ \frac{\sqrt{\ell}\,\epsilon}{4},$$

and $\mathcal{L}_{\hat{p}}$ is a $\frac{\sqrt{\ell}\,\epsilon}{4}$-projective composition of $\{\mathcal{L}_{q_i}\}$ with respect to $\Pi_i(y) = \mathcal{D}(y)_{M_i}$.

Proof. Theorem 3 applies to Q and identifies $\mathcal{D}\#\mathcal{L}_{\hat{q}}$ with the displayed block product. Total variation is invariant under the bijection $\mathcal{D}$ (Lemma 2), so $\mathrm{TV}(\mathcal{D}\#\mathcal{L}_{\hat{p}}, \mathcal{D}\#\mathcal{L}_{\hat{q}}) = \mathrm{TV}(\mathcal{L}_{\hat{p}}, \mathcal{L}_{\hat{q}}) \leq \sqrt{\ell}\,\epsilon/4$, and pushforwards contract it. □

The hypothesis of Corollary 3 refers to a feature space, but the quantity it bounds does not: the defect is the same whether it is computed before or after a causal change of coordinates. This is the approximate counterpart of the invariance discussed in Remark 4, and it is what makes the hypothesis testable, since the deviations can be read off in token space where the models are evaluated.

Proposition 4 (The defect is a causal invariant)**.** *Let $\mathcal{D}$ be a causal reparameterization in the sense of Definition 5 and let P, Q be two families with common support. Then the defect of $(\mathcal{D}\#\mathcal{L}_{q_1}, \ldots, \mathcal{D}\#\mathcal{L}_{q_b})$ from $(\mathcal{D}\#\mathcal{L}_{p_1}, \ldots, \mathcal{D}\#\mathcal{L}_{p_b})$ at a step t and feature prefix $z_{<t} = \mathcal{D}(y)_{<t}$ equals the defect of Q from P at t and $y_{<t}$.*

Proof. Fix t, a prefix $y_{<t}$ with feature prefix $z_{<t}$, and let $w = \mathcal{D}_t(y_{<t}, v)$. By Lemma 3 each model's conditional is carried over unchanged, so the log-ratios defining the field are unchanged component by component: $\log(\mathcal{D}\#\mathcal{L}_{q_i})(w \mid z_{<t}) - \log(\mathcal{D}\#\mathcal{L}_{p_i})(w \mid z_{<t}) = \Delta_i(v)$. In the continuous case of Remark 1 each conditional acquires the factor $|\det \partial_v \mathcal{D}_t(y_{<t}, v)|^{-1}$, which depends on $\mathcal{D}$ alone and cancels in the difference. Hence the feature-space field at w equals $f_t(v)$, and $v \mapsto w$ is a bijection from $\mathcal{X}$ onto $\mathcal{Z}_t$ by Definition 5, so the two oscillations are taken over the same set of values. □

E.3 Certifying the Defect from Forward Passes

Definition 7 asserts the existence of a witness, which makes the approximate statements clean but leaves open how a practitioner would exhibit one. There is a canonical choice. Given any candidate assignment of regions to components, project each component onto that assignment by replacing it with the background model wherever it is not meant to act. What is left over is exactly the component's deviation off its own region, and this is the quantity Equation 6 measures.

Proposition 5 (Projected family)**.** *Let $P = (p_1, \dots, p_k, p_b)$ have common support and let R assign to every step t and prefix $y_{<t}$ a unique active index $m(t, y_{<t}) \in \{b, 1, \dots, k\}$. Define the* projected family *Q^R by $q_b := p_b$ and, for $i \in \{1, \dots, k\}$,*

$$q_i(\cdot \mid x, y_{<t}) := \begin{cases} p_i(\cdot \mid x, y_{<t}), & m(t, y_{<t}) = i, \\ p_b(\cdot \mid x, y_{<t}), & \textit{otherwise}, \end{cases}$$

and write $\lambda_i(v \mid x, y_{<t}) := \log p_i(v \mid x, y_{<t}) - \log p_b(v \mid x, y_{<t})$. Then:

1. *every per-step normalizer of $\hat{q}$ equals 1 and $\mathcal{L}_{\hat{q}}(y_{1:\ell} \mid x) = \prod_{t=1}^{\ell} p_{m(t,y_{<t})}(y_t \mid x, y_{<t})$;*
2. *the defect of Q^R from P at $(t, y_{<t})$ is*
$$E_t(x, y_{<t}) = \operatorname*{osc}_{v \in \mathcal{X}} \sum_{i \neq m(t,y_{<t})} \lambda_i(v \mid x, y_{<t}) \;\le\; \sum_{i \neq m(t,y_{<t})} \operatorname*{osc}_{v \in \mathcal{X}} \lambda_i(v \mid x, y_{<t}),$$
the sums running over the inactive components only;
3. *if R is given by pairwise disjoint stopping-time intervals $I_i = \{\tau_1^i + 1, \dots, \tau_2^i\}$, then Q^R is jointly factorizable on T in the sense of Definition 4, with those same intervals.*

Proof. (1) At $(t, y_{<t})$ every inactive component satisfies $q_i = q_b$, so all but at most one of the ratios in Equation 2 equal 1 and the unnormalized score is $q_{m(t,y_{<t})}(\cdot \mid x, y_{<t})$, which is already a distribution on $\mathcal{X}$. This is the cancellation of Lemma 1, which uses only the uniqueness of the active index and not the structure of R. The chain rule and $q_m = p_m$ on the active index give the displayed law.

(2) At $(t, y_{<t})$ we have $\Delta_b \equiv 0$, and $\Delta_i \equiv 0$ for the active index, while $\Delta_i = \log p_b - \log p_i = -\lambda_i$ for every inactive i. Hence $f_t = -\sum_{i \neq m} \lambda_i$, whose oscillation is that of $\sum_{i \neq m} \lambda_i$; subadditivity of the oscillation gives the inequality.

(3) Fix i and a prompt $x \in T$. By construction q_i agrees with p_b at every step outside I_i, so

$$\mathcal{L}_{q_i}(y_{1:\ell} \mid x) = \prod_{t=1}^{\ell} q_i(y_t \mid x, y_{<t}) = \left(\prod_{t \le \tau_1^i} p_b \right) \left(\prod_{t=\tau_1^i+1}^{\tau_2^i} q_i \right) \left(\prod_{t > \tau_2^i} p_b \right),$$

which is the definition of skill factorizability with respect to p_b with the stopping times (τ_1^i, τ_2^i). The intervals are pairwise disjoint by assumption, so Definition 4 holds. □

Corollary 4 (A certificate for the defect)**.** *In the setting of Proposition 5,*

$$\mathrm{KL}\left(\mathcal{L}_{\hat{p}}(\cdot \mid x) \,\middle\|\, \prod_{t=1}^{\ell} p_{m(t,y_{<t})}(y_t \mid x, y_{<t}) \right) \;\le\; \frac{1}{8} \sum_{t=1}^{\ell} \mathbb{E}_{y_{<t} \sim \mathcal{L}_{\hat{p}}} \left[\left(\sum_{i \neq m(t,y_{<t})} \operatorname*{osc}_{v \in \mathcal{X}} \lambda_i \right)^2 \right].$$

If R is given by pairwise disjoint stopping-time intervals, then P is jointly ϵ-approximately factorizable on T with

$$\epsilon = \sup_{t \le \ell} \sup_{y_{<t}} \sum_{i \neq m(t,y_{<t})} \operatorname*{osc}_{v \in \mathcal{X}} \lambda_i(v \mid x, y_{<t}),$$

and Corollary 2 applies with that ϵ and the witness Q^R.

Proof. Combine Proposition 5 with Theorem 4, in its per-prefix form for the first display and in its uniform form for the second. □

Corollary 4 is what ties the theory to the measurement, in three respects. First, only the deviations of the *inactive* components enter: how far an expert moves from the background model on its own region, which is what fine-tuning is for, is irrelevant to the bound. Second, in the stopping-time setting of Theorem 2 nothing is left to assume. Any assignment of disjoint prefix-measurable intervals to the components makes the projected family jointly factorizable, so every family of components is jointly ϵ-approximately factorizable with ϵ equal to its own off-interval deviation, and one may take the infimum over assignments. For Theorems 1 and 3 the projected family delivers the off-region agreement but not the independence conditions 1–2 of Definition 1, which remain hypotheses on the components. Third, the deviations λ_i are differences of log-probabilities of models evaluated on the same prefix, so they are obtained from one forward pass of each component, with no labels and no decoding, which is exactly the diagnostic of Appendix G.3.

Instantiation at the measured values. Table 3 reports $\epsilon_i(t) = |\lambda_i(y_t \mid x, y_{<t})|$ for the off-domain expert, that is, the magnitude of λ_i at the realized token rather than its spread over the vocabulary. The two are related but not equal. Since $\sum_v p_b(v \mid x, y_{<t}) e^{\lambda_i(v|x,y_{<t})} = 1$, the function λ_i takes both signs unless it vanishes identically, so $\operatorname{osc}_v \lambda_i \geq |\lambda_i(v)|$ for every v: the measured quantity is a lower bound on the oscillation that enters Corollary 4, and instantiating the bound at it is optimistic. With that caveat, the resulting magnitudes are the following. There are $k = 2$ components, and under the assignment that gives each step to the on-domain expert the inactive sum has a single term, the off-domain expert's deviation, of mean 0.109 nats on mathematics prompts and 0.073 on coding prompts; at a step assigned to the background model both terms enter, for 0.182. Taking the last as E gives a per-token bound of $E^2/8 \approx 4.1 \times 10^{-3}$ nats, about 1.1 nats over a 256-token generation. The same computation at the p90 and p99 columns gives 3.6×10^{-2} and 0.49 nats per token, hence 9.3 and 125 nats over the same generation.

Both halves of that calculation are worth stating plainly. At the typical per-token deviation the guarantee is meaningful, and it is the per-token form that a practitioner deciding whether to compose two checkpoints should read. At the tail it is not: a uniform defect taken at a high quantile and multiplied by a realistic generation length gives a sequence-level bound with no content, and we do not claim otherwise. The forms that survive at length are the per-prefix average in Theorem 4 and the variance form of Proposition 2, both of which weight the deviation by how often and on which tokens it is actually incurred, rather than by its supremum; estimating either requires only the same forward passes and a second moment in place of a quantile.

F Model Merging

Model merging refers to the process of combining models trained on distinct tasks into a single model that achieves strong performance across all tasks, ideally matching or exceeding that of the individual task-specific experts. Various approaches to model merging have been proposed (see Appendix B); however, the underlying reasons for their empirical success remain poorly understood. In this section, we present insights suggesting that our results may help explain why such methods are effective in practice, and how they relate to prior attempts at theoretical justification (Ortiz-Jimenez et al., 2023). We begin with a simplified setting to illustrate the core intuition in Subsection F.1, and subsequently extend the discussion to a more general framework in Subsection F.2. In particular, we consider *task arithmetic*, a model merging technique defined as follows. Let $f_b : \mathbb{R}^n \to \mathbb{R}^m$ denote a base neural network trained on an initial task. This base model is then fine-tuned on multiple tasks, yielding a collection of expert models $f_1, f_2, \ldots, f_k$. Let θ_b denote the parameter vector of f_b, and let $\theta_1, \theta_2, \ldots, \theta_k$ denote the parameter vectors corresponding to $f_1, f_2, \ldots, f_k$, respectively. Task arithmetic constructs a merged model f_{merged} with parameters given by

$$\theta_{\text{merged}} := \theta_b + \sum_{i=1}^{k} (\theta_i - \theta_b). \tag{9}$$

Note that the formula for merging weights in task arithmetic is the same as the formula for merging scores (or log probabilities of tokens) in our composition (Equation 2).

F.1 Task Arithmetic: Linear Case

We present the idea under the simplifying assumption that the models $f_b, f_1, \ldots, f_k$ are linear maps from $\mathbb{R}^n$ to $\mathbb{R}^m$. In particular, for each i, $f_i(x) \in \mathbb{R}^m$. We assume that these models satisfy *factorized conditionals*, which in this case we call Disjoint coordinate factorization, meaning that each model f_i differs from the base model f_b only on a disjoint subset of output coordinates.

Definition 9 (Disjoint coordinate factorization)**.** *Let $f_b, f_1, \ldots, f_k : \mathbb{R}^n \to \mathbb{R}^m$ be linear maps. We say they satisfy disjoint coordinate factorization if there exist masks $M_1, \ldots, M_k \in \{0,1\}^m$ such that:*

1. *(Disjointness of the masks) For all $i \neq j$ and all $r \in \{1, \ldots, m\}$,*

$$M_i^{(r)} = 1 \;\Rightarrow\; M_j^{(r)} = 0.$$

2. *(Localized deviation) For every $i \in \{1, \ldots, k\}$ and all $x \in \mathbb{R}^n$,*

$$(f_i(x))^{(r)} = (f_b(x))^{(r)} \quad \text{whenever } M_i^{(r)} = 0.$$

Equivalently, each difference $f_i - f_b$ is supported only on the coordinates selected by M_i.

If the models $f_b, f_1, \ldots, f_k$ satisfy Definition 9, then the merged model f_{merged} defined in Equation 9 produces outputs that coincide with each expert model on its corresponding specialized coordinates.

This leads to the following key observation: **combining weights according to task arithmetic is equivalent to combining outputs according to our composition**, at least in the linear setting.

F.2 Task Arithmetic: General Case

Ortiz-Jimenez et al. (2023) show that task arithmetic can succeed when each task vector affects only its own region of the input space. More precisely, let $\mathcal{T} = \{\boldsymbol{\tau}_t\}_{t\in[T]}$ be a collection of task vectors, and let $\mathcal{D} = \{\mathcal{D}_t \subset \mathcal{X}\}_{t\in[T]}$ be a collection of pairwise disjoint task supports, that is $\mathcal{D}_t \cap \mathcal{D}_{t'} = \varnothing$ for $t \neq t'$. We say that a network f satisfies the task arithmetic property around $\boldsymbol{\theta}_0$ with respect to $\mathcal{T}$ and $\mathcal{D}$ if, for every choice of coefficients $(\alpha_1, \ldots, \alpha_T) \in \mathcal{A} \subseteq \mathbb{R}^T$,

$$f\left(\boldsymbol{x}; \boldsymbol{\theta}_0 + \sum_{t=1}^{T} \alpha_t \, \boldsymbol{\tau}_t\right) = \begin{cases} f(\boldsymbol{x}; \boldsymbol{\theta}_0 + \alpha_t \, \boldsymbol{\tau}_t), & \boldsymbol{x} \in \mathcal{D}_t, \\ f(\boldsymbol{x}; \boldsymbol{\theta}_0), & \boldsymbol{x} \notin \bigcup_{t=1}^{T} \mathcal{D}_t. \end{cases}$$

In other words, when the input belongs to the support of task t, the merged model behaves exactly as if only the corresponding task vector $\boldsymbol{\tau}_t$ had been added; on inputs outside all task supports, it reduces to the base model. This is the nonlinear analogue of the disjoint-coordinate condition discussed in Subsection F.1: in both cases, the different task updates do not interfere with one another. Thus, successful task arithmetic can be interpreted as a parameter-space realization of the same additive composition principle that our method applies directly to conditional output distributions: each component contributes only on its own region of action, while the base model governs the remaining behavior.

G Experiments

We evaluate the framework in three parts. We first revisit the controlled setting of Subsection 2.1, in which the factorization assumptions hold by construction and the composition rule can therefore be checked exactly. We then compose two domain-specialized large language models and compare the result against the individual experts, against three standard weight-space merging methods, and against the plain product of experts obtained by dropping the base-model division from Equation 2. Finally, since Definitions 1 and 4 are idealizations that real checkpoints cannot be expected to satisfy exactly, we measure how far they are from satisfying them, at the level of decoded generations and at the level of the conditional distributions the definitions actually constrain.

G.1 Letter Replacement

We train three small decoder-only Transformers from scratch on the letter-replacement task of Subsection 2.1: a background model p_b implementing the identity map, an expert p_1 replacing every `A` with `K`, and an expert p_2 replacing every `M` with `B`. Here the designated regions are disjoint by construction, so Definition 1 holds, and Theorem 1 predicts that the composition of Equation 2 applies both substitutions and leaves every other character untouched. This is what we observe: $\hat{p}$ performs both replacements simultaneously with a 100% success rate on random unseen strings.

We begin with the toy example from Subsection 2.1. We train a small decoder-only Transformer (GPT-2 style) from scratch using `GPT2LMHeadModel` (Radford et al., 2019; Wolf et al., 2020) on a synthetic letter-replacement task. Each token represents a single character.

The vocabulary consists of the 26 uppercase English letters plus a space token. Each synthetic input has length 16. For each of the three tasks below, the submitted notebook constructs 2,000 training examples by sampling random strings and applying the task-specific deterministic transformation. The model uses `GPT2Config` with `n_positions=32`, `n_embd=64`, `n_layer=2`, and `n_head=2`. Models are trained for 5 epochs with AdamW, learning rate $3 \cdot 10^{-4}$, and token-level cross-entropy loss. Decoding is greedy, taking the argmax token at each position.

We construct three models:

- p_b: the base model, which implements the identity mapping (i.e., it returns the input string unchanged),
- p_1: a model that replaces every occurrence of `A` with `K`,
- p_2: a model that replaces every occurrence of `M` with `B`.

We then compose these models using the rule in Equation 2. In the implementation, this corresponds to adding the log-probabilities of the two experts and subtracting the log-probabilities of the identity model. The resulting model $\hat{p}$ successfully combines both transformations: for any input string, it replaces every `A` with `K` and every `M` with `B`. In this toy setting, the method achieves a 100% success rate over random unseen strings, confirming that the proposed composition operator captures the intended behavior.

G.2 Experiments with LLMs

We compose the base model p_{base} given by `Gemma 2 2B` (DeepMind, 2024) with two task-specific fine-tuned versions from `MergeBench` (He et al., 2026), one specialized for mathematics (p_{math}) and one for coding (p_{coding}), via Equation 2. We evaluate the resulting model $\hat{p}$ on two coding benchmarks (`HumanEval+` (Chen et al., 2021) and `MBPP+` (Austin et al., 2021)) and two mathematics benchmarks (`GSM8K` (Cobbe et al., 2021) and `MATH` (Hendrycks et al., 2021)).

Baselines. Besides the three component checkpoints, we compare against three weight-space merging methods applied to the same two experts: *task arithmetic* (Ilharco et al., 2023) with $\alpha = 1.0$, which Appendix F identifies as the parameter-space analogue of Equation 2; *TIES-merging* (Yadav et al., 2023) with a top-20% magnitude trim and $\lambda = 1.0$; and a *model soup* (Wortsman et al., 2022) given by the uniform average of the two fine-tuned endpoints. We also run the ablation that isolates the role of the base-model division: the plain product of experts $p_{\text{math}} \cdot p_{\text{coding}}$ in the sense of Hinton (2002), renormalized, with everything else held fixed. Under Definition 1 each expert contributes only its likelihood ratio p_i / p_{base}, which equals 1 outside its designated region so that the expert drops out; without the division the shared background mass is counted k times over, and the composition is pulled towards tokens that are merely plausible under the background model.

All methods merge the same three checkpoints and are evaluated under an identical protocol: greedy decoding, the same prompts and the same few-shot prefixes, described in Appendix G.2.2. Results are reported in Table 2. Since decoding is deterministic, the reported uncertainty is sampling over the evaluation set, given as a binomial standard error with $n = 1319$, 5000, 164 and 378 problems respectively.

Selective composition combines complementary experts: it improves over the coding expert on both coding benchmarks while retaining most of the math expert's performance on the mathematics

Table 2: Accuracy (GSM8K, MATH) and pass@1 (HumanEval+, MBPP+) in %, with binomial standard errors. *Product* is Equation 2 without the base-model division. The last row is the mean across the four benchmarks.

Benchmark	p_{base}	p_{coding}	p_{math}	Product	Task arith.	TIES	Soup	$\hat{p}$ (Eq. 2)
GSM8K	$3.2_{\pm 0.5}$	$3.2_{\pm 0.5}$	$50.2_{\pm 1.4}$	$40.1_{\pm 1.4}$	$54.1_{\pm 1.4}$	$\mathbf{54.9}_{\pm 1.4}$	$41.1_{\pm 1.4}$	$41.2_{\pm 1.4}$
MATH	$16.3_{\pm 0.5}$	$16.7_{\pm 0.5}$	$\mathbf{25.5}_{\pm 0.6}$	$24.3_{\pm 0.6}$	$16.5_{\pm 0.5}$	$23.1_{\pm 0.6}$	$24.2_{\pm 0.6}$	$24.2_{\pm 0.6}$
HumanEval+	$3.7_{\pm 1.5}$	$24.4_{\pm 3.4}$	$13.4_{\pm 2.7}$	$20.1_{\pm 3.1}$	$\mathbf{31.1}_{\pm 3.6}$	$25.6_{\pm 3.4}$	$22.6_{\pm 3.3}$	$30.5_{\pm 3.6}$
MBPP+	$33.9_{\pm 2.4}$	$38.1_{\pm 2.5}$	$32.8_{\pm 2.4}$	$37.0_{\pm 2.5}$	$32.8_{\pm 2.4}$	$36.0_{\pm 2.5}$	$38.1_{\pm 2.5}$	$\mathbf{39.4}_{\pm 2.5}$
Mean	14.3	20.6	30.5	30.4	33.6	**34.9**	31.5	33.8

benchmarks. Two qualifications should be read alongside this. First, the coding-side gains over the coding expert are individually within the standard errors; what the table supports unambiguously is the cross-skill retention, that is, a single model at 30.5 on HumanEval+ and 24.2 on MATH where each expert is strong on one side and weak on the other. Second, on GSM8K both task arithmetic and TIES are ahead of $\hat{p}$ by roughly 13 points, and the across-benchmark means (33.8 for $\hat{p}$, 33.6, 34.9 and 31.5 for the three merging baselines) place selective composition on a par with weight-space merging rather than ahead of it. We regard this as the appropriate reading: the operator of Equation 2 works purely at the level of next-token distributions, requiring no access to weights, no shared parameterization and no merge coefficient to tune, and the aim of this work is to explain *why* such a rule composes skills and when it should fail, not to win a merging benchmark. The cost of operating at the distribution level is that inference requires $k+1$ forward passes where a merged model requires one.

The ablation isolates what the base-model division buys. Removing it costs 10.4 points on HumanEval+ and 2.4 on MBPP+, while on the two mathematics benchmarks the two rules are indistinguishable ($+1.1$ and -0.1). The base-corrected rule is therefore not a cosmetic variant of a product of experts: where the two experts disagree most about the background, dividing out the $k-1$ redundant copies of it is worth a large fraction of the total gain.

One asymmetry in Table 2 we are not able to explain. The composition exceeds both experts on HumanEval+ but sits below the math expert on GSM8K, and, as reported below, this does not correspond to a domain-level difference in how well the checkpoints satisfy Definition 1. We regard it as the main open empirical question raised by this work. We also experimented with composing more than two experts and observed a sharp degradation in generation quality, driven by disagreement about end-of-sequence behavior; this failure mode is described in Appendix G.2.2.

We now provide further details of the experiments with LLMs reported above. We consider three Hugging Face checkpoints: `google/gemma-2-2b` for p_{base}, `MergeBench/gemma-2-2b_math` for p_{math}, and `MergeBench/gemma-2-2b_coding` for p_{coding}. The MergeBench checkpoints are domain-specialized variants derived from `Gemma2 2B`. The submitted supplementary notebook loads the models with `AutoModelForCausalLM` and `AutoTokenizer`; Hugging Face access is supplied through an external `HF_TOKEN` environment variable, not hard-coded in the notebook.

We combine $p_{\text{base}}, p_{\text{math}}$, and p_{coding} using our proposed *selective composition* operator (Equation 2) to produce a merged model $\hat{p}$. The implementation asserts that all tokenizers have the same vocabulary size and token-id mapping. During generation it stores separate key-value caches for the base, math, and coding models, computes

$$\log \hat{p}_t = \log p_{\text{math},t} + \log p_{\text{coding},t} - \log p_{\text{base},t},$$

renormalizes with a softmax, and decodes greedily. We use `MAX_NEW=512`, `SAMPLE_MERGED=False`, `torch.manual_seed(0)`, float16 on CUDA and float32 otherwise, and `device_map="auto"` when CUDA is available. The base log-probabilities are clamped below at -10^4 for numerical stability.

G.2.1 Baselines

All baselines in Table 2 combine the same two fine-tuned checkpoints, with `google/gemma-2-2b` as the base where a base is required, and are evaluated with the protocol of Appendix G.2.2: same

prompts, same few-shot prefixes, greedy decoding, same generation budget. Writing $\theta_b, \theta_{\text{math}}, \theta_{\text{coding}}$ for the parameter vectors:

- **Task arithmetic** (Ilharco et al., 2023): $\theta_{\text{merged}} = \theta_b + \alpha\big[(\theta_{\text{math}} - \theta_b) + (\theta_{\text{coding}} - \theta_b)\big]$ with $\alpha = 1.0$, the exact parameter-space analogue of Equation 2 discussed in Appendix F.
- **TIES-merging** (Yadav et al., 2023): each task vector $\theta_i - \theta_b$ is trimmed by a magnitude threshold targeting its top 20% of entries, signs are elected coordinate-wise, and the surviving entries are averaged and added back with $\lambda = 1.0$. The realized kept fractions were 19.9% and 20.5% for the math and coding task vectors.
- **Model soup** (Wortsman et al., 2022): the uniform average $\frac{1}{2}(\theta_{\text{math}} + \theta_{\text{coding}})$ of the two fine-tuned endpoints.
- **Product of experts** (Hinton, 2002): identical to our generation code except that the score is $\log p_{\text{math},t} + \log p_{\text{coding},t}$, without subtracting $\log p_{\text{base},t}$, followed by softmax renormalization and greedy decoding. This is the ablation that isolates the base-model division in Equation 2.

MergeBench's own published numbers for `Gemma 2 2B` merge five experts rather than two, so they are not directly comparable with Table 2; the baseline runs above are therefore performed with the same two experts our composition uses.

G.2.2 Evaluation Protocol

Mathematics. Performance on mathematical reasoning is measured on two benchmarks: `GSM8K` and `MATH`. For `GSM8K`, the notebook calls `lm_eval.simple_evaluate` (Gao et al., 2024) with task `gsm8k`, `num_fewshot=8`, and `batch_size=1`; we report the flexible exact-match score. For the `MATH` benchmark, the notebook uses `DigitalLearningGmbH/MATH-lighteval`, configuration `all`, split `test`. It prepends a two-shot prompt and asks models to put the final answer in `\boxed{}`. Correctness is determined by exact agreement of the extracted boxed answer after light normalization.

Coding. For code generation, we evaluate on `HumanEval+` and `MBPP+` using `evalplus` (Liu et al., 2023). The notebook obtains problems via `get_human_eval_plus` and `get_mbpp_plus`, writes generated completions to `samples_humaneval.jsonl` and `samples_mbpp.jsonl`, and then runs `evalplus.evaluate` once with dataset `humaneval` and once with dataset `mbpp`. Performance is measured using `pass@1`, which considers whether the first generated solution passes all tests. Generated programs are executed against the provided unit tests to determine correctness; for reproduction, code execution should be run in an isolated environment as recommended by `evalplus`.

Uncertainty. Every benchmark is run on its full test set, of size $n = 1319$ (`GSM8K`), 5000 (`MATH`), 164 (`HumanEval+`) and 378 (`MBPP+`). Decoding is greedy and therefore deterministic, so there is no seed variance to report and the relevant uncertainty is sampling over the evaluation set. The error bars in Table 2 are binomial standard errors $\sqrt{\hat{q}(1-\hat{q})/n}$ computed from the number of correct problems, where $\hat{q}$ is the reported score. On the small coding sets these intervals are wide, and we read the table accordingly in Section 3.

G.2.3 Failure Mode with Additional Experts

We also experimented with extending the composition operator to more than two specialized models by adding additional experts to the merge. In practice, performance dropped very sharply once more models were included. The merged model often became unstable and continued generating gibberish even after producing what looked like a complete code solution. A likely explanation is disagreement among the models about the end-of-sequence token and termination behavior. When the experts do not agree on when generation should stop, their combined log-probabilities can keep assigning nontrivial mass to continuation tokens, causing the merged model to over-generate instead of cleanly stopping at the end of the answer. This suggests that reliable multi-model composition may require stronger alignment of stopping criteria or an explicit mechanism for handling termination. The two-expert regime is therefore the scope of our empirical claims.

G.3 Measuring the Factorization Assumptions

Definitions 1 and 4 require that, off its designated region, an expert *coincides with the background model*. Fine-tuned checkpoints descending from an internet-scale pretrained base cannot be expected to do so exactly, so the useful question is how far from it they are. It is also the only question the guarantee depends on: by Corollary 4 the defect of Definition 6 is driven by the deviations of the components off their own regions and by nothing else, so measuring those deviations is measuring how much of Theorem 4 is available. We measure this directly, per token, through the deviation $\epsilon_i(t)$ of Equation 6. On a mathematics prompt, Definition 1 requires the *coding* expert to be inert, $\epsilon_{\text{off}} \equiv 0$, and symmetrically on a coding prompt. We evaluate 200 prompts, 50 each from `MATH`, `GSM8K`, `HumanEval` and `MBPP`, fixing a continuation per prompt and reading ϵ from one teacher-forced forward pass of each of the three models over it, at 37,532 generated positions in total. The diagnostic needs only forward passes and no labels, so a practitioner can run it on a candidate set of experts before deciding to compose them.

Table 3: Per-token deviation ϵ from the background model, in nats, over $n = 100$ prompts per domain. Rows marked **off** are the ones Definition 1 requires to vanish.

ϵ (nats)	mean	p50	p90	p95	p99	max
p_{math} on math prompts (on-domain)	0.169	0.021	0.50	0.80	1.94	7.49
p_{coding} on math prompts (**off**)	0.109	0.015	0.33	0.55	1.12	5.64
p_{coding} on coding prompts (on-domain)	0.122	0.011	0.36	0.68	1.44	5.85
p_{math} on coding prompts (**off**)	0.073	0.008	0.21	0.37	0.86	4.39

The ordering the definitions require holds essentially universally. On 199 of the 200 prompts the in-domain expert deviates from the background model by more than the off-domain one ($p \approx 10^{-58}$, exact binomial against $1/2$), and the same $199/200$ holds if prompts are ranked by KL divergence or by total variation instead. Reading down a block of Table 3, this ordering is stable across the whole distribution rather than an artifact of averaging: the on-domain expert exceeds the off-domain one by 1.3–1.4$\times$ at the median, 1.45–1.84$\times$ at p90–p95, and still 1.3$\times$ at the maximum.

The magnitudes show the assumption is approximate, and the two experts differ sharply. The off-domain rows are small but plainly nonzero, so Definition 1 is satisfied only approximately, and comparing an expert's two rows asks whether fine-tuning stayed inside its own domain. The math expert is well localized and stably so: it perturbs the background model 2.31$\times$ more on mathematics than on code at the mean, and 2.16–2.63$\times$ from the median through p99. The coding expert is not: the ratio is 1.11$\times$ at the mean but 0.73$\times$ at the median, so at a typical token it perturbs the background model *more* on mathematics than on code. This is the concrete sense in which fine-tuning does not respect the disjointness that Definitions 1 and 4 posit; Appendix E.3 instantiates the bound of Theorem 4 at these values and reads off where it is informative and where it is not. It does not, however, line up with Table 2: the expert with the weaker localization is the one on whose domain the composition gains most, which is why we describe the benchmark asymmetry as unexplained.

A behavioral counterpart. Appendix G.3.2 reports a second, coarser diagnostic that checks the same structure at the level of decoded text: on a mathematics prompt, the off-domain expert should stay near the background model while the on-domain expert moves away (*divergence*), and the composition should track the on-domain expert (*tracking*). Under a deliberately unforgiving test, tracking holds on 74% of the 200 prompts and divergence on 62%, against a chance rate of 50% for each, with both holding jointly on 48% against a chance rate of 25% ($p = 1.9 \times 10^{-12}$). Separating the two conditions localizes which half of the assumption is fragile: the composition does follow the on-domain expert, whereas the premise that fine-tuning perturbs only its own domain is the weaker half, and on `GSM8K` it is at chance. The two diagnostics measure different objects, decoded continuations versus the conditionals the definitions constrain, and agree on the same conclusion: the structure posited by Definition 1 is present in real checkpoints and detectable well above chance, and it holds approximately rather than exactly.

This appendix gives the details of the two diagnostics reported above. Both use the same three checkpoints as above and the same 200 prompts: 50 each from `MATH`, `GSM8K`, `HumanEval` and

MBPP. Neither uses labels, and both require only forward passes, so either can be run on a candidate set of experts before deciding whether to compose them.

G.3.1 Conditional-Level Diagnostic

Definitions 1 and 4 constrain next-token conditionals, so the most direct measurement is at that level. For each prompt we fix a continuation, run one teacher-forced forward pass of p_{base}, p_{math} and p_{coding} over it, and read the per-token deviation $\epsilon_i(t)$ of Equation 6 at every position. Continuations are produced by greedy decoding and are capped at 256 tokens; ϵ is read only at generated positions, never on the prompt, which yields 37,532 scored positions in total. Because reused continuations are stored as text and re-tokenized, token identifiers need not reproduce bit-exactly across runs. Per-prompt summaries are means over the positions of that prompt; the quantiles in Table 3 are taken over all positions of a domain.

Table 4 reports how often the ordering that Definition 1 requires holds per prompt, that is, how often the in-domain expert deviates from the background model by more than the off-domain one, alongside the corresponding rate for the generation-level diagnostic of Appendix G.3.2. The overall rate of $199/200$ gives $p \approx 10^{-58}$ under an exact binomial test against $1/2$, and the same $199/200$ is obtained if prompts are ranked by the per-token KL divergence or total variation between the expert and background conditionals rather than by ϵ.

Table 4: Fraction of prompts on which the in-domain expert deviates from the background model more than the off-domain expert, under the two diagnostics.

Per prompt: $\bar{\epsilon}_{\text{in}} > \bar{\epsilon}_{\text{off}}$	MATH	GSM8K	HumanEval	MBPP	Overall
generation-level (Appendix G.3.2)	70%	56%	60%	62%	62%
conditional-level (Table 3)	100%	98%	100%	100%	**100%**

Two caveats are worth stating. First, the ordering test is ordinal: it asks only that the off-domain deviation be smaller than the in-domain one, whereas Definitions 1 and 4 ask that it be zero. The magnitudes in Table 3 are therefore the substantive measurement, and they are small but not zero. Second, ϵ in Equation 6 is evaluated at the realized token y_t, which is a single slice of the conditional; the per-token KL and total variation, which summarize the whole conditional, give the same $199/200$ ordering. The quantity entering Theorem 4 is the spread of the same log-ratio over the whole vocabulary, which dominates its value at any one token, so Table 3 understates it; Appendix E.3 makes the comparison precise.

G.3.2 Generation-Level Diagnostic

The second diagnostic tests the behavioral consequence of the same structure. Definition 1 makes a per-prompt prediction about decoded text: on a mathematics prompt, the coding expert should behave like the background model while the math expert diverges from it, and the composition should follow the math expert rather than the coding one. Writing sim for a similarity score between greedy generations, this reads

$$\textbf{Divergence:} \quad \mathrm{sim}(p_{\text{coding}}, p_{\text{base}}) > \mathrm{sim}(p_{\text{math}}, p_{\text{base}}),$$
$$\textbf{Tracking:} \quad \mathrm{sim}(\hat{p}, p_{\text{math}}) > \mathrm{sim}(\hat{p}, p_{\text{coding}}),$$

and symmetrically on a coding prompt. We take sim to be the mean of three scores: a character-level ratio, a token-level Jaccard index, and the cosine similarity of `all-MiniLM-L6-v2` sentence embeddings (Reimers & Gurevych, 2019). We generate greedily from the background model, both experts and the composition on each prompt, and check the two inequalities.

The criterion is deliberately unforgiving: one unsmoothed inequality per condition, no tolerance band, and both required for the strict column. Under a null of no factorization structure each condition holds with probability $1/2$ and the conjunction with probability $1/4$. The observed rates are well above chance: exact one-sided binomial tests give $p = 4.2 \times 10^{-4}$ for divergence, $p = 1.0 \times 10^{-11}$ for tracking and $p = 1.9 \times 10^{-12}$ for the conjunction.

Reporting the two conditions separately is what makes the diagnostic informative, because they come apart. Tracking holds at 74%, so the composition does follow the on-domain expert; divergence holds

Table 5: Generation-level diagnostic. Fraction of prompts satisfying each condition; chance rates are 50%, 50% and 25%.

Benchmark	Divergence	Tracking	Both (strict)
`MATH` ($n = 50$)	70%	70%	52%
`GSM8K` ($n = 50$)	56%	68%	42%
`HumanEval` ($n = 50$)	60%	72%	48%
`MBPP` ($n = 50$)	62%	84%	50%
Overall ($n = 200$)	**62%**	**74%**	**48%**

at only 62%, so the fragile half of the assumption is the premise that fine-tuning perturbs only its own domain. This is sharpest on `GSM8K`, where divergence is at chance (56%, $p = 0.24$) because the coding expert departs from the background model by about as much as the math expert does (mean similarity to base 0.393 and 0.394 respectively). Interference of this kind is expected when both experts are fine-tuned from a base model pretrained on internet-scale text, and it is the same phenomenon that Table 3 quantifies as the coding expert's 1.11× own-domain-to-other-domain ratio.

Finally, the diagnostic does not predict where composition helps at benchmark granularity. Strict satisfaction is essentially equal on mathematics and coding prompts (47% versus 49%, Fisher $p = 0.89$), and the divergence condition fails in both directions to a similar extent (63% on mathematics prompts, 61% on coding prompts), while the benchmark outcomes in Table 2 are strongly asymmetric. Whatever drives that asymmetry, it is not a domain-level difference in how well these checkpoints satisfy Definition 1.

G.4 Compute and Supplementary Material

Experiments were run on two NVIDIA H100 GPUs with 94 GB memory each. The supplementary material includes notebooks and a README with the dependency installation command, model identifiers, evaluation commands, decoding settings, baseline merge configurations, the diagnostic scripts, and the expected result tables. Exact wall-clock times were not logged for every benchmark run. An anonymized copy of the code is included with the supplementary material.

H FIGURES

$$\texttt{Problem : Show that } \frac{\sec\theta - \cos\theta}{\cot\theta + \tan\theta} = \sin^3\theta$$

$$\texttt{Solution :}$$

$$\begin{aligned}\frac{\sec\theta - \cos\theta}{\cot\theta + \tan\theta} &= \frac{\frac{1}{\cos\theta} - \cos\theta}{\frac{1}{\tan\theta} + \tan\theta} = \frac{\frac{1-\cos^2\theta}{\cos\theta}}{\frac{1}{\tan\theta} + \tan\theta} = \frac{\frac{\sin^2\theta}{\cos\theta}}{\frac{1}{\tan\theta} + \tan\theta} = \frac{\frac{\sin^2\theta}{\cos\theta}}{\frac{1+\tan^2\theta}{\tan\theta}} \\ &= \frac{\sin^2\theta}{\cos\theta} \cdot \frac{\tan\theta}{1+\tan^2\theta} = \frac{\sin^2\theta}{\cos\theta} \cdot \frac{\tan\theta}{\sec^2\theta} = \frac{\sin^2\theta}{\cos\theta} \cdot \frac{\sin\theta}{\cos\theta} \cdot \cos^2\theta \\ &= \sin^3\theta.\end{aligned}$$

Figure 1: A trigonometry problem requiring the use of basic algebraic manipulation and the identities $\sin^2(x) + \cos^2(x) = 1$ and $\sec^2(x) - \tan^2(x) = 1$.

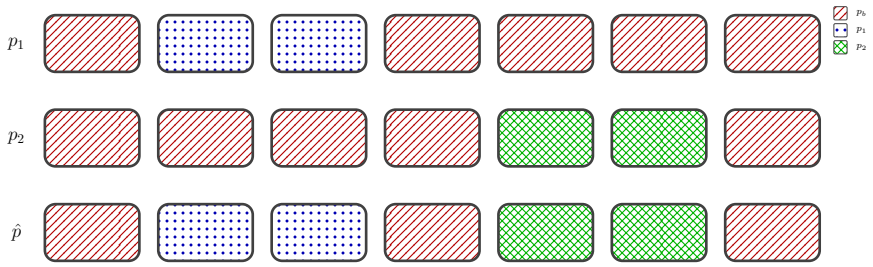


Figure 2: Illustration of two specialized models p_1 and p_2 satisfying factorized conditionals (Definition 1) and their composition $\hat{p}$ under Equation 2. Each column represents a token coordinate in the generated sequence. Blue cells mark the active mask M_1 where p_1 differs from the background model p_b, green cells mark the active mask M_2 where p_2 differs from p_b, and red cells indicate coordinates on which the corresponding specialized model agrees with p_b. Because the active masks are disjoint, the likelihood-ratio composition cancels the inactive factors and yields a composed model that follows p_1 on M_1, follows p_2 on M_2, and follows p_b elsewhere, giving the projective behavior stated in Theorem 1.