TREATING ALIGNMENT AS AN IN-CONTEXT APPROXIMATION PROBLEM

Anonymous authors
Paper under double-blind review

ABSTRACT

Alignment changes model weights, yet much of what steers model behavior at test time happens in context. Their immediate relationship is so far little understood. We study when context-induced distribution shifts can reproduce those induced by parameter updates. We first show that, in linear attention, a fixed KV prefix exactly reproduces a low-rank bilinear update, and characterize where this fails, including updates whose effects depend on task content. This motivates task-dependent context, where relevant clauses are retrieved from an approved constitution. We identify three conditions for close approximation: the retrieved clauses fit the update, the frozen model realizes their declared effects, and the clauses retain those effects when combined. Controlled experiments agree with this picture: retrieving the relevant clause recovers the trained update in frozen Qwen3 models and brings them much closer to their aligned counterparts, while presenting the full constitution can introduce errors from irrelevant and competing rules. Finally, we identify two fundamental limits of in-context alignment. Exact bounded clause selection is NP-complete in the worst case, and some training-induced shifts lie in directions that no deterministic context can follow.

1 INTRODUCTION

A language model's policy can be changed through its weights or through its context. Fine-tuning on safety demonstrations makes language models less likely to comply with harmful requests (Bianchi et al., 2024; Han et al., 2024), while adversarial prompts can make safety-aligned models comply without changing their weights (Andriushchenko et al., 2025). In-parameter training and in-context conditioning therefore provide two distinct ways of moving a model towards or away from a target behavior.

This suggests a relationship between context-induced and weight-update-induced policy shifts that is not yet well characterized. Recent work has begun to exploit one direction of this relationship. Self-Distillation Policy Optimization (SDPO), for instance, trains a model to internalize a policy shift first induced through context (Hübotter et al., 2026). Here, we mathematically explore the converse direction, asking whether, and under what conditions, an appropriate context can approximately reproduce the policy shift caused by a training update. Through theoretical analysis and empirical experiments, we show that such an approximation is indeed possible, while characterizing its limits and principal sources of error.

We make three contributions. First, we connect errors in next-token log-probability shifts to differences in complete response distributions and characterize when a key–value prefix exactly reproduces a bilinear update (Proposition 1, Theorem 3). Second, for task-specific clauses retrieved from a constitution, we identify theoretically and observe empirically three conditions for close approximation: the retrieved clauses fit the update, the frozen model realizes their declared effects, and the clauses retain those effects when combined (Proposition 6, Section 6). Third, we show that exact bounded clause selection is NP-complete and that some training-induced shift directions cannot be reproduced by any deterministic context (Proposition 7, Theorem 8).

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

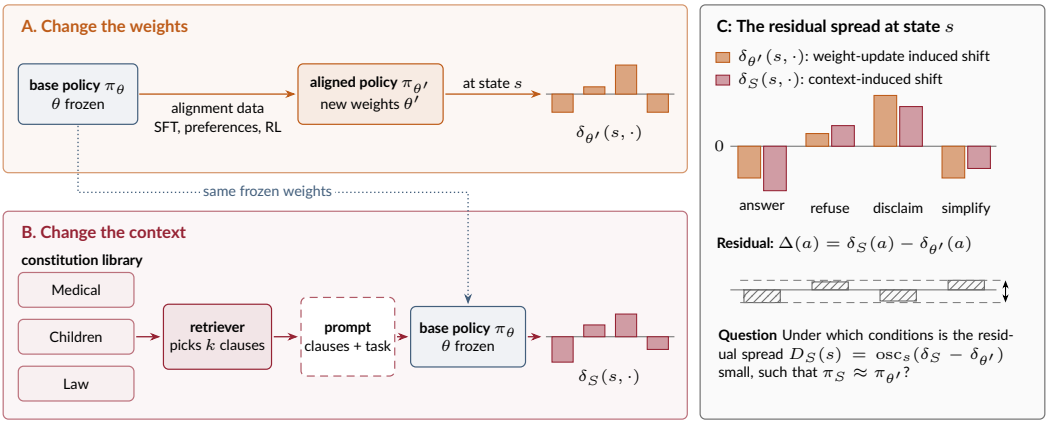

Figure 1: Schematic comparison of weight-update-induced and context-induced shifts.

2 Related Work

Prior work provides several partial links between contextual adaptation and parameter-space learning, without establishing that a context can reproduce a given training-induced policy shift. In linear attention, context can be represented as an additive fast-weight update (Schlag et al., 2021). He et al. (2022) place adapters, prefix tuning, and LoRA in a common framework of modifications to hidden representations (Li & Liang, 2021; Lester et al., 2021; Hu et al., 2022), while Petrov et al. (2024b) show that prefixes preserve relative attention among content tokens within a layer and restrict attention-output changes. Work on prompt expressivity and the softmax bottleneck further characterizes what contexts and model outputs can represent (Wang et al., 2023; Petrov et al., 2024a; Yang et al., 2018; Finlayson et al., 2024). A related line shows that self-attention can implement gradient descent on regression tasks (von Oswald et al., 2023) and interprets in-context learning as implicit fine-tuning (Dai et al., 2023). More direct conversions map demonstrations to biases in linearized transformers or, under specified architectural assumptions, contextual effects to implicit MLP weight updates (Chen et al., 2024; Dherin et al., 2025; Mazzawi et al., 2025; Goldwaser et al., 2025). How these mechanisms relate matters especially for alignment. Contextual and parameter interventions can alter safety behavior (Andriushchenko et al., 2025; Qi et al., 2024; Han et al., 2024), while prompts can elicit behavior that alignment has not eliminated (Wolf et al., 2024).

Existing methods have primarily exploited one direction of this relationship: from context-induced behavior to weights. Context distillation trains a model to reproduce behavior elicited by instructions or other context without that context at inference time (Askell et al., 2021; Snell et al., 2022). Self-Distillation Policy Optimization similarly distills a policy's feedback-conditioned distribution into its weights (Hübotter et al., 2026), while written principles can guide alignment training (Bai et al., 2022; Guan et al., 2024).

These works provide rich accounts of in-context adaptation and increasingly use context-induced behavior as a training signal. In-context examples can also align base models (Lin et al., 2024), although this alignment can fall short of fine-tuning (Zhao et al., 2025). The reverse direction, however, remains less understood: given a particular training-induced policy shift, can a context reproduce it, and under what conditions? In this work, we address this question.

3 Setup

This section introduces the setting and the notation used throughout our theoretical analysis.

Models and states. Tokens come from a vocabulary of size V. A task x is drawn from a distribution D over a finite set of tasks, with $D(x) > 0$ for every task, and a response is a token sequence $y = (y_1, \ldots, y_T)$. A *state* $s = (x, y_{<t})$ consists of the task and the response generated so far. At each state s, the base model defines a next-token distribution $\pi_\theta(v \mid s)$ and the aligned model defines $\pi_{\theta'}(v \mid s)$.

Contexts. Supplying a context S to the frozen base model induces the policy $\pi_{\theta,S}(v \mid s)$. We study two context mechanisms. The first is a fixed *key–value prefix*, consisting of free key and value vectors appended to those processed by an attention head, with the same prefix used for every task. The second is a task-dependent *hard prompt* obtained by retrieving clauses from a constitution. A constitution is an approved library $\mathcal{C} = (c_1, \ldots, c_M)$ of M natural-language clauses. A retriever ρ maps each task x to a selection $R = \rho(x) \subseteq \{1, \ldots, M\}$ with $|\rho(x)| \leq k$ which are prepended to the task as a hard prompt.

Approximation error. Our goal is to determine how closely a context-induced policy can reproduce the aligned policy over complete responses. Write

$$\pi_{\theta,S}(y \mid x) = \prod_{t=1}^{T} \pi_{\theta,S}(y_t \mid x, y_{<t}),$$

and define $\pi_{\theta'}(y \mid x)$ analogously. For a fixed context S, we consider its *approximation error* as

$$d(S) := \mathbb{E}_{x\sim D}\left[\mathrm{TV}\left(\pi_{\theta,S}(y \mid x), \pi_{\theta'}(y \mid x)\right)\right], \tag{3.1}$$

where TV is the total variation distance between two response distributions. For a retriever ρ, define similarly

$$d(\rho) := \mathbb{E}_{x\sim D}\left[\mathrm{TV}\left(\pi_{\theta,R}(y \mid x), \pi_{\theta'}(y \mid x)\right)\right].$$

Induced shifts. The approximation error is defined over complete responses, but our analysis begins locally, at the level of next-token probabilities. At a state s and token v, define the shifts induced by the aligned weights and by the context as

$$\delta_{\theta'}(s, v) := \log \frac{\pi_{\theta'}(v \mid s)}{\pi_{\theta}(v \mid s)}, \qquad \delta_S(s, v) := \log \frac{\pi_{\theta,S}(v \mid s)}{\pi_{\theta}(v \mid s)}. \tag{3.2}$$

Thus, $\delta_{\theta'}(s, \cdot)$ and $\delta_S(s, \cdot)$ are vectors in $\mathbb{R}^V$ describing how the aligned weights and the context, respectively, change the base model's next-token log probabilities.

Oscillation. To compare two shifts, we need a measure that respects an important invariance of the softmax: adding the same constant to every token logit leaves the output distribution unchanged. Consequently, two induced shifts that differ only by a constant over the vocabulary should be treated as equivalent. We therefore measure their difference using the subadditive seminorm *oscillation*, and define *residual oscillation* as:

$$D_S(s) := \mathrm{osc}_s(\delta_S - \delta_{\theta'}) = \max_v[\delta_S(s, v) - \delta_{\theta'}(s, v)] - \min_v[\delta_S(s, v) - \delta_{\theta'}(s, v)], \tag{3.3}$$

with $\mathrm{osc}(w) \in \mathbb{R}$ for $w \in \mathbb{R}^V$ and $\mathrm{osc}(w) = 0$ for constant w.

Translating shift differences to the approximation error. The approximation error $d(S)$ measures disagreement over complete responses, whereas $D_S(s)$ captures the local mismatch between the two interventions at a single state. To connect the local quantity that we can analyze to the response-level quantity of interest, we combine Lemmas 9 and 10 (Appendix) with the following proposition. Intuitively, small residual oscillations imply similar next-token distributions, which in turn imply similar complete-response distributions.

Proposition 1 (bridge). *For every context S, with $s_t = (x, y_{<t})$,*

$$d(S) \leq \mathbb{E}_{x\sim D} \sum_{t=1}^{T} \mathbb{E}_{y_{<t}\sim\pi_{\theta'}} \tanh\left(\frac{1}{4} D_S(s_t)\right) \leq \frac{T}{4} \max_s D_S(s). \tag{3.4}$$

For the proof, see Appendix B.3.

For long responses, Proposition 11 in Appendix B.4 gives a complementary root-sum-square bound that improves the worst-case dependence on the response length from T to $\sqrt{T}$.

This effectively establishes residual oscillations as the core object of our further investigations.

4 CAN WE APPROXIMATE ALIGNMENT UPDATES WITH TASK-AGNOSTIC CONTEXT?

We first consider a context that is fixed across tasks: a key–value prefix (Li & Liang, 2021) in a linear-attention model. This setting isolates how closely a context can reproduce a training-induced shift, without also modeling how a frozen language model interprets natural-language instructions. A prefix of length n acts through a fixed operator of rank at most n, which characterizes when a prefix reproduces a bilinear update exactly and how the error depends on the part of the update that a shorter prefix cannot capture. The same analysis shows where fixed prefixes fail: when the effect of the update depends on task content, one fixed prefix cannot in general reproduce it across tasks, which motivates the task-dependent retrieval of the next section.

4.1 WHEN EXACT REPRODUCTION WITH FIXED KEY–VALUE PREFIXES IS POSSIBLE

We first define the linear attention head in which we compare the training update with a key–value prefix. At a state s, let

$$\mathbf{c}_1, \dots, \mathbf{c}_{|s|} \in \mathbb{R}^{d_{\text{in}}}$$

be the embeddings of the tokens in the state. We stack them as the rows of the content matrix $C_s \in \mathbb{R}^{|s| \times d_{\text{in}}}$. The key and value maps $W_K, W_V \in \mathbb{R}^{d_{\text{in}} \times d}$ give

$$K_s = C_s W_K \in \mathbb{R}^{|s| \times d}, \qquad V_s = C_s W_V \in \mathbb{R}^{|s| \times d}.$$

Each row of K_s is the key of one token, and the corresponding row of V_s is its value. For a query $\mathbf{q}(s) \in \mathbb{R}^d$, unnormalised linear attention produces the head output

$$\mathbf{o}(s)^\top = \mathbf{q}(s)^\top K_s^\top V_s = \mathbf{q}(s)^\top (C_s W_K)^\top (C_s W_V) \in \mathbb{R}^{1 \times d}. \tag{4.1}$$

A linear layer $G \in \mathbb{R}^{d \times V}$ maps the head output to vocabulary logits, $z(s, \cdot) = \mathbf{o}(s)^\top G$. Writing $\mathbf{g}_v$ for the v-th column of G, the logit of token v is $z(s, v) = \mathbf{o}(s)^\top \mathbf{g}_v$.

As in Section 3, shifts that differ only by a constant shared by all tokens produce the same output distribution. We write $f \equiv f'$ when $f(s, v) - f'(s, v)$ does not depend on v. The oscillation removes exactly these constant shifts.

Within this setting, we constrain the shift that we want to reproduce:

Assumption 2 (bilinear decomposition)**.** Let $\delta_{\theta'}$ be bilinearly decomposable into

$$\delta_{\theta'}(s, v) \equiv \mathbf{q}(s)^\top B\, \mathbf{g}_v + \nu(s, v)$$

with $B \in \mathbb{R}^{d \times d}$ and *non-bilinear remainder* $\bar{\nu} := \max_s \operatorname{osc}_s \nu$ over reachable states. Let Π_Q project onto the span of the reachable queries and Π_G onto the complement of the hidden directions $\mathbf{w}$ for which $\mathbf{w}^\top \mathbf{g}_v$ does not depend on v. The *canonical operator* is $B^\star = \Pi_Q B\, \Pi_G$.

Here, B describes the fixed bilinear part of the training update, while ν contains everything that does not have this form. The quantity $\bar{\nu}$ measures the largest token-dependent variation in that remainder. When $\bar{\nu} = 0$, the remainder is constant across tokens at every state and therefore has no effect on the output distribution. This is the strict bilinear case.

Some parts of B may be unreachable by every query, while others shift every token equally. Neither affects residual oscillation. The projectors remove exactly these parts: Π_Q restricts B to the queries that actually occur, and Π_G removes directions that shift every token logit equally. Thus $B^\star = \Pi_Q B \Pi_G$ produces the same token-dependent shifts as B on every reachable state and has the least rank among all matrices that do so (Appendix C.4).

The assumption describes the structure of the training update that we want to reproduce. We now ask whether a key–value prefix can supply the same operator.

Constructing the key–value prefix. The full derivations of this paragraph, step by step, are in Appendix C.3. A key–value prefix of length n supplies n additional keys and values, collected as the rows of

$$P_K, P_V \in \mathbb{R}^{n \times d}.$$

Adding these rows to the keys and values of the state gives

$$K_{s,P} = \begin{bmatrix} K_s \\ P_K \end{bmatrix}, \qquad V_{s,P} = \begin{bmatrix} V_s \\ P_V \end{bmatrix} \in \mathbb{R}^{(|s|+n)\times d}.$$

Multiplying the stacked matrices gives $K_{s,P}^\top V_{s,P} = K_s^\top V_s + P_K^\top P_V$ (more details in Appendix C.3). Let $B_P = P_K^\top P_V \in \mathbb{R}^{d\times d}$. The head output with the prefix is therefore

$$\mathbf{o}_P(s)^\top = \mathbf{q}(s)^\top K_{s,P}^\top V_{s,P} = \mathbf{o}(s)^\top + \mathbf{q}(s)^\top B_P. \tag{4.2}$$

Thus, the prefix adds $\mathbf{q}(s)^\top B_P$ to the head output. The query changes with the state, but B_P does not. The same matrix acts at every state and for every task.

We next pass this new head output through the linear layer G. The logit with the prefix is

$$z_P(s,v) = \mathbf{o}_P(s)^\top \mathbf{g}_v = z(s,v) + \mathbf{q}(s)^\top B_P \mathbf{g}_v. \tag{4.3}$$

At the level of logits, the prefix therefore adds $\mathbf{q}(s)^\top B_P \mathbf{g}_v$ to token v.

The induced shift δ_P is defined using log-probabilities rather than logits. Passing from logits to log-probabilities introduces a softmax normalizer that depends on s but not on v. It therefore disappears under the oscillation (derivation in Appendix C.3).

Because $B_P = P_K^\top P_V$ is the product of a $d \times n$ matrix and an $n \times d$ matrix, its rank is at most n. Thus

$$\delta_P(s,v) \equiv \mathbf{q}(s)^\top B_P\, \mathbf{g}_v, \qquad \operatorname{rank} B_P \le n. \tag{4.4}$$

Conversely, every matrix of rank at most n can be factored as $P_K^\top P_V$. Hence a length-n prefix can realize any fixed bilinear operator of rank at most n.

The comparison with the training update is now direct. In the strict case $\bar\nu = 0$, the update acts through the canonical operator $B^\star$, while the prefix acts through B_P. Exact reproduction is possible if a prefix can realize $B^\star$. Since a prefix of length n can realize exactly the operators of rank at most n, the relevant question is whether $n \ge \operatorname{rank} B^\star$.

If the prefix is too short or the update has a non-bilinear remainder, we ask for the smallest residual a length-n prefix can achieve. At state s, the residual of a candidate operator B' is measured by $\operatorname{osc}_s\left(\mathbf{q}(s)^\top B' \mathbf{g}_\cdot - \delta_{\theta'}(s,\cdot)\right)$. Taking the largest residual over reachable states and then choosing the best operator of rank at most n gives

$$H_n := \inf_{\operatorname{rank} B' \le n} \max_s \operatorname{osc}_s\left(\mathbf{q}(s)^\top B' \mathbf{g}_\cdot - \delta_{\theta'}(s,\cdot)\right). \tag{4.5}$$

Thus, H_n is the best worst-state approximation error of a length-n prefix. The following theorem characterizes when it is zero and directly bounds the resulting policy approximation error using the first singular value of $B^\star$ that a rank-n prefix leaves out.

Theorem 3 (Approximation error with key–value prefixes)**.** *Under Assumption 2 and the normalization of Assumption 13 (queries of norm at most one, $g = \max_v \|\mathbf{g}_v\|_2$, readout visibility κ, and query richness γ), let $\sigma_1 \ge \sigma_2 \ge \cdots$ be the singular values of $B^\star$. For $n \ge 0$, define*

$$E_n^{\mathrm{KV}} = \inf\left\{d(P) : P \text{ is a fixed key–value prefix of length at most } n\right\}.$$

Then:

(i) If $\bar\nu = 0$, then $H_n = 0$ if and only if $n \ge \operatorname{rank} B^\star$. Thus the minimum length of a key–value prefix that exactly reproduces an exactly bilinear update is $\operatorname{rank} B^\star$.

(ii)

$$E_n^{\mathrm{KV}} \le \min\left\{1, \frac{\sqrt{T}}{4}\left(2g\,\sigma_{n+1}(B^\star) + \bar\nu\right)\right\}.$$

(iii) For every fixed key–value prefix P of length at most n, some task x has, at its first token,

$$\operatorname{osc}\left(\delta_P - \delta_{\theta'}\right) \ge \Delta_n := \left[\gamma\kappa\,\sigma_{n+1}(B^\star) - \bar\nu\right]_+.$$

If the two tokens attaining this oscillation each have aligned probability at least m, then

$$d(P) \ge D(x)\, m\left(1 - e^{-\Delta_n/2}\right).$$

For the proof, see Appendix C.5.

4.2 WHERE FIXED KEY–VALUE PREFIXES FAIL

Theorem 3 names what close approximation with key-value prefixes needs: a bilinear update, of rank no larger than the prefix, acting through one matrix for every task. Each requirement can fail.

High rank and non-bilinear updates. For an exactly bilinear update, every prefix shorter than $\operatorname{rank} B^\star$ leaves some error. If the singular values of $B^\star$ decay slowly, this error can remain large even for a fairly long prefix. Increasing the prefix length helps only until $n = d$, because every $d \times d$ matrix has rank at most d. If the update lies outside the bilinear image, adding more prefix positions cannot remove the remaining error.

Updates that depend on task content. If the training-update's bilinear operator changes with the task content, a fixed prefix operating always with the same B_P at every state cannot in general follow such a change. As shown in the following proposition, a value-weight update provides a simple example of this failure case.

Proposition 4 (Limitation of fixed prefixes under value-weight updates)**.** *Let θ' differ from θ only by $W_V \mapsto W_V + \Lambda_V$ at this head. Then $\delta_{\theta'}(s, v) \equiv \mathbf{q}(s)^\top B_s\, \mathbf{g}_v$ with $B_s = W_K^\top C_s^\top C_s\, \Lambda_V$, a core that depends on the state through $C_s^\top C_s$. For any fixed prefix P and any two tasks x_1, x_2 with the same first-token query $\mathbf{q}$,*

$$\max_{i \in \{1,2\}} \operatorname{osc}\left(\delta_P - \delta_{\theta'}\right)\big|_{x_i} \;\geq\; \tfrac{1}{2} \operatorname{osc}_v\left(\mathbf{q}^\top (B_{x_1} - B_{x_2})\, \mathbf{g}_v\right). \tag{4.6}$$

For the proof, see Appendix C.7.

In other words, a fixed prefix produces a shift from the current query through the same matrix B_P at every task, which breaks as soon as two tasks require different shifts. A context chosen separately for each task does not face this restriction. This motivates a task-dependent retrieval setup which we discuss next.

5 CAN WE APPROXIMATE ALIGNMENT UPDATES WITH TASK-DEPENDENT CONTEXT?

The previous section considered a fixed key–value prefix that must use the same operator for every task. We now move to a task-dependent context. Instead of compiling the whole alignment update into one fixed prefix, we store natural-language clauses in an approved constitution and retrieve the clauses that are relevant to the current task.

Let the constitution contain M clauses. For each task x, a retriever ρ selects a set

$$\rho(x) \subseteq \{1, \dots, M\}, \qquad |\rho(x)| \leq k.$$

The selected clauses form the hard prompt $\mathcal{C}[\rho(x)]$ in front of the task, which induces the shift $\delta_{\mathcal{C}[\rho(x)]}$. The context is task dependent because two tasks x_1 and x_2 may receive different selections: $\rho(x_1) \neq \rho(x_2)$. This allows the retriever to supply different rules even when a fixed prefix would have to use the same operator for both tasks.

To analyze this construction, we distinguish what each clause is intended to do from what the frozen model actually does when it reads the clause.

Clause j has a *declared effect* $u_j(s, v)$: the shift that the constitution says the clause should produce at state s for token v. We assume that this declared effect separates into a scope and a token preference.

Assumption 5 (scoped rank-one effects)**.** $u_j(s, v) = a_j(s)\, \mathbf{w}_j(v)$ with a scope $a_j(s) \in [0, 1]$ and a token preference $\mathbf{w}_j \in \mathbb{R}^V$. The certified impact of clause j at state s is $a_j(s) \operatorname{osc}(\mathbf{w}_j)$.

The scope $a_j(s)$ determines where the clause applies, while $\mathbf{w}_j$ determines which tokens it favors when it applies. For a selection R, the declared effects add to

$$u_R(s, v) = \sum_{j \in R} u_j(s, v).$$

This is the shift that the selected clauses are intended to produce. The assumption constrains only this declaration. It does not assume that the frozen model follows the clauses correctly.

Realized clause effects and interpreter defects. For a selection R, write $\delta_R = \delta_{\mathcal{C}[R]}$ for the shift induced by its hard prompt. The *realized effect* of clause j is $\delta_j = \delta_{\{j\}}$, the shift produced by the frozen model when clause j is shown on its own. For a task x, let Σ_x be its reachable states and define

$$\eta_j(x) := \max_{s \in \Sigma_x} \operatorname{osc}_s \big(u_j - \delta_j\big).$$

The interpreter defect $\eta_j(x)$ measures how far the model's realized effect is from the clause's declared effect on task x. It is zero when the model follows the declaration up to constants shared by all tokens.

Interactions between clauses. Even if the model follows every clause correctly in isolation, their effects need not add when several clauses are shown together. For a selection $\rho(x) = R$, define

$$I_R := \delta_R - \sum_{j \in R} \delta_j, \qquad \iota(x, R) := \max_{s \in \Sigma_x} \operatorname{osc}_s(I_R).$$

The interaction term $\iota(x, R)$ measures how much the jointly rendered selection differs from the sum of the clauses' individual realized effects. It is zero when the clauses compose additively on task x.

Clause fit. The remaining question is whether the clauses selected by the retriever are the right clauses for the task. Define

$$\mathcal{A}(x, R) := \max_{s \in \Sigma_x} \operatorname{osc}_s(\delta_{\theta'} - u_R). \tag{5.1}$$

The clause fit $\mathcal{A}(x, R)$ measures how well the declared effects of the selected clauses reproduce the alignment update.

Three sources of error. As the following proposition shows, the approximation error of task-dependent retrieval is in direct relation to the previously outlined quantities.

Proposition 6 (decomposition)**.** *For any retriever ρ with $|\rho(x)| \le k$,*

$$d(\rho) \le \frac{\sqrt{T}}{4}\, \mathbb{E}_{x \sim D}\Big[\underbrace{\mathcal{A}(x, \rho(x))}_{\textit{clause fit}} + \underbrace{\textstyle\sum_{j \in \rho(x)} \eta_j(x)}_{\textit{interpreter}} + \underbrace{\iota(x, \rho(x))}_{\textit{interaction}} \Big], \tag{5.2}$$

and the same holds with $T/4$ in place of $\sqrt{T}/4$. In a rollout form the factor is $\frac{1}{4}$ and each maximum over Σ_x becomes a sum over positions of expectations along the aligned model's trajectories.

For the proof, see Appendix D.1.

The proposition suggests three distinct ways to improve an in-context alignment system. First, the constitution should cover what alignment teaches and the retriever should select the relevant clauses for each task. Second, the frozen model should realize the intended effects of those clauses. Third, the rules should be as non-overlapping and non-conflicting as possible, reducing potential interactions when they are presented together. If these conditions make the clause-fit, interpreter and interaction terms small, Proposition 6 guarantees that the context-induced policy is close in total variation to the trained policy. Section 6 examines this empirically.

6 Experiments

Setup. To isolate the mechanisms in Proposition 6 we need a controlled target behavior that is absent from the base model and can be expressed as explicit clauses. Existing safety fine-tunes are unsuitable as they mix many behaviors without a direct constitution relation required for our controlled experiment. We therefore construct a synthetic constitution in which each clause maps an invented disease to a unique doctor, e.g., "If the user asks about Zumgaimann syndrome, answer with exactly this sentence and nothing else: Please ask Dr. Pirleiholt about this condition." We fine-tune Qwen3-1.7B and 8B (Yang et al., 2025) with LoRA on $5{,}264$ behavior-demonstration question–answer pairs for 256 clauses, while withholding the clauses themselves from training. Constitutions range from 16 to 896 clauses. We then compare the resulting aligned model to the frozen base model given no context, the full constitution, or the *oracle context* containing only the

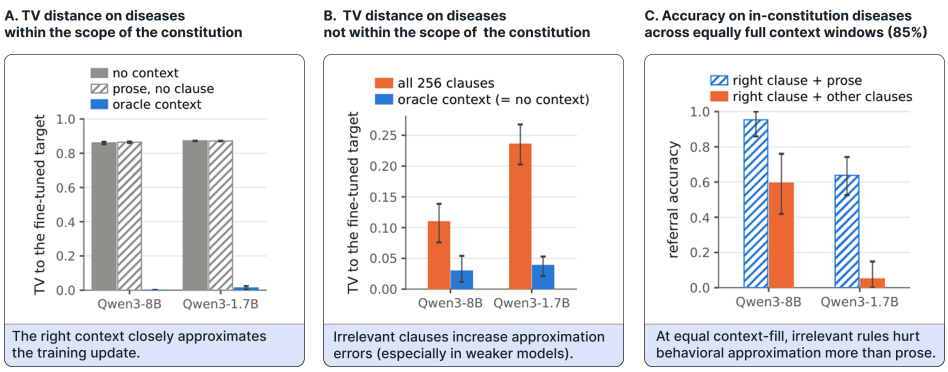

Figure 2: **The oracle clause against the whole constitution.** Qwen3 targets are fine-tuned, without seeing the clauses, to name one doctor per invented disease. Every context is shown to the frozen model. A) Diseases covered by a constitution of 256 clauses: per-token TV to the target, with prose as long as the constitution as a control. B) The same on real diseases, which no clause covers, so the oracle context is empty. C) Constitutions of 896 clauses, over 85% of the 32,768-token context window: referral accuracy when every line other than the asked-about clauses holds another clause or prose of the same token count (one arrangement). Bars: 95% intervals over diseases.

relevant clause. On held-out question wordings, we report per-token TV over the first $T = 8$ answer tokens and referral accuracy (share of answers naming the right doctor). Brackets are 95% intervals over diseases; Appendix F gives the full protocol and OLMo-3-7B (Team Olmo, 2025) results.

Throughout, we read each clause as declaring an effect only on questions about the disease it names, and none elsewhere. The comparisons below probe the phenomena behind the clause-fit, interpreter and interaction terms of Proposition 6. They report mean per-token TV and referral rates and do not measure those worst-state quantities.

Results. *The oracle clause recovers the trained rule.* On the diseases the 256 clauses cover, the oracle clause brings the frozen Qwen3-8B from TV 0.861 with no context to 0.000 and Qwen3-1.7B from 0.872 to 0.013, while prose as long as the constitution leaves both where they were (Figure 2a). The targets name the right doctor in 0.99 and 0.96 of their sampled answers, the frozen models under the oracle clause in 1.00 and 0.95, and over whole eight-token answers, which carry the doctor's name, the two differ by at most 0.003 and 0.101 in TV (eight times the per-token TV, Lemma 10).

Irrelevant clauses move the model away from the target where no clause applies (interpreter term). On questions about real diseases, which no clause covers, the oracle context is empty and leaves the frozen models as far from their targets as fine-tuning moved them (TV 0.029 and 0.038), while all 256 clauses raise the TV to 0.109 [0.076, 0.139] and 0.236 [0.203, 0.268] (Figure 2b). For Qwen3-1.7B the shift is a wrong action: it names one of the constitution's doctors in every answer on an eponymous disease and in 0.83 of those on other real diseases. Qwen3-8B never refers a real disease; only its wording changes, which counts as distance to the target but is not wrong in itself.

For Qwen3, many clauses fail through the other rules, not through their length (interpreter and interaction terms). With 896 clauses, which fill more than 85% of the context window, Qwen3-8B names the right doctor in 0.59 [0.42, 0.76] of answers, and in 0.95 [0.86, 1.00] when prose of the same length replaces the other clauses (Figure 2c); the rest mostly name no doctor (0.30 of all answers), few another clause's doctor (0.01). Qwen3-1.7B names the right doctor in only 0.05 of answers under the clauses and 0.64 under prose, so for the small reader length costs too, but the other rules cost most. The prose control does not separate the other clauses' effects from their interaction.

7 Global constraints on in-context alignment

The preceding sections studied when retrieved clauses work and how the three sources of approximation error appear in practice. We show two broader constraints that limit how far in-context shifts can reach.

7.1 Exact clause retrieval is NP-complete

Suppose the frozen model realizes every clause exactly and clauses never interact, so that $\eta_j = 0$ and $\iota = 0$ and only the clause-fit term remains. Choosing the exactly right clauses is still hard.

Proposition 7 (hardness)**.** *Given declared effects with indicator scopes and the update $\delta_{\theta'}$ at the states of one task, deciding whether some selection R with $|R| \le k$ has $\mathcal{A}(x, R) = 0$ is NP-complete, already for two-token outputs. It remains NP-complete when the selected clauses may carry arbitrary real coefficients.*

For the proof, see Appendix D.2.

7.2 Some alignment updates are beyond every context

The previous sections bound how well particular contexts approximate an update. However, we find that some updates are out of reach of every context, whatever its length, and the reason is the model's last layer, the softmax bottleneck of Yang et al. (2018). Write the logits as $z_{\theta,S}(s) = G_{\text{out}}^{\top}\mathbf{h}_{\theta,S}(s) + \mathbf{b}$. A context may move the final hidden state $\mathbf{h}_{\theta,S}(s) \in \mathbb{R}^d$ anywhere, but it cannot change the readout $G_{\text{out}} \in \mathbb{R}^{d\times V}$ or the bias $\mathbf{b}$. When $V > d + 1$ there is a nonzero $\mathbf{a} \in \mathbb{R}^V$ with $G_{\text{out}}\mathbf{a} = \mathbf{0}$ and $\mathbf{1}^{\top}\mathbf{a} = 0$. Contracting the log-probabilities with such an $\mathbf{a}$ removes both the hidden state and the softmax normalizer, so the contraction is the same for every context.

Theorem 8 (frozen readout)**.** *Let $\mathbf{a} \neq \mathbf{0}$ satisfy $G_{\text{out}}\mathbf{a} = \mathbf{0}$ and $\mathbf{1}^{\top}\mathbf{a} = 0$. Then*

$$\mathbf{a}^{\top}\delta_S(s, \cdot) = 0 \qquad \text{for every state } s \text{ and every context } S. \tag{7.1}$$

If a first-token state $s_0 = (x_0, \varnothing)$ has $\mathbf{a}^{\top}\delta_{\theta'}(s_0, \cdot) \neq 0$, no context reproduces the aligned policy at s_0, and with $\eta = |\mathbf{a}^{\top}\delta_{\theta'}(s_0, \cdot)|/\|\mathbf{a}\|_1$ and m the least aligned probability on the tokens where $\mathbf{a}$ is nonzero, every context has $d(S) \ge D(x_0)\, m(1 - e^{-\eta})$.

For the proof, see Appendix E.1.

8 Discussion

Much of deployed alignment now lives in context, as system prompts and retrieved policies. Our results say when that substitutes for training and decompose its failures into context selection, interpretation and interference, while the hardness and readout results show where no context, however chosen, can replace in-parameter alignment.

These findings may motivate future research on the implications of this boundary for context-induced distillation, including the conditions under which context-reachable behaviors can be reliably transferred into parameters and whether such updates exhibit distinctive vulnerabilities to adversarial context or jailbreaks. More broadly, our mathematical framework helps separate alignment behaviors that can be expressed through context from those that inherently require changes to the model, providing a basis for studying how contextual and parametric alignment should interact in robustly aligned systems.

Our theoretical results rely on several idealizations. The exact prefix results hold for one unnormalized linear-attention head. The constitution results assume declared clause effects whose realization by a frozen language model remains an empirical question. The experiments are also deliberately controlled. Most targets are fine-tuned with LoRA to follow synthetic rules about invented diseases, use one seed and an 8-token evaluation horizon, and rely on a retriever that does not have to resolve overlapping scopes. The targets themselves name the right doctor in only 0.58–0.99 of their sampled answers, and every target prescribes a fixed reply, so free-text behaviors are not tested. A smaller study in which clause scope must be inferred from the disease is reported in Appendix F.4.

AI USE STATEMENT

Generative AI was used for literature synthesis, proof writing assistance, proof verification, LaTeX preparation, writing the experiment code, and generating the synthetic data of Section 6 (the invented diseases, clause libraries and question templates). All proofs were re-derived by hand, the code was verified and tested for correctness by one author, and the submitting authors take responsibility for the final content.

ETHICS STATEMENT

Alignment here means reproduction of the behavior of a declared target model. Neither the legitimacy nor the completeness of that target is established. Average total-variation error permits concentrated failures and is not a worst-case safety guarantee. The invented diseases and doctors of Section 6 are names filtered against ICD-10-CM and a dictionary, the real diseases used as out-of-scope questions are public disease names, and no patient or clinical data is involved.

REPRODUCIBILITY STATEMENT

Every proof is in Appendices A–E. The finite inequalities are checked by `scripts/check_algebra.py` in the source package, which needs only Python. Lemmas 9 (with its equality characterization), 10 and 12, Propositions 1 and 11, and Theorem 3 in full (part (i) with the attainment of the infimum defining H_n, parts (ii) and (iii) with the Eckart–Young–Mirsky theorem proved from Mathlib's spectral theorem, and their policy-error consequences), and the correctness of the reduction behind Proposition 7 with its real-coefficient form, are formalized in Lean 4 (de Moura & Ullrich, 2021) with Mathlib (The mathlib Community, 2020) in the directory `lean/` of the source package, which `lake build` compiles and whose `README.md` describes the layout. The measurements of Section 6 and Appendix F run from the same code (`experiments/`, configurations `configs/fict_*.json`) on one NVIDIA H200 per configuration, from public Qwen3 and OLMo-3 checkpoints; their result files, logs, configurations and data are archived with SHA-256 sums, and every table of that appendix and every number its prose quotes is written from those files by `fictional_report.py`. The runs use `torch` 2.14.0 (CUDA 13.0), `transformers` 5.17.0 and `peft` 0.21.0 in bfloat16; a fine-tuning run takes from half a minute to about 25 minutes, and a configuration's measurements 10 to 60 minutes.

REFERENCES

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned LLMs with simple adaptive attacks. In *International Conference on Learning Representations*, 2025.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In *International Conference on Learning Representations*, 2024.

Garrett Birkhoff. Extensions of Jentzsch's theorem. *Transactions of the American Mathematical Society*, 85(1):219–227, 1957. doi: 10.2307/1992971.

Brian K. Chen, Tianyang Hu, Hui Jin, Hwee Kuan Lee, and Kenji Kawaguchi. Exact conversion of in-context learning to model weights in linearized-attention transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6833–6846, 2024.

Samuel N. Cohen and Eliana Fausti. Hyperbolic contractivity and the Hilbert metric on probability measures. *arXiv preprint arXiv:2309.02413*, 2023.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can GPT learn in-context? Language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4005–4019, 2023. doi: 10.18653/v1/2023.findings-acl.247.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, Lecture Notes in Computer Science, pp. 625–635, 2021. doi: 10.1007/978-3-030-79876-5_37.

Benoit Dherin, Michael Munn, Hanna Mazzawi, Michael Wunder, and Javier Gonzalvo. Learning without training: The implicit dynamics of in-context learning. *arXiv preprint arXiv:2507.16003*, 2025.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. doi: 10.1007/BF02288367.

Matthew Finlayson, Xiang Ren, and Swabha Swayamdipta. Logits of API-protected LLMs leak proprietary information. *arXiv preprint arXiv:2403.09539*, 2024.

Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

Adrian Goldwaser, Michael Munn, Javier Gonzalvo, and Benoit Dherin. Equivalence of context and parameter updates in modern transformer blocks. *arXiv preprint arXiv:2511.17864*, 2025.

Melody Y. Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, et al. Deliberative alignment: Reasoning enables safer language models. *arXiv preprint arXiv:2412.16339*, 2024.

Tessa Han, Aounon Kumar, Chirag Agarwal, and Himabindu Lakkaraju. MedSafetyBench: Evaluating and improving the medical safety of large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 33423–33454, 2024.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2022.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pp. 4582–4597, 2021.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. In *International Conference on Learning Representations*, 2024.

Hanna Mazzawi, Benoit Dherin, Michael Munn, Adrian Goldwaser, Michael Wunder, and Javier Gonzalvo. Transmuting prompts into weights. *arXiv preprint arXiv:2510.08734*, 2025.

Leon Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960. doi: 10.1093/qmath/11.1.50.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2): 227–234, 1995. doi: 10.1137/S0097539792240406.

Aleksandar Petrov, Philip H. S. Torr, and Adel Bibi. Prompting a pretrained transformer can be a universal approximator. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 40523–40550, 2024a.

Aleksandar Petrov, Philip H. S. Torr, and Adel Bibi. When do prompting and prefix-tuning work? A theory of capabilities and limitations. In *International Conference on Learning Representations*, 2024b.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 2009. doi: 10.1561/1500000019.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9 of *Proceedings of Machine Learning Research*, pp. 661–668, 2010.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9355–9366, 2021.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Team Olmo. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pp. 367–381, 2020. doi: 10.1145/3372885.3373824.

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174, 2023.

Yihan Wang, Jatin Chauhan, Wei Wang, and Cho-Jui Hsieh. Universality and limitations of prompt tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Yotam Wolf, Noam Wies, Oshri Avnery, Yoav Levine, and Amnon Shashua. Fundamental limitations of alignment in large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53079–53112, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*, 2018.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Is in-context learning sufficient for instruction following in LLMs? In *International Conference on Learning Representations*, 2025.

A FOUNDATIONS

Notation at a glance. θ, θ': base and aligned model. $\pi_\theta, \pi_{\theta'}$: their next-token policies, $\pi_{\theta,S}$: the base model's policy with context S in front (the induced policy). π is never used for anything else. s: state $(x, y_{<t})$, v: next token. V: vocabulary size. S: context. $\delta_{\theta'}(s, v)$: the update (shift of log-probabilities from π_θ to $\pi_{\theta'}$). $\delta_S(s, v)$: shift a context realizes. osc_s: spread over tokens at state s, maximum minus minimum. E_n^{KV}: key–value prefix frontier, the least error at length n. $B^\star$: canonical operator of a bilinear decomposition. H_n: rank-n frontier in shift space, $r_{\mathrm{eff}} = \min\{n : H_n = 0\}$. $u_j = a_j \mathbf{w}_j$: clause j's declared effect, scope times token preference. η_j: interpreter defect of clause j. ι: interaction between selected clauses. γ, κ: query richness and readout visibility. μ: dictionary coherence.

Further notation. Logarithms are natural. $\|\cdot\|_{\mathrm{op}}$ is the Euclidean operator norm of a matrix, its largest singular value, and σ_j is the j-th singular value in nonincreasing order. The approximation error $d(\cdot)$ always carries an argument (a context, a prefix, or a retriever). A bare d is the head dimension in Appendix C and the final hidden width in Appendix E. Here ε alone is a generic tolerance. We distinguish throughout between "for every task there is a context" and "there is one context serving every task". Throughout this appendix, p and q denote probability distributions on a finite set Ω (tokens, or whole responses), written as functions $p : \Omega \to [0, 1]$ with $\sum_{\omega\in\Omega} p(\omega) = 1$, and $p(A) = \sum_{\omega\in A} p(\omega)$ is the probability of a subset $A \subseteq \Omega$.

Total variation: two forms, and monotonicity under maps. Every error in the paper is a total variation distance, and two different formulas for it are used: the sum form, which is what the proofs compute with, and the set form, which is what gives total variation its meaning and yields the two facts used in lower bounds (a single point's discrepancy is a lower bound, and reporting less about the outcome cannot increase the distance). For two distributions p, q on a finite set Ω, the total variation distance is defined and can be rewritten as

$$\mathrm{TV}(p, q) = \tfrac{1}{2} \sum_{\omega\in\Omega} |p(\omega) - q(\omega)| = \max_{A\subseteq\Omega} \big(p(A) - q(A)\big).$$

In words: the set form says that $\mathrm{TV}(p, q)$ is the largest amount by which the two distributions can disagree about the probability of any single event A. The sum form says it is half the total absolute discrepancy across all outcomes, equivalently the amount of probability mass that has to be moved to turn p into q. The two forms agree for the following reason. Write $a(\omega) = p(\omega) - q(\omega)$, so that $\sum_\omega a(\omega) = 1 - 1 = 0$: the positive entries of a and the negative entries of a have the same total size, call it τ, and $\sum_\omega |a(\omega)| = 2\tau$, so the sum form equals τ. For any set A, $p(A) - q(A) = \sum_{\omega\in A} a(\omega)$ is largest when A collects every positive entry of a and no negative one, that is, at $A^\star = \{\omega : p(\omega) > q(\omega)\}$, where it equals τ. So the set form also equals τ. Two consequences are used in the paper. First, since $\{\omega\}$ and its complement are admissible sets, $\mathrm{TV}(p, q) \geq |p(\omega) - q(\omega)|$ for every single point ω: any one outcome on which the two distributions disagree already forces that much distance (used in Lemma 12). Second, let $h : \Omega \to \Omega'$ be a deterministic map (in the paper, the map $y \mapsto y_1$ that reports the first token of a response), and let $p \circ h^{-1}$ and $q \circ h^{-1}$ be the image distributions on Ω', so that $(p \circ h^{-1})(A') = p(h^{-1}(A'))$ for $A' \subseteq \Omega'$. In words, these are the distributions of $h(\omega)$ when ω is drawn from p or from q. Every event of the form $h^{-1}(A')$ is a subset of Ω, so the maximum in the set form of $\mathrm{TV}(p \circ h^{-1}, q \circ h^{-1})$ runs over a subfamily of the sets that the maximum in $\mathrm{TV}(p, q)$ runs over, and a maximum over fewer candidates is no larger. Therefore

$$\mathrm{TV}\left(p \circ h^{-1}, q \circ h^{-1}\right) \leq \mathrm{TV}(p, q) :$$

total variation cannot increase under a deterministic map of the outcome. In words: if two models are close on whole responses they are close on any summary of the response, and, read the other way, disagreement on a summary such as the first token implies at least that much disagreement on whole responses. This is what the first-token statement of Lemma 12 uses. Total variation is symmetric and satisfies the triangle inequality, since it is half the ℓ_1 distance between p and q viewed as vectors in $\mathbb{R}^{|\Omega|}$. The triangle inequality is what allows a distance to be bounded by going through intermediate distributions, as in Lemmas 10 and 12 and the restricted form of Proposition 1.

Total variation controls bounded reward. The paper measures error in total variation, but what one usually cares about is whether a compiled model obtains the same reward as the target. This

paragraph shows that the first controls the second, which is the justification for the choice of distance. Let P, Q be two distributions on a finite response space $\mathcal{Y}$ and let $r : \mathcal{Y} \to [0,1]$ be a reward. Then $|\mathbb{E}_P r - \mathbb{E}_Q r| \le \mathrm{TV}(P,Q)$, where $\mathbb{E}_P r = \sum_y P(y) r(y)$. To see this, write $a(y) = P(y) - Q(y)$, so that $\mathbb{E}_P r - \mathbb{E}_Q r = \sum_y a(y) r(y)$. By the previous paragraph the positive entries of a sum to $\mathrm{TV}(P,Q)$ and the negative entries to $-\mathrm{TV}(P,Q)$. Multiplying each entry by a number $r(y) \in [0,1]$ can only shrink it toward zero, so the sum of the positive terms of $a(y)r(y)$ lies in $[0, \mathrm{TV}(P,Q)]$ and the sum of the negative terms in $[-\mathrm{TV}(P,Q), 0]$. Their total therefore lies in $[-\mathrm{TV}(P,Q), \mathrm{TV}(P,Q)]$. In words: the reward can differ between the two models only on the probability mass that the two distributions place differently, and that mass is exactly the total variation. Applying this at each task x with $P = \pi_{\theta,S}(\cdot \mid x)$ and $Q = \pi_{\theta'}(\cdot \mid x)$, and averaging over $x \sim D$ (the absolute value of an average is at most the average of the absolute values), gives $|\mathbb{E}_{x\sim D}[\mathbb{E}_{\pi_{\theta,S}} r - \mathbb{E}_{\pi_{\theta'}} r]| \le \mathbb{E}_{x\sim D} \mathrm{TV}(\pi_{\theta,S}(\cdot \mid x), \pi_{\theta'}(\cdot \mid x)) = d(S)$, since $d(S)$ is exactly that average. For a fixed task the bound is tight for the reward that is the indicator of the maximizing event $A^\star$ of the set form; averaged over tasks with one task-independent reward it need not be. The converse fails: matching one reward does not identify a policy, since two very different distributions can have the same mean reward, which is why the paper works with the distance and not with a reward gap.

Tilts. Fix a state s. Let $\mathbf{z} \in \mathbb{R}^V$ be the base model's logit vector at s, so that its next-token distribution is $p_0 = \mathrm{softmax}(\mathbf{z})$, $p_0(v) = e^{\mathbf{z}(v)} / \sum_u e^{\mathbf{z}(u)}$, a strictly positive distribution on the V tokens. For a vector $\delta \in \mathbb{R}^V$ (a shift, indexed by the token), the *tilt* of p_0 by δ is the distribution obtained by multiplying each probability by $e^{\delta(v)}$ and renormalizing. It coincides with the softmax of the shifted logits:

$$\mathrm{softmax}(\mathbf{z} + \delta)(v) = \frac{e^{\mathbf{z}(v)+\delta(v)}}{\sum_u e^{\mathbf{z}(u)+\delta(u)}} = \frac{p_0(v) e^{\delta(v)}}{\sum_u p_0(u) e^{\delta(u)}},$$

because $e^{\mathbf{z}(v)} = p_0(v) \cdot \sum_u e^{\mathbf{z}(u)}$, and the common factor $\sum_u e^{\mathbf{z}(u)}$ appears in the numerator and in every term of the denominator and cancels. In words: adding δ to the logits and multiplying the probabilities by e^{δ} are the same operation. Three facts follow. (1) Every strictly positive distribution p on the tokens is the tilt of p_0 by the shift $\delta(v) = \log\big(p(v)/p_0(v)\big)$: substituting, the numerator becomes $p(v)$ and the denominator $\sum_u p(u) = 1$. So "a shift" is not a restriction on what a model can do. Any positive distribution is some shift of the base. (2) A tilt is unchanged when a constant c is added to every entry of δ, because the factor e^c appears in the numerator and in every term of the denominator and cancels. So a shift is meaningful only up to an additive constant that does not depend on the token. This is why the size of a shift is measured by its oscillation, which ignores such constants, and not by a norm, which does not. (3) Applying (1) with $p_0 = \pi_\theta(\cdot \mid s)$ to the aligned distribution $p = \pi_{\theta'}(\cdot \mid s)$ and to the induced distribution $p = \pi_{\theta,S}(\cdot \mid s)$, the shifts obtained are exactly $\delta_{\theta'}(s,\cdot)$ and $\delta_S(s,\cdot)$ of (3.2), so

$$\pi_{\theta'}(v \mid s) = \frac{p_0(v) e^{\delta_{\theta'}(s,v)}}{\sum_u p_0(u) e^{\delta_{\theta'}(s,u)}}, \qquad \pi_{\theta,S}(v \mid s) = \frac{p_0(v) e^{\delta_S(s,v)}}{\sum_u p_0(u) e^{\delta_S(s,u)}},$$

where in fact both denominators equal one. This is (3.2) rearranged, not a modeling step: $\delta_{\theta'}$ and δ_S are defined as log-ratios against the same base distribution, so both models are tilts of that distribution, and Lemma 9 compares the two tilts.

The oscillation seminorm. Let $f : \{1, \dots, V\} \to \mathbb{R}$ be a function of the token at a fixed state, equivalently a vector $f \in \mathbb{R}^V$. The paper writes $\mathrm{osc}_s f$ when $f = f(s,\cdot)$ comes from a function of state and token, and $\mathrm{osc}(\mathbf{w})$ for a bare vector $\mathbf{w} \in \mathbb{R}^V$, but the functional is the same:

$$\mathrm{osc}\, f = \max_v f(v) - \min_v f(v) \ \ge 0.$$

In words: the oscillation is the spread of the entries, the distance between the largest and the smallest. Four properties make it a seminorm. *Subadditivity*, $\mathrm{osc}(f+g) \le \mathrm{osc}\, f + \mathrm{osc}\, g$: because $\max_v (f+g)(v) \le \max_v f(v) + \max_v g(v)$ (the maximum of a sum is at most the sum of the maxima, since the token that maximizes $f+g$ cannot beat the separate maximizers of f and of g) and $\min_v (f+g)(v) \ge \min_v f(v) + \min_v g(v)$ (likewise for minima), subtracting the second inequality from the first gives the claim. *Homogeneity*, $\mathrm{osc}(cf) = |c| \,\mathrm{osc}\, f$ for a scalar c: for $c \ge 0$

the maximum and minimum scale by c. For $c < 0$ they scale by $|c|$ and exchange roles, which leaves the difference unchanged. In particular $\operatorname{osc}(-f) = \operatorname{osc} f$. *Null space*: $\operatorname{osc} f = 0$ if and only if $\max f = \min f$, that is, if and only if f is a constant vector $c\mathbf{1}$ with $\mathbf{1}$ the all-ones vector. *Invariance*: $\operatorname{osc}(f + c\mathbf{1}) = \operatorname{osc} f$, since adding c to every entry adds c to both the maximum and the minimum. So osc is a seminorm on $\mathbb{R}^V$ whose null space is the line $\mathbb{R}\mathbf{1}$ of constant vectors, equivalently a norm on the quotient space $\mathbb{R}^V/\mathbb{R}\mathbf{1}$. By fact (2) of the previous paragraph, that quotient is exactly the space a softmax sees. In words: the oscillation is zero on precisely the shifts that do nothing, and it treats two shifts that differ by a constant as the same shift, which is the right behavior for a measure of what a shift does to a softmax. A reverse form of subadditivity is also used: applying subadditivity to $f = (f + g) + (-g)$ gives $\operatorname{osc} f \le \operatorname{osc}(f + g) + \operatorname{osc} g$, hence $\operatorname{osc}(f + g) \ge \operatorname{osc} f - \operatorname{osc} g$. In words, adding a small perturbation cannot reduce a spread by more than the perturbation's own spread, which is how lower bounds survive the presence of a residual term.

The oscillation is the natural size of a shift in the sup-norm, in the following exact sense. The distance from f to the constant vectors is

$$\min_{c \in \mathbb{R}} \max_{v} |f(v) - c| = \tfrac{1}{2} \operatorname{osc} f,$$

attained at the midrange $c^\star = \frac{1}{2}(\max_v f(v) + \min_v f(v))$. Indeed, at $c^\star$ the largest deviation is $\max f - c^\star = c^\star - \min f = \frac{1}{2} \operatorname{osc} f$. For any other c, either $c < c^\star$, in which case $\max f - c > \frac{1}{2} \operatorname{osc} f$, or $c > c^\star$, in which case $c - \min f > \frac{1}{2} \operatorname{osc} f$. In words: after the best possible choice of the irrelevant constant, the largest entry of a shift has absolute value exactly half its oscillation, so a shift of oscillation Δ is one that moves some logit by $\Delta/2$ relative to the rest. Finally, the oscillation is dominated by the Euclidean norm of any centered version of f: for every constant c,

$$\operatorname{osc} f = \operatorname{osc}(f - c\mathbf{1}) \le 2 \max_v |f(v) - c| \le 2\|f - c\mathbf{1}\|_2,$$

the first step by invariance, the second because the maximum and the minimum of $f - c\mathbf{1}$ each have absolute value at most $\max_v |f(v) - c|$ so their difference is at most twice that, and the third because the largest absolute entry of a vector is at most its Euclidean norm (one square is at most the sum of all the squares). Taking c to be the mean of the entries of f gives a sufficient bound on the oscillation in terms of the centered Euclidean residual. The reverse direction fails by a factor that grows with the vocabulary, which is why the Euclidean norm is only ever used to certify smallness.

B PROOFS FOR THE BRIDGE

This appendix proves the five results of Section 3: Lemma 9 (contraction), about a single next-token distribution. Lemma 10 (rollout) passes from one token to a response of T tokens, one term per position. Proposition 1 (bridge) combines them and averages over tasks, and Proposition 11 replaces the factor T by $\sqrt{T}$. Lemma 12 (converse) shows that a large residual at the first token forces a large error. The distance throughout is total variation, $\mathrm{TV}(p, q) = \frac{1}{2} \sum_v |p(v) - q(v)|$, the largest amount by which two distributions can disagree on the probability of any event (Appendix A). Two conventions are used. Within a proof at one fixed state s, a shift such as $\delta_{\theta'}(s, \cdot)$ is a vector in $\mathbb{R}^V$ indexed by the token, and we write $\delta(v)$ for its entry, suppressing the state. For a distribution p over tokens and a function f of the token, $\mathbb{E}_p f = \sum_v p(v) f(v)$.

B.1 LEMMA 9 (CONTRACTION)

Lemma 9 (contraction)**.** *For any logits* $\mathbf{z}$ *and shifts* δ, δ' *at a state* s*,*

$$\mathrm{TV}\left(\operatorname{softmax}(\mathbf{z} + \delta), \operatorname{softmax}(\mathbf{z} + \delta')\right) \le \tanh\Big(\tfrac{1}{4} \operatorname{osc}_s(\delta - \delta')\Big),$$

writing $\Delta = \operatorname{osc}_s(\delta - \delta')$*, with equality at* $\Delta = 0$ *and, for* $\Delta > 0$ *and every* $V \ge 2$*, exactly when* $\delta - \delta'$ *takes the two values* $\pm\frac{1}{2}\Delta$ *about its midrange and* $\operatorname{softmax}(\mathbf{z} + \delta')$ *gives the larger-valued tokens total probability* $1/(e^{\Delta/2} + 1)$*.*

↑ Back to page 3

The objects. Fix a state s and let V be the size of the vocabulary. Three vectors in $\mathbb{R}^V$ are given, each indexed by the token $v \in \{1, \dots, V\}$:

- the logits $\mathbf{z} = (\mathbf{z}(1), \dots, \mathbf{z}(V))$. In the paper these are the base model's logits at s, but the lemma holds for any real vector.
- two shifts $\delta = (\delta(1), \dots, \delta(V))$ and $\delta' = (\delta'(1), \dots, \delta'(V))$. In the paper these are the induced shift $\delta_S(s, \cdot)$ of a context and the update $\delta_{\theta'}(s, \cdot)$ of (3.2), but again the lemma holds for any two real vectors.

Adding a shift to the logits and applying the softmax produces a probability distribution over the V tokens. The two distributions being compared are

$$\begin{aligned} q &= \mathrm{softmax}(\mathbf{z} + \delta), & q(v) &= \frac{e^{\mathbf{z}(v)+\delta(v)}}{\sum_{u=1}^{V} e^{\mathbf{z}(u)+\delta(u)}}, \\ p &= \mathrm{softmax}(\mathbf{z} + \delta'), & p(v) &= \frac{e^{\mathbf{z}(v)+\delta'(v)}}{\sum_{u=1}^{V} e^{\mathbf{z}(u)+\delta'(u)}}. \end{aligned} \tag{B.1}$$

Both are strictly positive at every token, because every exponential is positive. Their distance is the total variation

$$\mathrm{TV}(q, p) = \tfrac{1}{2} \sum_{v=1}^{V} |q(v) - p(v)|,$$

and the quantity that will control this distance is the oscillation of the difference of the two shifts,

$$\Delta := \mathrm{osc}_s(\delta - \delta') = \max_v \big(\delta(v) - \delta'(v)\big) - \min_v \big(\delta(v) - \delta'(v)\big) \ \geq 0.$$

The lemma claims that $\mathrm{TV}(q, p) \leq \tanh(\Delta/4)$, and the proof below also determines exactly when equality holds.

Idea of the proof. A softmax does not change when the same constant is added to every logit. So the only thing that can separate q from p is how unevenly the difference $\delta - \delta'$ is spread across tokens, and Δ measures exactly that spread. The proof makes this precise in three moves. It first writes q as a reweighting of p by a likelihood ratio that is confined to an interval whose width is set by Δ. It then observes that the total variation between q and p is the mean absolute deviation of that ratio about its mean. Finally, it bounds a mean absolute deviation on an interval by an elementary convexity argument and maximizes the resulting expression, which produces the hyperbolic tangent.

Proof of Lemma 9. Step 0: the degenerate case. Suppose $\Delta = 0$. Then $\delta(v) - \delta'(v)$ has the same value at every token, say c, so $\mathbf{z} + \delta = (\mathbf{z} + \delta') + c\mathbf{1}$, where $\mathbf{1}$ is the all-ones vector. A softmax is invariant to adding a constant to every coordinate: in (B.1), the factor e^c appears in the numerator and in every term of the denominator and cancels. Hence $q = p$, $\mathrm{TV}(q, p) = 0$ and $\tanh(0) = 0$, so the claim holds with equality. From now on assume $\Delta > 0$.

Step 1: q is a reweighting of p. Let $N = \sum_u e^{\mathbf{z}(u)+\delta(u)}$ and $N' = \sum_u e^{\mathbf{z}(u)+\delta'(u)}$ be the two normalizing constants in (B.1). For every token v,

$$p(v)\, e^{\delta(v)-\delta'(v)} = \frac{e^{\mathbf{z}(v)+\delta'(v)}}{N'}\, e^{\delta(v)-\delta'(v)} = \frac{e^{\mathbf{z}(v)+\delta(v)}}{N'}.$$

Summing this identity over v gives $\sum_v p(v) e^{\delta(v)-\delta'(v)} = N/N'$. Dividing the identity by its sum,

$$q(v) = \frac{e^{\mathbf{z}(v)+\delta(v)}}{N} = \frac{e^{\mathbf{z}(v)+\delta(v)}/N'}{N/N'} = \frac{p(v)\, e^{\delta(v)-\delta'(v)}}{\sum_u p(u)\, e^{\delta(u)-\delta'(u)}}. \tag{B.2}$$

In words: q is obtained from p by multiplying each probability by the factor $e^{\delta(v)-\delta'(v)}$ and renormalizing. The logits $\mathbf{z}$ have disappeared. From here on only p and the difference $\delta - \delta'$ matter.

Step 2: confining the reweighting factor to an interval. Write $g(v) = \delta(v) - \delta'(v)$ for the difference of the shifts, and let $g_{\min} = \min_v g(v)$ and $g_{\max} = \max_v g(v)$, so that $\Delta = g_{\max} - g_{\min}$. Define

$$Z(v) = e^{\, g(v) - g_{\min}} \qquad \text{and} \qquad \omega = e^{\Delta} > 1.$$

Since $0 \le g(v) - g_{\min} \le \Delta$ for every token, $1 \le Z(v) \le \omega$. Moreover $Z(v) = 1$ at any token where g attains its minimum and $Z(v) = \omega$ at any token where g attains its maximum. Substituting $e^{g(v)} = e^{g_{\min}} Z(v)$ into (B.2), the constant factor $e^{g_{\min}}$ appears in the numerator and in every term of the denominator and cancels, leaving

$$q(v) = \frac{p(v)\, Z(v)}{\bar{Z}}, \qquad \bar{Z} := \sum_u p(u) Z(u) = \mathbb{E}_p Z. \tag{B.3}$$

It is convenient to think of Z as a random variable: draw a token v from p and report $Z(v)$. Its mean is $\bar{Z}$. Because $\bar{Z}$ is a weighted average, with positive weights $p(u)$, of numbers in $[1, \omega]$, and because Z takes both endpoint values 1 and ω on tokens of positive probability, the mean lies strictly inside the interval: $1 < \bar{Z} < \omega$.

Step 3: the total variation is a mean absolute deviation. By (B.3), $q(v) - p(v) = p(v)\big(Z(v)/\bar{Z} - 1\big) = p(v)\big(Z(v) - \bar{Z}\big)/\bar{Z}$ for every token. Taking absolute values, summing over tokens and halving,

$$\mathrm{TV}(q, p) = \tfrac{1}{2} \sum_v p(v)\, \frac{|Z(v) - \bar{Z}|}{\bar{Z}} = \frac{\mathbb{E}_p |Z - \bar{Z}|}{2\bar{Z}}. \tag{B.4}$$

This is an identity, not a bound: the distance between the two distributions is the mean absolute deviation of Z about its mean, divided by twice that mean. Everything that follows bounds the right-hand side of (B.4) using only the two facts that Z lives in $[1, \omega]$ and has mean $\bar{Z}$.

Step 4: the chord bound. Consider the function $\phi(z) = |z - \bar{Z}|$ on the interval $[1, \omega]$. It is convex, being the absolute value of an affine function, and its values at the two endpoints are

$$\phi(1) = \bar{Z} - 1 \qquad \text{and} \qquad \phi(\omega) = \omega - \bar{Z},$$

both positive because $1 < \bar{Z} < \omega$. A convex function lies below the chord that joins the two endpoints of its graph. Explicitly, any $z \in [1, \omega]$ can be written as the convex combination $z = (1 - \lambda) \cdot 1 + \lambda \cdot \omega$ with $\lambda = (z - 1)/(\omega - 1) \in [0, 1]$, and convexity gives

$$\phi(z) \le (1 - \lambda)\phi(1) + \lambda \phi(\omega) = \phi(1) + \frac{z - 1}{\omega - 1}\big(\phi(\omega) - \phi(1)\big) =: \ell(z),$$

where ℓ is the affine function through the points $(1, \phi(1))$ and $(\omega, \phi(\omega))$. Apply this at $z = Z(v)$ for each token v, multiply by $p(v) \ge 0$ and sum over v:

$$\mathbb{E}_p |Z - \bar{Z}| = \mathbb{E}_p \phi(Z) \le \mathbb{E}_p \ell(Z).$$

Because ℓ is affine, its expectation equals its value at the mean: $\mathbb{E}_p \ell(Z) = \ell(\mathbb{E}_p Z) = \ell(\bar{Z})$. This is the one place where the mean enters, and it is what ties the bound to $\bar{Z}$. Evaluating ℓ at $\bar{Z}$,

$$\begin{aligned} \ell(\bar{Z}) &= (\bar{Z} - 1) + \frac{\bar{Z} - 1}{\omega - 1}\Big((\omega - \bar{Z}) - (\bar{Z} - 1)\Big) \\ &= (\bar{Z} - 1)\, \frac{(\omega - 1) + (\omega + 1 - 2\bar{Z})}{\omega - 1} = \frac{2(\bar{Z} - 1)(\omega - \bar{Z})}{\omega - 1}. \end{aligned}$$

Combining with (B.4),

$$\mathrm{TV}(q, p) \le \frac{(\bar{Z} - 1)(\omega - \bar{Z})}{\bar{Z}\,(\omega - 1)}. \tag{B.5}$$

Step 5: maximizing over the mean. The right-hand side of (B.5) depends on the distribution p only through the single number $\bar{Z} \in (1, \omega)$, so it remains to find its largest possible value as $\bar{Z}$ ranges over that interval. Expanding the numerator, $(\bar{Z} - 1)(\omega - \bar{Z}) = \omega \bar{Z} - \bar{Z}^2 - \omega + \bar{Z}$, and dividing each term by $\bar{Z}$,

$$\frac{(\bar{Z} - 1)(\omega - \bar{Z})}{\bar{Z}} = \omega + 1 - \Big(\bar{Z} + \frac{\omega}{\bar{Z}}\Big), \qquad \text{so} \qquad \mathrm{TV}(q, p) \le \frac{\omega + 1 - \big(\bar{Z} + \omega/\bar{Z}\big)}{\omega - 1}.$$

The arithmetic–geometric mean inequality gives $\bar{Z} + \omega/\bar{Z} \ge 2\sqrt{\omega}$. In this two-term case it is simply the expansion of a square,

$$\bar{Z} + \frac{\omega}{\bar{Z}} - 2\sqrt{\omega} = \Big(\sqrt{\bar{Z}} - \sqrt{\omega/\bar{Z}}\Big)^2 \ge 0,$$

with equality if and only if $\bar{Z} = \sqrt{\omega}$, which is a point of the interval $(1, \omega)$. Subtracting a larger quantity in the numerator makes the fraction smaller, and the denominator $\omega - 1$ is positive, so

$$\mathrm{TV}(q, p) \le \frac{\omega + 1 - 2\sqrt{\omega}}{\omega - 1}.$$

Step 6: the hyperbolic tangent. Factor numerator and denominator: $\omega + 1 - 2\sqrt{\omega} = (\sqrt{\omega} - 1)^2$ and $\omega - 1 = (\sqrt{\omega} - 1)(\sqrt{\omega} + 1)$. Cancelling the common factor $\sqrt{\omega} - 1 > 0$,

$$\mathrm{TV}(q, p) \le \frac{\sqrt{\omega} - 1}{\sqrt{\omega} + 1} = \frac{e^{\Delta/2} - 1}{e^{\Delta/2} + 1},$$

using $\sqrt{\omega} = e^{\Delta/2}$. Finally, multiplying the numerator and denominator of $\tanh u = (e^u - e^{-u})/(e^u + e^{-u})$ by e^u gives $\tanh u = (e^{2u} - 1)/(e^{2u} + 1)$, and with $u = \Delta/4$ this is $(e^{\Delta/2} - 1)/(e^{\Delta/2} + 1)$. Hence $\mathrm{TV}(q, p) \le \tanh(\Delta/4)$, which is the inequality of the lemma.

Equality. The proof used exactly two inequalities, the chord bound of Step 4 and the arithmetic–geometric mean inequality of Step 5, so equality holds if and only if both are tight.

The chord bound is tight if and only if $\phi(Z(v)) = \ell(Z(v))$ for every token v, since every token has positive probability under p. Now ϕ and ℓ agree at the endpoints $z = 1$ and $z = \omega$, but they differ strictly everywhere in between. To see this, note that ϕ is affine on $[1, \bar{Z}]$ and on $[\bar{Z}, \omega]$ separately, with a kink at $\bar{Z}$ where $\phi(\bar{Z}) = 0$, whereas $\ell(\bar{Z}) = 2(\bar{Z} - 1)(\omega - \bar{Z})/(\omega - 1) > 0$. On $[1, \bar{Z}]$ the difference $\ell - \phi$ is affine, zero at $z = 1$ and positive at $z = \bar{Z}$, hence positive on $(1, \bar{Z}]$. On $[\bar{Z}, \omega]$ it is affine, positive at $z = \bar{Z}$ and zero at $z = \omega$, hence positive on $[\bar{Z}, \omega)$. So tightness forces $Z(v) \in \{1, \omega\}$ for every token, that is, $g(v) \in \{g_{\min}, g_{\max}\}$: *the difference $\delta - \delta'$ takes exactly two values across the vocabulary.* Write $H = \{v : g(v) = g_{\max}\}$ for the tokens receiving the larger shift and $L = \{v : g(v) = g_{\min}\}$ for the others. Both sets are nonempty and together they exhaust the vocabulary.

The arithmetic–geometric mean inequality is tight if and only if $\bar{Z} = \sqrt{\omega}$. When Z is two-valued, $\bar{Z} = \omega\, p(H) + 1 \cdot p(L) = 1 + (\omega - 1)p(H)$, where $p(H) = \sum_{v \in H} p(v)$ is the total probability of the high group. Setting this equal to $\sqrt{\omega}$ and solving,

$$p(H) = \frac{\sqrt{\omega} - 1}{\omega - 1} = \frac{1}{\sqrt{\omega} + 1} = \frac{1}{e^{\Delta/2} + 1}, \qquad p(L) = 1 - p(H) = \frac{e^{\Delta/2}}{e^{\Delta/2} + 1}.$$

Conversely, if $\delta - \delta'$ is two-valued and $p(H) = 1/(e^{\Delta/2} + 1)$, then both inequalities are equalities and $\mathrm{TV}(q, p) = \tanh(\Delta/4)$ exactly.

In summary, for $\Delta > 0$ equality holds if and only if (i) $\delta - \delta'$ takes only two values across the vocabulary, and (ii) the distribution $p = \mathrm{softmax}(\mathbf{z} + \delta')$ gives the tokens at the larger value total probability $1/(e^{\Delta/2} + 1)$. Subtracting the midrange $\frac{1}{2}(g_{\max} + g_{\min})$ from $\delta - \delta'$, which does not change either softmax, the two values become $\pm\Delta/2$, the form quoted in the statement of the lemma. Nothing constrains how many tokens sit in each group, only the probability the group carries, so the bound is attained for every vocabulary size $V \ge 2$: at $V = 2$ each group is a single token, and for $V > 2$ a group may contain many tokens. For instance, $V = 4$ with $\delta - \delta' = (0, 0, \Delta, \Delta)$ and p giving the last two tokens total probability $1/(e^{\Delta/2} + 1)$, split between them in any way, attains the bound. Because total variation is symmetric, the condition can equally be stated for q: by (B.3), $q(H) = p(H)\,\omega/\bar{Z} = \omega/\big((\sqrt{\omega} + 1)\sqrt{\omega}\big) = e^{\Delta/2}/(e^{\Delta/2} + 1)$, which is the same condition with the roles of the two distributions exchanged.

The quantity $\mathrm{osc}_s(\delta - \delta')$ is the Hilbert projective distance between the two tilted distributions, and the bound is the sharp comparison between that distance and total variation given by Cohen & Fausti (2023), itself in the lineage of Birkhoff's contraction coefficient (Birkhoff, 1957). ■

Machine check. This lemma and its equality characterization are formalized in Lean 4, as are most but not all of the endpoints proved in this appendix: the restricted-vocabulary form of Proposition 1 is not.

Comparison with a Euclidean bound. The paper measures the size of a shift by its oscillation rather than by the Euclidean norm. The Euclidean norm also bounds total variation, but worse: the bound grows with the number of tokens a shift touches, and it is not invariant to the additive constants a softmax ignores.

The objects. Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^V$ be two logit vectors, which in the lemma's setting are $\mathbf{u} = \mathbf{z} + \delta'$ and $\mathbf{v} = \mathbf{z} + \delta$. Let $A(\mathbf{x}) = \log \sum_i e^{\mathbf{x}(i)}$ be the log-sum-exp, with $\nabla A(\mathbf{x}) = \mathrm{softmax}(\mathbf{x})$ and $\nabla^2 A(\mathbf{x}) = \mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top$ for $\mathbf{p} = \mathrm{softmax}(\mathbf{x})$, obtained by differentiating the gradient once more.

(a) The Hessian's quadratic form is a variance. For any $\mathbf{w} \in \mathbb{R}^V$,

$$\mathbf{w}^\top \big(\mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top \big) \mathbf{w} = \sum_i p_i w_i^2 - \Big(\sum_i p_i w_i \Big)^2 = \mathrm{Var}_{I \sim \mathbf{p}}(w_I),$$

the mean of the square minus the square of the mean. In words: the curvature of the log-normalizer in a direction $\mathbf{w}$ is how much $\mathbf{w}$ varies across tokens, weighted by their probabilities.

(b) A variance is at most a quarter of the squared range. Popoviciu's inequality gives $\mathrm{Var}(w_I) \le (\max_i w_i - \min_i w_i)^2/4$. Since $(a-b)^2 \le 2(a^2+b^2)$ and the maximizing and minimizing coordinates are distinct when they differ, $(\max_i w_i - \min_i w_i)^2 \le 2\|\mathbf{w}\|_2^2$, so the quadratic form is at most $\frac{1}{2}\|\mathbf{w}\|_2^2$ everywhere.

(c) Divergence is curvature, and Pinsker converts it. The divergence between two softmaxes is the Bregman divergence of A,

$$\mathrm{KL}\big(\mathrm{softmax}(\mathbf{u}) \,\|\, \mathrm{softmax}(\mathbf{v}) \big) = A(\mathbf{v}) - A(\mathbf{u}) - \nabla A(\mathbf{u})^\top (\mathbf{v} - \mathbf{u}),$$

checked by writing $\log \mathrm{softmax}(\mathbf{x})(i) = \mathbf{x}(i) - A(\mathbf{x})$ and using $\sum_i \mathrm{softmax}(\mathbf{u})(i) = 1$. Taylor's theorem with the Lagrange remainder, applied to $t \mapsto A(\mathbf{u} + t(\mathbf{v} - \mathbf{u}))$ on $[0, 1]$, makes this $\frac{1}{2}(\mathbf{v} - \mathbf{u})^\top \nabla^2 A(\boldsymbol{\xi})(\mathbf{v} - \mathbf{u})$ for some $\boldsymbol{\xi}$ on the segment, hence at most $\frac{1}{4}\|\mathbf{u} - \mathbf{v}\|_2^2$ by (b). Pinsker's inequality then gives $\mathrm{TV}(\mathrm{softmax}(\mathbf{u}), \mathrm{softmax}(\mathbf{v})) \le \|\mathbf{u} - \mathbf{v}\|_2/(2\sqrt{2})$.

Why the oscillation bound is better. Take a shift difference of $+\varepsilon$ on half the tokens and $-\varepsilon$ on the other half. Its Euclidean norm is $\varepsilon\sqrt{V}$, so the Euclidean bound reads $\varepsilon\sqrt{V/8}$ and grows without limit with the vocabulary, although the two distributions are not moving further apart in any meaningful sense. Its oscillation is 2ε, so the contraction bound is $\tanh(\varepsilon/2) \le \varepsilon/2$ regardless of V. The rank law of Appendix C produces exactly such spread-out shifts. The second defect is that the Euclidean norm changes when a constant is added to every coordinate although the softmax does not, so it is meaningful only after centering. The oscillation is invariant automatically.

B.2 Lemma 10 (rollout)

Lemma 10 (rollout)**.** *For any task x and context S, with $s_t = (x, y_{<t})$,*

$$\mathrm{TV}\left(\pi_{\theta,S}(y \mid x), \pi_{\theta'}(y \mid x)\right) \le \sum_{t=1}^{T} \mathbb{E}_{y_{<t} \sim \pi_{\theta'}} \mathrm{TV}\left(\pi_{\theta,S}(\cdot \mid s_t), \pi_{\theta'}(\cdot \mid s_t)\right).$$

↑ Back to page 3

The objects. Fix a task x and a context S. A response is a sequence $y = (y_1, \ldots, y_T)$ of T tokens, and $s_t = (x, y_{<t})$ denotes the state before the t-th token is generated, where $y_{<t} = (y_1, \ldots, y_{t-1})$. For $t = 1$ the prefix $y_{<1}$ is empty and $s_1 = (x, \emptyset)$. Two next-token policies are given at every state: the induced policy $\pi_{\theta,S}(\cdot \mid s)$ of the base model with the context in front, and the aligned policy $\pi_{\theta'}(\cdot \mid s)$. Each defines a distribution over whole responses by multiplying its next-token probabilities along the response:

$$P(y) := \pi_{\theta,S}(y \mid x) = \prod_{t=1}^{T} \pi_{\theta,S}(y_t \mid s_t), \qquad Q(y) := \pi_{\theta'}(y \mid x) = \prod_{t=1}^{T} \pi_{\theta'}(y_t \mid s_t).$$

In words: to sample a response from P, generate the tokens one at a time, each from the induced policy at the current state. The distribution Q does the same with the aligned policy. The lemma

bounds $\mathrm{TV}(P, Q)$, the total variation between the two response distributions, by a sum over positions t of the expected total variation between the two next-token distributions at s_t, where the expectation is over prefixes $y_{<t}$ drawn from the aligned model. Total variation between response distributions is the right quantity because, by Appendix A, it bounds the difference in expected value of any reward with values in $[0, 1]$.

Idea of the proof. The two response distributions differ at every one of the T steps, which makes them hard to compare directly. The trick is to interpolate: build a chain of $T+1$ "hybrid" distributions, the t-th of which uses the aligned policy for the first t steps and the induced policy for the rest. The first hybrid is P and the last is Q, and consecutive hybrids differ in the policy used at exactly one step. The triangle inequality reduces the distance from P to Q to a sum of T distances between consecutive hybrids, and each of those is computed exactly: it is the expected one-step distance at the step where the two hybrids differ. This is the compounding-error argument of behavior cloning (Ross & Bagnell, 2010), with the roles of the two policies fixed so that the expectation is under the aligned model.

Proof of Lemma 10. Step 1: a chain of hybrid distributions. For $t = 0, 1, \ldots, T$ define the hybrid response distribution that generates the first t tokens with the aligned model and the remaining $T - t$ tokens with the induced policy:

$$H_t(y) = \prod_{i=1}^{t} \pi_{\theta'}(y_i \mid s_i) \prod_{i=t+1}^{T} \pi_{\theta,S}(y_i \mid s_i).$$

Each H_t is a probability distribution over responses: its values are nonnegative, and they sum to one, as one sees by summing over y_T first (the last factor is a next-token distribution, so the sum is one), then over y_{T-1}, and so on down to y_1. A hybrid need not correspond to any model one could deploy. It is a bookkeeping device. The two ends of the chain are the distributions of interest: $H_0 = P$ (no aligned steps) and $H_T = Q$ (all aligned steps). Total variation is a metric on distributions, being half the ℓ_1 distance between them viewed as vectors, so the triangle inequality applied along the chain $H_0, H_1, \ldots, H_T$ gives

$$\mathrm{TV}(P, Q) = \mathrm{TV}(H_0, H_T) \le \sum_{t=1}^{T} \mathrm{TV}(H_{t-1}, H_t). \tag{B.6}$$

This is the only inequality in the proof. It remains to compute each term on the right exactly.

Step 2: consecutive hybrids differ at one step only. Fix $t \in \{1, \ldots, T\}$ and compare H_{t-1} with H_t. Both generate tokens $1, \ldots, t-1$ with the aligned model and tokens $t+1, \ldots, T$ with the induced policy. They differ only in which policy generates token t: H_{t-1} uses $\pi_{\theta,S}$ there and H_t uses $\pi_{\theta'}$. To make this visible, split a response as $y = (a, b, c)$ with $a = y_{<t}$ the prefix, $b = y_t$ the token at the differing step, and $c = y_{>t} = (y_{t+1}, \ldots, y_T)$ the continuation, and write

$$\mu(a) = \prod_{i=1}^{t-1} \pi_{\theta'}(y_i \mid s_i), \qquad \Gamma(c \mid a, b) = \prod_{i=t+1}^{T} \pi_{\theta,S}(y_i \mid s_i),$$
$$K_1(b \mid a) = \pi_{\theta,S}(b \mid x, a), \qquad K_2(b \mid a) = \pi_{\theta'}(b \mid x, a).$$

In words: μ is the aligned model's distribution of the first $t-1$ tokens (for $t = 1$ it is the trivial distribution on the empty prefix), K_1 and K_2 are the two next-token distributions at the state $s_t = (x, a)$, and Γ is the induced policy's distribution of the continuation given everything before it. Note that Γ may depend on b through the states s_i for $i > t$. That is allowed, because the only property of Γ that will be used is that it sums to one over c for each fixed (a, b). With this notation,

$$H_{t-1}(a, b, c) = \mu(a)\, K_1(b \mid a)\, \Gamma(c \mid a, b), \qquad H_t(a, b, c) = \mu(a)\, K_2(b \mid a)\, \Gamma(c \mid a, b).$$

These are identities, just the definition of H_{t-1} and H_t with the factors grouped.

Step 3: the distance between consecutive hybrids is an expected one-step distance. By the definition of total variation,

$$\begin{aligned}\mathrm{TV}(H_{t-1}, H_t) &= \tfrac{1}{2}\sum_{a,b,c}\big|\mu(a)K_1(b \mid a)\Gamma(c \mid a,b) - \mu(a)K_2(b \mid a)\Gamma(c \mid a,b)\big| \\ &= \tfrac{1}{2}\sum_{a,b,c}\mu(a)\big|K_1(b \mid a) - K_2(b \mid a)\big|\,\Gamma(c \mid a,b) \\ &= \tfrac{1}{2}\sum_{a,b}\mu(a)\big|K_1(b \mid a) - K_2(b \mid a)\big|\sum_{c}\Gamma(c \mid a,b) \\ &= \tfrac{1}{2}\sum_{a}\mu(a)\sum_{b}\big|K_1(b \mid a) - K_2(b \mid a)\big| = \sum_{a}\mu(a)\ \mathrm{TV}\big(K_1(\cdot \mid a), K_2(\cdot \mid a)\big).\end{aligned}$$

The second line takes the common factors $\mu(a)$ and $\Gamma(c \mid a, b)$ out of the absolute value, which is allowed because they are nonnegative. The third line sums over the continuation c first, which is legitimate because the factor $|K_1 - K_2|$ does not depend on c. The fourth line uses $\sum_c \Gamma(c \mid a, b) = 1$, which holds because $\Gamma(\cdot \mid a, b)$ is a product of next-token distributions and hence itself a distribution over continuations. The last equality is the definition of total variation between the two next-token distributions at the state (x, a). Every step is an equality, and nothing has been discarded. In words: once the prefix is fixed, the continuation after step t is generated identically under both hybrids, so it contributes nothing to their distance. All that remains is the distance at step t, averaged over the prefixes that the aligned model produces. Recalling what μ, K_1 and K_2 are,

$$\mathrm{TV}(H_{t-1}, H_t) = \mathbb{E}_{y_{<t}\sim\pi_{\theta'}}\ \mathrm{TV}\left(\pi_{\theta,S}(\cdot \mid s_t), \pi_{\theta'}(\cdot \mid s_t)\right),$$

where $y_{<t} \sim \pi_{\theta'}$ means that the prefix is drawn from the aligned model's response distribution Q restricted to its first $t-1$ tokens. Inserting this into (B.6) and summing over t proves the lemma.

Why the expectation is under the aligned model. The hybrids were built aligned-first: H_t uses the aligned model for the early steps and the induced policy for the late ones. As a result the prefix distribution μ shared by consecutive hybrids is always the aligned model's. Building the chain in the opposite order, induced-first, would give the same inequality with the expectation over prefixes drawn from $\pi_{\theta,S}$ instead. The two bounds are different, and the lemma as stated is the aligned-first one, so it is not symmetric in the two policies. The aligned-first form is the useful one for the paper: the per-token quantities are then averaged over the states that the target model visits, which are the states one can sample without having the context in hand. ■

B.3 Proposition 1 (bridge), including the restricted-vocabulary form

The restricted form. Proposition 1 also holds with the oscillations taken over a token set of mass at least $1 - \zeta$ under both $\pi_{\theta,S}(\cdot \mid s)$ and $\pi_{\theta'}(\cdot \mid s)$ at every state, with $2T\zeta$ added to the middle and right-hand expressions of (3.4). The full vocabulary's oscillation is typically attained on tokens of negligible probability, which this form avoids. It is proved in the last part of the proof below.

The objects. The compilation error of a context S is $d(S) = \mathbb{E}_{x\sim D}\,\mathrm{TV}\left(\pi_{\theta,S}(y \mid x), \pi_{\theta'}(y \mid x)\right)$, the total variation between the two response distributions averaged over tasks (equation (3.1)). At a state s, the base model's next-token distribution is $\pi_\theta(\cdot \mid s) = \mathrm{softmax}(\mathbf{z})$ for its logit vector $\mathbf{z} \in \mathbb{R}^V$ at that state, and the two shifts $\delta_S(s, \cdot)$ and $\delta_{\theta'}(s, \cdot)$ of (3.2) are the log-ratios of the induced and the aligned next-token probabilities to the base ones. The set Σ_x of states reachable from a task x within the horizon is finite, because tasks and tokens are finite, and every state in it is reached with positive probability by either policy, because both are strictly positive.

Idea of the proof. The proposition assembles the two lemmas. (3.2), read backwards, says that at every state the induced and the aligned next-token distributions are both softmaxes of the base logits plus a shift, so Lemma 9 bounds their distance at that state by $\tanh$ of a quarter of the oscillation of the difference of the shifts. Lemma 10 adds these per-step distances along the response, and averaging over tasks turns the result into the compilation error. That is the middle expression of (3.4). The right-hand expression is obtained by two further relaxations, $\tanh u \le u$ and replacing every oscillation by the worst one over all states, which are convenient but lossy. The restricted-vocabulary form repeats the argument on a subset of tokens that carries almost all the probability. The price is the missing mass, paid twice per step.

Proof of Proposition 1. Step 1: both next-token distributions are the base distribution with a shift added to its logits. Fix a state s and write $p_0 = \pi_\theta(\cdot \mid s) = \mathrm{softmax}(\mathbf{z})$. Rearranging the definition $\delta_{\theta'}(s,v) = \log\big(\pi_{\theta'}(v \mid s)/p_0(v)\big)$ gives $\pi_{\theta'}(v \mid s) = p_0(v)\, e^{\delta_{\theta'}(s,v)}$ for every token v. Now $p_0(v) = e^{\mathbf{z}(v)}/\sum_u e^{\mathbf{z}(u)}$, so $\pi_{\theta'}(v \mid s)$ is proportional to $e^{\mathbf{z}(v)+\delta_{\theta'}(s,v)}$ with a proportionality constant, $1/\sum_u e^{\mathbf{z}(u)}$, that does not depend on v. A probability distribution proportional to $e^{\mathbf{z}(v)+\delta_{\theta'}(s,v)}$ is, by definition, $\mathrm{softmax}(\mathbf{z} + \delta_{\theta'}(s,\cdot))$, since the softmax is the unique normalization of those numbers. Hence

$$\pi_{\theta'}(\cdot \mid s) = \mathrm{softmax}\big(\mathbf{z} + \delta_{\theta'}(s,\cdot)\big) \quad \text{and, in the same way,} \quad \pi_{\theta,S}(\cdot \mid s) = \mathrm{softmax}\big(\mathbf{z} + \delta_S(s,\cdot)\big).$$

This is only (3.2) read backwards. No modeling assumption is involved (see also the paragraph on tilts in Appendix A). In words: whatever the two models do at a state, each of them is the base model with its logits shifted, and the shift is by definition the log-ratio to the base.

Step 2: one step. By Step 1 the two next-token distributions at s have exactly the form required by Lemma 9, with logits $\mathbf{z}$ and shifts $\delta = \delta_S(s,\cdot)$ and $\delta' = \delta_{\theta'}(s,\cdot)$. The lemma gives, at every state s,

$$\mathrm{TV}\big(\pi_{\theta,S}(\cdot \mid s), \pi_{\theta'}(\cdot \mid s)\big) \le \tanh\Big(\tfrac{1}{4}\,\mathrm{osc}_s(\delta_S - \delta_{\theta'})\Big). \tag{B.7}$$

In words: at one state, the two models disagree by at most $\tanh$ of a quarter of the spread, across tokens, of the gap between the shift the context realizes and the shift the weights realize.

Step 3: a whole response, then all tasks. Fix a task x. Lemma 10 bounds the total variation between the two response distributions at x by $\sum_{t=1}^{T} \mathbb{E}_{y_{<t}\sim\pi_{\theta'}}\, \mathrm{TV}\big(\pi_{\theta,S}(\cdot \mid s_t), \pi_{\theta'}(\cdot \mid s_t)\big)$. Inserting (B.7) at each state s_t inside the expectation, which is legitimate because an expectation of a smaller function is smaller (expectations preserve pointwise inequalities), gives

$$\mathrm{TV}\big(\pi_{\theta,S}(y \mid x), \pi_{\theta'}(y \mid x)\big) \le \sum_{t=1}^{T} \mathbb{E}_{y_{<t}\sim\pi_{\theta'}} \tanh\Big(\tfrac{1}{4}\,\mathrm{osc}_{s_t}(\delta_S - \delta_{\theta'})\Big).$$

Averaging both sides over $x \sim D$ (again an expectation of a pointwise inequality) turns the left-hand side into $d(S)$ and produces the middle expression of (3.4).

Step 4: the right-hand form. Two further relaxations produce the right-hand expression of (3.4). First, $\tanh u \le u$ for $u \ge 0$: $\tanh 0 = 0$ and $\tanh'(u) = 1 - \tanh^2 u \le 1$, so $\tanh u = \int_0^u \tanh'(r)\, dr \le u$. Applying this inside each of the T terms with $u = \frac{1}{4}\,\mathrm{osc}_{s_t}(\delta_S - \delta_{\theta'})$ replaces $\tanh(\frac{1}{4}\,\mathrm{osc}_{s_t}(\cdot))$ by $\frac{1}{4}\,\mathrm{osc}_{s_t}(\cdot)$. Second, each oscillation is at most its maximum over all states $s = (x, y_{<t})$ with $t \le T$, a finite set, each reachable with positive probability because both policies are strictly positive. The expression inside the expectations is then a constant, the expectations disappear, and the T terms sum to $T \cdot \frac{1}{4}\max_s \mathrm{osc}_s(\delta_S - \delta_{\theta'})$. Both relaxations lose. Since $\tanh u \ge u - u^3/3$ for $u \ge 0$ (the function $\tanh u - u + u^3/3$ vanishes at 0 and has derivative $u^2 - \tanh^2 u \ge 0$), the first costs at most $u^3/3$, negligible when the oscillation is well below 4. But $\tanh u$ never exceeds 1 while u grows without bound, so for large residuals the linear form can exceed the trivial bound $\mathrm{TV} \le 1$. The second replaces a typical state by the worst one. This is why the middle expression is the one to use for long responses. The same two relaxations applied before averaging over tasks give the per-task form used in Appendix D: $\mathrm{TV}(\pi_{\theta,S}(y \mid x), \pi_{\theta'}(y \mid x)) \le \frac{T}{4}\max_{s\in\Sigma_x} \mathrm{osc}_s(\delta_S - \delta_{\theta'})$, since every s_t inside the expectations belongs to Σ_x. It holds for a context allowed to depend on x, the bound being proved one task at a time.

Restricted vocabulary: why and what. Over a vocabulary of tens of thousands of tokens, the oscillation of a shift difference is typically attained on tokens that both models assign negligible probability, so the full-vocabulary bound can be loose for a reason that has nothing to do with the behavior anyone observes. The restricted form measures the spread only over a set of tokens that carries almost all the probability under both models, and pays for the omitted tokens with their mass. Fix the context S and a state s, and abbreviate $p = \pi_{\theta,S}(\cdot \mid s)$ and $q = \pi_{\theta'}(\cdot \mid s)$. Let $\mathcal{V}_s \subseteq \{1, \dots, V\}$ be a set of tokens with $p(\mathcal{V}_s) \ge 1 - \zeta$ and $q(\mathcal{V}_s) \ge 1 - \zeta$, where $p(\mathcal{V}_s) = \sum_{v\in\mathcal{V}_s} p(v)$ is the probability of the set. For a distribution p and the set $\mathcal{V}_s$, the *conditioned* distribution is

$$p|_{\mathcal{V}_s}(v) = \frac{p(v)\,\mathbf{1}\{v \in \mathcal{V}_s\}}{p(\mathcal{V}_s)},$$

which keeps the probabilities inside $\mathcal{V}_s$ in the same proportions, rescales them to sum to one, and sets the probabilities outside $\mathcal{V}_s$ to zero. In words: it is the distribution of the next token given that the next token lies in $\mathcal{V}_s$.

(a) Conditioning costs exactly the missing mass. Inside $\mathcal{V}_s$, $p|_{\mathcal{V}_s}(v) - p(v) = p(v)\big(1/p(\mathcal{V}_s) - 1\big) \geq 0$, and these differences sum over $v \in \mathcal{V}_s$ to $p(\mathcal{V}_s)\big(1/p(\mathcal{V}_s) - 1\big) = 1 - p(\mathcal{V}_s)$. Outside $\mathcal{V}_s$, $|p|_{\mathcal{V}_s}(v) - p(v)| = p(v)$, and these sum to $1 - p(\mathcal{V}_s)$ as well. Therefore

$$\mathrm{TV}\left(p, p|_{\mathcal{V}_s}\right) = \tfrac{1}{2}\Big[\big(1 - p(\mathcal{V}_s)\big) + \big(1 - p(\mathcal{V}_s)\big)\Big] = 1 - p(\mathcal{V}_s) \leq \zeta,$$

and likewise $\mathrm{TV}(q, q|_{\mathcal{V}_s}) = 1 - q(\mathcal{V}_s) \leq \zeta$. This is an identity, not a bound, until the last step: conditioning moves exactly the mass that lies outside the set, no more.

(b) Three-term triangle inequality. Since total variation is a metric, going from p to q through the two conditioned distributions costs at most the sum of the three legs:

$$\mathrm{TV}(p, q) \leq \mathrm{TV}\left(p, p|_{\mathcal{V}_s}\right) + \mathrm{TV}\left(p|_{\mathcal{V}_s}, q|_{\mathcal{V}_s}\right) + \mathrm{TV}\left(q|_{\mathcal{V}_s}, q\right) \leq \zeta + \mathrm{TV}\left(p|_{\mathcal{V}_s}, q|_{\mathcal{V}_s}\right) + \zeta.$$

(c) The middle term is a contraction on the smaller vocabulary. By Step 1, $p(v) = e^{\mathbf{z}(v)+\delta_S(s,v)} \big/ \sum_u e^{\mathbf{z}(u)+\delta_S(s,u)}$. Conditioning on $\mathcal{V}_s$ replaces the denominator by the sum over $u \in \mathcal{V}_s$ only, because the common normalizing constant cancels in the ratio $p(v)/p(\mathcal{V}_s)$:

$$p|_{\mathcal{V}_s}(v) = \frac{e^{\mathbf{z}(v)+\delta_S(s,v)}}{\sum_{u \in \mathcal{V}_s} e^{\mathbf{z}(u)+\delta_S(s,u)}} \quad (v \in \mathcal{V}_s),$$

which is the softmax, over the token set $\mathcal{V}_s$, of the vector $\mathbf{z} + \delta_S(s, \cdot)$ restricted to $\mathcal{V}_s$. The same holds for $q|_{\mathcal{V}_s}$ with $\delta_{\theta'}$ in place of δ_S. In words: conditioning a softmax on a set of tokens is the softmax of the same logits over that set. The proof of Lemma 9 never used which finite set the tokens come from, so it applies verbatim with $\{1, \ldots, V\}$ replaced by $\mathcal{V}_s$, and gives

$$\mathrm{TV}\left(p|_{\mathcal{V}_s}, q|_{\mathcal{V}_s}\right) \leq \tanh\Big(\tfrac{1}{4} \max_{v \in \mathcal{V}_s} \big(\delta_S - \delta_{\theta'}\big)(s, v) - \tfrac{1}{4} \min_{v \in \mathcal{V}_s} \big(\delta_S - \delta_{\theta'}\big)(s, v)\Big),$$

the oscillation now taken over $\mathcal{V}_s$ only.

(d) Assembling. Combining (b) and (c), the one-step bound (B.7) holds with the oscillation over $\mathcal{V}_s$ and 2ζ added. Running Steps 3 and 4 with this one-step bound adds 2ζ to each of the T terms, hence $2T\zeta$ in total, to both the middle and the right-hand expression of (3.4). The token set $\mathcal{V}_s$ is allowed to depend on the state and on the context S. The restricted form is a statement about one given context, not a uniform statement over contexts. ■

↑ Back to page 3

B.4 Proposition 11 (root-sum-square bridge)

This complementary form of the bridge improves the worst-case dependence on the response length from T to $\sqrt{T}$. Here it is in full, with its hypotheses.

Proposition 11 (root-sum-square bridge)**.** *Write $D_S(s) = \mathrm{osc}_s(\delta_S - \delta_{\theta'})$ for the residual spread at a state. For every context S,*

$$d(S) \;\leq\; \mathbb{E}_{x \sim D} \min\Big\{1,\; \tfrac{1}{4}\sqrt{\textstyle\sum_{t \leq T} \mathbb{E}_{y_{<t} \sim \pi_{\theta'}} D_S(s_t)^2}\Big\} \;\leq\; \min\Big\{1,\; \tfrac{\sqrt{T}}{4} \max_s D_S(s)\Big\}. \tag{B.8}$$

↑ Back to page 3

The objects. The objects are those of Proposition 1: a context S, a task $x \sim D$, states $s_t = (x, y_{<t})$, the induced policy $\pi_{\theta,S}(\cdot \mid s)$ and the aligned policy $\pi_{\theta'}(\cdot \mid s)$, both strictly positive distributions on the V tokens, and the residual shift $\delta_S - \delta_{\theta'}$ whose spread at a state is $\mathrm{osc}_s(\delta_S - \delta_{\theta'})$. Two further quantities appear. For distributions P, Q on a finite set, $\mathrm{KL}(Q\|P) = \sum_\omega Q(\omega) \log\big(Q(\omega)/P(\omega)\big)$ is the Kullback–Leibler divergence, which is nonnegative and, unlike total variation, is not symmetric. For a real random variable W with distribution Q, $\mathrm{Var}_Q(W)$ is its variance. Write $P(\cdot \mid x) = \pi_{\theta,S}(y \mid x)$ and $Q(\cdot \mid x) = \pi_{\theta'}(y \mid x)$ for the two response distributions at the task x, as in Appendix B.2.

Idea of the proof. Proposition 1 pays one full per-token total variation at each of the T positions, because total variation does not add along a trajectory: only the triangle inequality is available, and T terms of size ε give $T\varepsilon$. Divergence does add, exactly, along the trajectories of whichever model the expectation is taken under, by the chain rule for the logarithm of a product. So if the one-step divergence is controlled by the same premise, the response costs a sum of T one-step divergences, and Pinsker converts the total back into total variation with a square root: $\sqrt{T\varepsilon^2} = \sqrt{T}\varepsilon$. The one-step divergence bound is Hoeffding's: the log-ratio of the two policies at a state has range exactly the residual oscillation, and a random variable of range r has variance at most $r^2/4$, which bounds the curvature of a cumulant generating function and hence the divergence by $r^2/8$.

Proof of Proposition 11. Step 1: at one state, the divergence is at most an eighth of the squared oscillation. Fix a state s and write $q = \pi_{\theta'}(\cdot \mid s)$ and $p = \pi_{\theta,S}(\cdot \mid s)$, both strictly positive, and

$$r(v) = \log \frac{p(v)}{q(v)}, \qquad F(\tau) = \log \sum_v q(v) e^{\tau r(v)} \quad (\tau \in \mathbb{R}).$$

In words: F is the cumulant generating function of the log-ratio r under the aligned policy. Three values of F and its derivatives are needed. First, $F(0) = \log \sum_v q(v) = 0$. Second, $F(1) = \log \sum_v q(v)\, p(v)/q(v) = \log \sum_v p(v) = 0$, because both are probability distributions. This is the one place where normalization enters. Third, differentiating under the finite sum,

$$F'(\tau) = \frac{\sum_v q(v) r(v) e^{\tau r(v)}}{\sum_v q(v) e^{\tau r(v)}} = \mathbb{E}_{q_\tau} r, \qquad q_\tau(v) = \frac{q(v) e^{\tau r(v)}}{\sum_u q(u) e^{\tau r(u)}},$$

so that $F'(0) = \mathbb{E}_q r = \sum_v q(v) \log\big(p(v)/q(v)\big) = -\operatorname{KL}(q\|p)$, and differentiating once more gives $F''(\tau) = \operatorname{Var}_{q_\tau}(r)$.

Now, r is a function of the token only, and by (3.2) it is the residual *exactly*: both shifts are log-ratios against the same base distribution, so the base cancels and $r(v) = \big(\delta_S - \delta_{\theta'}\big)(s, v)$ with no additive constant left over. Its range is therefore

$$\max_v r(v) - \min_v r(v) = \operatorname{osc}_s(\delta_S - \delta_{\theta'}) =: D_S(s),$$

the same number at every τ, since the tilting q_τ changes the weights but not the values $r(v)$. Popoviciu's inequality bounds the variance of a random variable confined to an interval of length $D_S(s)$ by $D_S(s)^2/4$ (the same inequality used in Appendix B.1), so $F''(\tau) \le D_S(s)^2/4$ for every $\tau \in [0, 1]$. Taylor's theorem with the integral remainder, applied to F on $[0, 1]$, gives

$$0 = F(1) = F(0) + F'(0) + \int_0^1 (1-\tau) F''(\tau)\, d\tau = -\operatorname{KL}(q\|p) + \int_0^1 (1-\tau) F''(\tau)\, d\tau,$$

hence, using $\int_0^1 (1-\tau)\, d\tau = \frac{1}{2}$,

$$\operatorname{KL}\big(\pi_{\theta'}(\cdot \mid s) \,\|\, \pi_{\theta,S}(\cdot \mid s)\big) = \int_0^1 (1-\tau) F''(\tau)\, d\tau \le \frac{D_S(s)^2}{8}. \tag{B.9}$$

In words: a bounded log-ratio range controls the divergence quadratically, whereas Lemma 9 controlled the distance linearly. This is exactly the quadratic gain that the square root will later convert into $\sqrt{T}$.

Step 2: divergences add along the aligned model's trajectories. Fix a task x. Both response distributions factor along the response, so for every response y,

$$\log \frac{Q(y \mid x)}{P(y \mid x)} = \sum_{t=1}^{T} \log \frac{\pi_{\theta'}(y_t \mid s_t)}{\pi_{\theta,S}(y_t \mid s_t)}.$$

This is an identity, not a bound: the logarithm of a product is the sum of the logarithms. Taking the expectation under Q and conditioning each term on its own prefix,

$$\operatorname{KL}\big(Q(\cdot \mid x) \,\|\, P(\cdot \mid x)\big) = \sum_{t=1}^{T} \mathbb{E}_{y_{<t} \sim \pi_{\theta'}} \operatorname{KL}\big(\pi_{\theta'}(\cdot \mid s_t) \,\|\, \pi_{\theta,S}(\cdot \mid s_t)\big),$$

because the expectation under Q of the t-th term is, by the tower rule, the expectation over the prefix of the conditional divergence at s_t. This is the chain rule for divergence, and it is an equality. Nothing is discarded. Inserting (B.9) at each state,

$$\mathrm{KL}\left(Q(\cdot \mid x) \,\|\, P(\cdot \mid x)\right) \le \frac{1}{8} \sum_{t=1}^{T} \mathbb{E}_{y_{<t} \sim \pi_{\theta'}} D_S(s_t)^2. \tag{B.10}$$

Step 3: Pinsker, then averaging over tasks. Pinsker's inequality states $\mathrm{TV}(Q, P) \le \sqrt{\mathrm{KL}(Q\|P)/2}$. Combining it with (B.10) and using that the square root is increasing,

$$\mathrm{TV}\left(P(\cdot \mid x), Q(\cdot \mid x)\right) \le \sqrt{\frac{1}{16} \sum_{t=1}^{T} \mathbb{E}_{y_{<t} \sim \pi_{\theta'}} D_S(s_t)^2} = \frac{1}{4} \sqrt{\sum_{t=1}^{T} \mathbb{E}_{y_{<t} \sim \pi_{\theta'}} D_S(s_t)^2},$$

where total variation is symmetric so the order of the arguments does not matter. A total variation never exceeds 1, so at each task the smaller of the two bounds holds,

$$\mathrm{TV}\left(P(\cdot \mid x), Q(\cdot \mid x)\right) \le \min\left\{1,\ \tfrac{1}{4}\sqrt{\textstyle\sum_{t=1}^{T} \mathbb{E}_{y_{<t} \sim \pi_{\theta'}} D_S(s_t)^2}\right\},$$

and averaging this over $x \sim D$ turns the left-hand side into $d(S)$ and gives the first inequality of (B.8). The clipping has to be done here, inside the average and before the square root is discarded: the square root itself is an unbounded quantity, and asserting that it is at most 1 would be false. A two-token example makes this concrete: with a uniform target, an induced distribution $\mathrm{softmax}(4, -4)$, $T = 1$ and the base equal to the target, the residual oscillation is 8, so the square root term equals 2 while the true distance is 0.4997.

Step 4: the uniform form. If $D_S(s) \le M$ at every reachable state, each of the T expectations in the sum is at most M^2, so the sum is at most TM^2 and its square root is $\sqrt{T}\,M$. Dividing by 4 gives $\sqrt{T}M/4$ with $M = \max_s \mathrm{osc}_s(\delta_S - \delta_{\theta'})$. Only monotonicity of the square root is used here, not concavity. Taking the minimum with 1 on both sides preserves the inequality, and the average of the clipped per-task bounds is at most the clip of the average, since $\mathbb{E}\min\{1, f\} \le \min\{1, \mathbb{E}f\}$. This is the second inequality of (B.8), and the intermediate form $\min\{1, \frac{1}{4}\mathbb{E}_x\sqrt{\cdots}\}$ also holds but is weaker than the task-wise one. ■

Which of the two bridges to use. The two bounds are not comparable in general and both are available, so one takes their minimum. Proposition 1 is sharper per token, giving the exact $\tanh(\Delta/4)$ at $T = 1$ against the weaker $\Delta/4$. Proposition 11 is better as soon as the horizon is long and the residual small, since for a uniform Δ the two read $T\Delta/4$ and $\sqrt{T}\Delta/4$. Three cautions attach to the second. It needs the second moment of the residual oscillations along aligned trajectories, and the square of an empirical mean is not an average of squares. A restricted token set requires its own missing-mass accounting exactly as in Appendix B.3. And a sample of states does not by itself certify a uniform bound over all reachable histories.

B.5 Lemma 12 (converse)

Lemma 12 (converse)**.** *Let p, q be positive distributions over tokens and let two tokens a, b satisfy $\log \frac{p(a)}{q(a)} - \log \frac{p(b)}{q(b)} \ge \Delta$ and $q(a), q(b) \ge m$. Then $\mathrm{TV}(p, q) \ge m(1 - e^{-\Delta/2})$. At the first token of a response, $d(S) \ge D(x)$ $\mathrm{TV}\left(\pi_{\theta,S}(y_1 \mid x), \pi_{\theta'}(y_1 \mid x)\right)$ for every task x.*

↑ Back to page 5 or page 9

The objects. Two strictly positive probability distributions p and q over the vocabulary are given, together with two tokens a and b, a number $\Delta \ge 0$ and a number $m > 0$, such that

$$\log \frac{p(a)}{q(a)} - \log \frac{p(b)}{q(b)} \ge \Delta, \qquad q(a) \ge m, \qquad q(b) \ge m.$$

In words: relative to q, the distribution p favors token a over token b by a factor of at least e^{Δ}, and both tokens have probability at least m under q. In use, p is the induced next-token distribution and q

the aligned one at a first-token state, the two tokens are those at which the residual $\delta_S - \delta_{\theta'}$ attains its maximum and its minimum, and Δ is the residual's oscillation. The hypothesis is then exactly the statement that the residual has oscillation at least Δ, because additive constants cancel in the difference of two log-ratios. The first claim is the lower bound $\mathrm{TV}(p, q) \geq m(1 - e^{-\Delta/2})$. The second claim concerns whole responses and says that the compilation error $d(S)$ is at least $D(x)$ times the total variation between the two first-token distributions at any task x.

Idea of the proof. The upper bounds of the paper say that a small residual oscillation forces a small policy error. This lemma is the converse, a large residual forces a large error, and it is where the asymmetry between upper and lower bounds appears. A ratio gap of e^{Δ} between two tokens cannot be shared out so that neither token deviates much: at least one of the two must carry a multiplicative discrepancy of $e^{\Delta/2}$ on its own. A multiplicative discrepancy on a token becomes an absolute discrepancy once the token's probability is known to be at least m, and any single token's absolute discrepancy is a lower bound on total variation. The mass factor m is unavoidable: two distributions can differ by an enormous ratio on tokens of negligible probability and still be close in total variation. The first-token statement follows because reporting only the first token of a response can only bring the two response distributions closer.

Proof of Lemma 12. Step 1: one of the two tokens carries a large multiplicative discrepancy. Write $r_a = p(a)/q(a)$ and $r_b = p(b)/q(b)$ for the two likelihood ratios, which are positive because both distributions are. The hypothesis says $\log r_a - \log r_b \geq \Delta$, that is, $r_a/r_b \geq e^{\Delta}$. Suppose, for contradiction, that both $r_a < e^{\Delta/2}$ and $r_b > e^{-\Delta/2}$ held. Then $r_a/r_b < e^{\Delta/2}/e^{-\Delta/2} = e^{\Delta}$, contradicting the hypothesis. Hence at least one of the following holds:

$$\text{(A)}\ \ r_a \geq e^{\Delta/2} \qquad \text{or} \qquad \text{(B)}\ \ r_b \leq e^{-\Delta/2}.$$

In words: token a is over-represented in p by a factor of at least $e^{\Delta/2}$, or token b is under-represented in p by at least that factor. The gap of e^{Δ} cannot be split so that both halves are smaller than $e^{\Delta/2}$.

Step 2: a multiplicative discrepancy on a token of mass m is an additive gap. In case (A), $p(a) - q(a) = q(a)(r_a - 1)$, and $r_a - 1 \geq e^{\Delta/2} - 1 \geq 0$, so the factor $q(a)$ multiplies a nonnegative number and the lower bound $q(a) \geq m$ may be inserted:

$$p(a) - q(a) \geq m\,(e^{\Delta/2} - 1) \geq m\,(1 - e^{-\Delta/2}),$$

where the last inequality is $e^u - 1 \geq 1 - e^{-u}$ for $u \geq 0$, equivalent to $e^u + e^{-u} \geq 2$, itself the arithmetic–geometric mean inequality for the pair e^u, e^{-u} whose product is one. In case (B), $q(b) - p(b) = q(b)(1 - r_b)$, and $1 - r_b \geq 1 - e^{-\Delta/2} \geq 0$, so again $q(b) \geq m$ may be inserted:

$$q(b) - p(b) \geq m\,(1 - e^{-\Delta/2}).$$

In either case some single token v satisfies $|p(v) - q(v)| \geq m(1 - e^{-\Delta/2})$. The two cases give the same bound because $e^{\Delta/2} - 1$ was deliberately weakened to $1 - e^{-\Delta/2}$ in case (A). Case (B) is the binding one.

Step 3: a gap on one token is a lower bound on total variation. Total variation can be written as $\mathrm{TV}(p, q) = \max_A \big(p(A) - q(A)\big)$, the maximum over subsets A of the vocabulary (Appendix A). Taking $A = \{v\}$ gives $\mathrm{TV}(p, q) \geq p(v) - q(v)$, and taking the complement of $\{v\}$ gives $\mathrm{TV}(p, q) \geq q(v) - p(v)$, because $p(A^c) - q(A^c) = (1 - p(v)) - (1 - q(v)) = q(v) - p(v)$. Together, $\mathrm{TV}(p, q) \geq |p(v) - q(v)|$ for every token v. With the token from Step 2, $\mathrm{TV}(p, q) \geq m(1 - e^{-\Delta/2})$, which is the first claim. Note that only the masses $q(a)$ and $q(b)$ under q were used. Nothing was assumed about p beyond positivity. In use this means the mass condition can be checked on the aligned model alone, before any context is proposed.

Step 4: the first-token statement. Fix a task x and consider the two response distributions $\pi_{\theta,S}(y \mid x)$ and $\pi_{\theta'}(y \mid x)$ over sequences $y = (y_1, \ldots, y_T)$. The map $y \mapsto y_1$ that reports the first token is deterministic, and total variation cannot increase under a deterministic map of the outcome: in the set form $\max_A(p(A) - q(A))$, every event defined in terms of y_1 is also an event defined in terms of y, so the maximum over the smaller family of events is no larger (Appendix A). The image distributions of y_1 are the two first-token distributions $\pi_{\theta,S}(y_1 \mid x)$ and $\pi_{\theta'}(y_1 \mid x)$, so

$$\mathrm{TV}\left(\pi_{\theta,S}(y \mid x), \pi_{\theta'}(y \mid x)\right) \geq \mathrm{TV}\left(\pi_{\theta,S}(y_1 \mid x), \pi_{\theta'}(y_1 \mid x)\right).$$

In words: if two models already disagree on the first token, they disagree at least as much on whole responses. Finally, $d(S) = \sum_{x'} D(x')\ \mathrm{TV}\big(\pi_{\theta,S}(y \mid x'), \pi_{\theta'}(y \mid x')\big)$ is a sum of nonnegative terms, so keeping only the term for the task x gives $d(S) \geq D(x)\ \mathrm{TV}\big(\pi_{\theta,S}(y \mid x), \pi_{\theta'}(y \mid x)\big)$, and combining with the display above proves the second claim.

Why lower bounds are stated at the first token. The same argument at a later state $s = (x, y_{<t})$ would go through the event that the response passes through s, and would therefore carry the probability of reaching s as an additional factor. At the first token that probability is one, so the bound is cleanest there. This is why every lower bound in the paper is stated at first-token states. ■

C PROOFS FOR PREFIXES

This appendix proves Theorem 3 and Proposition 4, characterizes key–value prefixes, and verifies the three claims of Assumption 2 about the canonical operator. Two conventions hold throughout. For two functions f, f' of the token v at a fixed state, $f \equiv f'$ means that $f(v) - f'(v)$ is the same number at every token. Since a softmax is unchanged by such a constant, and osc_s annihilates it, every statement about shifts is made modulo this relation. Vectors are columns, $\|\cdot\|_2$ is the Euclidean norm, $\|A\|_{\mathrm{op}} = \sigma_1(A)$ is the operator norm, and $\sigma_1(A) \geq \sigma_2(A) \geq \cdots$ are the singular values in nonincreasing order, with $\sigma_j(A) = 0$ once j exceeds the rank.

Roadmap. The argument has four layers. (1) In unnormalized linear attention the head output is bilinear in the query and the content, and a key–value prefix *adds* a fixed matrix B_P to it, so a prefix of length n produces exactly the shifts of matrices of rank at most n. (2) Two matrices can produce the same shift, parts of a matrix being invisible to the reachable queries or to the logits, and the canonical operator $B^\star$ is what remains after those parts are removed. (3) The rank theorem is then low-rank approximation of $B^\star$, plus two conversions between matrix distance and oscillation. (4) A value-weight update is bilinear too, but with a core that moves with the state.

The objects. The following symbols are used across the appendix.

- V is the vocabulary size, d the head dimension, and d_{in} the dimension of the token embeddings.
- A state $s = (x, y_{<t})$ consists of $|s|$ tokens with embeddings $\mathbf{c}_1, \ldots, \mathbf{c}_{|s|} \in \mathbb{R}^{d_{\mathrm{in}}}$. The *content matrix* $C_s \in \mathbb{R}^{|s| \times d_{\mathrm{in}}}$ has these embeddings as its rows, $\mathbf{c}_i^\top$ being row i.
- $W_K, W_V \in \mathbb{R}^{d_{\mathrm{in}} \times d}$ are the key and value maps of the head. The matrices $C_s W_K \in \mathbb{R}^{|s| \times d}$ and $C_s W_V \in \mathbb{R}^{|s| \times d}$ hold the keys and values of the $|s|$ content tokens as rows.
- $\mathbf{q}(s) \in \mathbb{R}^d$ is the query at state s. Assumption 13 normalizes it to $\|\mathbf{q}(s)\|_2 \leq 1$. For a task x, $\mathbf{q}(x)$ abbreviates the *first-token query* $\mathbf{q}((x, \emptyset))$, the query at the state where nothing of the response has been generated yet.
- $\mathbf{o}(s) \in \mathbb{R}^d$ is the head output. The matrix $G \in \mathbb{R}^{d \times V}$ combines the output projection with the unembedding, its column $\mathbf{g}_v \in \mathbb{R}^d$ belonging to token v, and $g = \max_v \|\mathbf{g}_v\|_2$. The logit of token v is $z(s, v) = \mathbf{o}(s)^\top \mathbf{g}_v$, so $\pi_\theta(\cdot \mid s) = \mathrm{softmax}(z(s, \cdot))$.
- A key–value prefix P of length n consists of $P_K, P_V \in \mathbb{R}^{n \times d}$, n free key vectors and n free value vectors as rows. We write $B_P = P_K^\top P_V \in \mathbb{R}^{d \times d}$. The base model with P inserted follows the induced policy $\pi_{\theta,P}$, and $\delta_P(s, v) = \log\big(\pi_{\theta,P}(v \mid s)/\pi_\theta(v \mid s)\big)$ is its induced shift. Here $d(P)$ is its compilation error (3.1) and $E_n^{\mathrm{KV}} = \inf_{|P| \leq n} d(P)$ the frontier of the class.
- $\delta_{\theta'}(s, v) = \log\big(\pi_{\theta'}(v \mid s)/\pi_\theta(v \mid s)\big)$ is the update. Assumption 2 fixes a decomposition $\delta_{\theta'}(s, v) \equiv \mathbf{q}(s)^\top B \mathbf{g}_v + \nu(s, v)$ with $B \in \mathbb{R}^{d \times d}$ and a non-bilinear residual ν, and sets $\bar{\nu} = \max_s \mathrm{osc}_s\, \nu$ over reachable states. In words: B is the part of the update that looks like a prefix, and ν is everything else.
- $\Pi_Q \in \mathbb{R}^{d \times d}$ is the orthogonal projector onto $\mathrm{span}\{\mathbf{q}(s) : s \text{ reachable}\}$, the subspace of $\mathbb{R}^d$ that queries actually visit. $N = \{\mathbf{w} \in \mathbb{R}^d : \mathbf{w}^\top \mathbf{g}_v \text{ does not depend on } v\}$ is the subspace of hidden directions the logits cannot see (moving the head output along such a direction adds the same amount to every logit, which the softmax ignores), and $\Pi_G \in \mathbb{R}^{d \times d}$ is the

orthogonal projector onto its orthogonal complement $N^{\perp}$. Both projectors are symmetric and idempotent: $\Pi^{\top} = \Pi$, $\Pi^2 = \Pi$, and $\Pi\mathbf{a} = \mathbf{a}$ exactly when $\mathbf{a}$ lies in the range of Π.

- The canonical operator is $B^{\star} = \Pi_Q B \Pi_G \in \mathbb{R}^{d\times d}$: the matrix B with the directions that queries never reach (on the left) and the directions that logits never see (on the right) projected away.
- $\kappa = \min\{\mathrm{osc}_v(\mathbf{w}^{\top}\mathbf{g}_v) : \mathbf{w} \in \operatorname{range}\Pi_G,\ \|\mathbf{w}\|_2 = 1\}$ (readout visibility: how much spread across tokens a unit hidden vector is guaranteed to produce once the invisible directions are excluded) and $\gamma = \min\{\max_x |\mathbf{u}^{\top}\mathbf{q}(x)| : \mathbf{u} \in \operatorname{range}\Pi_Q,\ \|\mathbf{u}\|_2 = 1\}$ (query richness: how strongly the best-aligned first-token query is guaranteed to overlap with any unit direction of the reachable query space), the inner maximum running over tasks.
- $H_n = \inf\{\max_s \mathrm{osc}_s(\mathbf{q}(s)^{\top}B'\mathbf{g}_{\cdot} - \delta_{\theta'}(s,\cdot)) : B' \in \mathbb{R}^{d\times d},\ \operatorname{rank} B' \le n\}$ is the rank-n frontier in shift space, the maximum over reachable states: the smallest worst-state residual oscillation any rank-n matrix can achieve. The effective rank is $r_{\mathrm{eff}} = \min\{n : H_n = 0\}$, with the convention $r_{\mathrm{eff}} = +\infty$ when no H_n vanishes. Since the matrices are $d \times d$, H_n is constant for $n \ge d$, so an update outside the bilinear image stays outside it however many prefix positions are allowed.

C.1 The statements

The model, Assumption 2, Theorem 3 and Proposition 4 are stated in Sections 4 and 4.2, and the shift of a key–value prefix is (4.4). Every rank-n matrix is a key–value prefix, so $E_n^{\mathrm{KV}} \le \frac{T}{4} H_n$ by Proposition 1, and the proof brackets H_n by the spectrum of a decomposition's canonical operator $B^{\star}$. That operator depends on the decomposition, so the result is a bracket and not a formula. The effective rank $r_{\mathrm{eff}} = \min\{n : H_n = 0\}$ is an exact minimum. The one assumption stated only here fixes the constants.

Assumption 13 (readout constants)**.** $\|\mathbf{q}(s)\|_2 \le 1$ at every state and $g = \max_v \|\mathbf{g}_v\|_2$. Write $\kappa = \min \mathrm{osc}_v(\mathbf{w}^{\top}\mathbf{g}_v)$ over unit $\mathbf{w}$ in the range of Π_G (*readout visibility*, how much spread in the logits a unit hidden displacement is guaranteed to make) and $\gamma = \min_{\mathbf{u}} \max_x |\mathbf{u}^{\top}\mathbf{q}(x)|$ over unit $\mathbf{u}$ in the range of Π_Q (*query richness*, how well some task's first-token query sees the worst direction). Here $\kappa > 0$ by construction, and we assume $\gamma > 0$, which holds as soon as the first-token queries span the reachable query space. Query richness enters only the lower bound (iii).

For Proposition 4, a moving core is on its own no obstruction, which is why the collision of queries is in the hypothesis. With queries $\mathbf{e}_1, \mathbf{e}_2$ and cores $\mathrm{diag}(1,0), \mathrm{diag}(0,1)$ the single fixed matrix I satisfies $\mathbf{q}_i^{\top} I = \mathbf{q}_i^{\top} B_i$ at both tasks, so differing cores alone leave no residual. For $\mathbf{q}_1 \ne \mathbf{q}_2$ the right-hand side of (4.6) loses $g\|B_P\|_{\mathrm{op}}\|\mathbf{q}_1 - \mathbf{q}_2\|_2$, so a large-norm prefix can separate nearby tasks.

C.2 The head, written out

Idea. An attention head scores every content token by comparing the query with that token's key and returns a weighted sum of the tokens' values. In softmax attention the scores are exponentiated and normalized to sum to one. In the idealization of this paper they are not. The consequence, on which the whole appendix rests, is that the head output is *bilinear*, linear in the query and quadratic in the content, so adding tokens adds terms to it rather than rescaling the existing ones.

Using C_s, W_K, W_V, $\mathbf{q}(s)$ and $\mathbf{o}(s)$ as listed above, unnormalized linear attention scores each content token by the inner product of the query with that token's key and sums the values weighted by these scores:

$$\mathbf{o}(s) = \sum_{i=1}^{|s|} \underbrace{\left(\mathbf{q}(s)^{\top} W_K^{\top}\mathbf{c}_i\right)}_{\text{score of token } i,\ \text{a scalar}} \underbrace{W_V^{\top}\mathbf{c}_i}_{\text{value of token } i,\ \in\mathbb{R}^d}.$$

Here $W_K^{\top}\mathbf{c}_i$ is the i-th column of $(C_s W_K)^{\top}$, so the row of scores is $\mathbf{q}(s)^{\top}(C_s W_K)^{\top} \in \mathbb{R}^{1\times|s|}$, and $W_V^{\top}\mathbf{c}_i$ is the i-th row of $C_s W_V$ written as a column. Multiplying the score row by $C_s W_V$ forms exactly the weighted sum of value rows, hence

$$\mathbf{o}(s)^{\top} = \mathbf{q}(s)^{\top}(C_s W_K)^{\top}(C_s W_V) \in \mathbb{R}^{1\times d}, \quad \text{i.e.} \quad \mathbf{o}(s) = (C_s W_V)^{\top}(C_s W_K)\,\mathbf{q}(s) \in \mathbb{R}^d,$$

the expression quoted in the model above. This is an identity, a rewriting of the sum. In words: the head output is the query pushed through one $d \times d$ matrix, $(C_s W_V)^{\top}(C_s W_K)$, built from the content

of the state, and it is linear in $\mathbf{q}(s)$ and quadratic in C_s. The absence of a softmax is what makes it so, and it is why prepending a prefix adds to the content terms rather than rescaling them (with softmax attention and unchanged content queries and keys, a key–value prefix rescales the original content attention weights by a common factor for each query, as Petrov et al. (2024b) show).

C.3 KEY–VALUE PREFIXES

The objects. Besides the head above, this subsection uses a prefix $P_K, P_V \in \mathbb{R}^{n\times d}$, the matrix $B_P = P_K^\top P_V$, the prefixed policy $\pi_{\theta,P}$ and its shift δ_P.

Idea. Three questions are answered in turn. What does a prefix do to the head? It adds one fixed matrix to the bilinear form, the same at every state. What does that do to the next-token distribution? It shifts every logit by a bilinear term, which is the induced shift up to a constant the softmax ignores. Which shifts can a prefix of length n produce? Exactly those of matrices of rank at most n, in both directions, because a product of an n-column matrix with an n-row matrix has rank at most n, and any matrix of rank at most n factors that way.

Step 1: the prefix adds one fixed matrix. A key–value prefix inserts n extra positions whose keys are the rows of P_K and whose values are the rows of P_V, ahead of the content. Write $K_s = C_s W_K$ and $V_s = C_s W_V$ for the keys and values of the content, as in Section 4. The head then attends over $|s| + n$ positions, so its key and value matrices are the stacked matrices

$$K_{s,P} = \begin{bmatrix} K_s \\ P_K \end{bmatrix}, \qquad V_{s,P} = \begin{bmatrix} V_s \\ P_V \end{bmatrix} \in \mathbb{R}^{(|s|+n)\times d}.$$

Multiplying in blocks, the product of the transpose of the first with the second splits into a content part and a prefix part:

$$\begin{aligned} K_{s,P}^\top V_{s,P} &= \begin{bmatrix} K_s^\top & P_K^\top \end{bmatrix} \begin{bmatrix} V_s \\ P_V \end{bmatrix} \\ &= K_s^\top V_s + P_K^\top P_V, \end{aligned}$$

because the transpose of a stacked matrix is the side-by-side matrix $[K_s^\top \;\; P_K^\top]$ and a side-by-side times a stacked matrix is the sum of the blockwise products. Hence, with $B_P = P_K^\top P_V \in \mathbb{R}^{d\times d}$, the prefixed head output is

$$\begin{aligned} \mathbf{o}_P(s)^\top &= \mathbf{q}(s)^\top K_{s,P}^\top V_{s,P} \\ &= \mathbf{q}(s)^\top \big(K_s^\top V_s + P_K^\top P_V\big) \\ &= \mathbf{q}(s)^\top K_s^\top V_s + \mathbf{q}(s)^\top B_P \\ &= \mathbf{o}(s)^\top + \mathbf{q}(s)^\top B_P. \end{aligned}$$

This too is an identity. In words: the prefix *adds* the vector $B_P^\top \mathbf{q}(s)$ to the head output, through one fixed $d \times d$ matrix that is the same at every state and every task. Only the query changes from state to state. This additivity is a consequence of the missing softmax (with a softmax, the prefix positions would compete for a share of a total weight of one, and the content terms would be rescaled), and it is the single algebraic fact behind Theorem 3.

Step 2: from head output to shift. Logits are linear in the head output, $z(s,v) = \mathbf{o}(s)^\top \mathbf{g}_v$, so the prefixed logits are

$$\begin{aligned} z_P(s,v) &= \mathbf{o}_P(s)^\top \mathbf{g}_v \\ &= \big(\mathbf{o}(s)^\top + \mathbf{q}(s)^\top B_P\big)\mathbf{g}_v \\ &= z(s,v) + \mathbf{q}(s)^\top B_P \mathbf{g}_v. \end{aligned}$$

The induced shift is the difference of log-probabilities, and a log-probability is a logit minus the log-normalizer of its softmax. The original log-probability is

$$\log \pi_\theta(v \mid s) = z(s,v) - \log \sum_u e^{z(s,u)},$$

while the log-probability with the prefix is

$$\log \pi_{\theta,P}(v \mid s) = z_P(s,v) - \log \sum_u e^{z_P(s,u)}.$$

Taking their difference gives

$$\begin{aligned}\delta_P(s,v) &= \log \frac{\pi_{\theta,P}(v \mid s)}{\pi_\theta(v \mid s)} \\ &= z_P(s,v) - z(s,v) - \Big(\log \sum_u e^{z_P(s,u)} - \log \sum_u e^{z(s,u)}\Big) \\ &= \mathbf{q}(s)^\top B_P \mathbf{g}_v - c(s),\end{aligned}$$

where $c(s) = \log \sum_u e^{z_P(s,u)} - \log \sum_u e^{z(s,u)}$ is the difference of the two log-normalizers and does not depend on v. In the notation fixed above, $\delta_P(s,\cdot) \equiv \mathbf{q}(s)^\top B_P \mathbf{g}_\cdot$, which is (4.4). Why may the constant be dropped? Because every statement below is made in osc_s, which subtracts the minimum over tokens from the maximum over tokens and so annihilates anything that is the same at every token:

$$\mathrm{osc}_s \big(f(s,\cdot) - c(s)\big) = \mathrm{osc}_s\, f(s,\cdot).$$

In words: a prefix shifts the log-probability of token v by the bilinear quantity $\mathbf{q}(s)^\top B_P \mathbf{g}_v$, up to the bookkeeping constant that keeps the probabilities summing to one.

Step 3: which matrices a prefix of length n reaches. $B_P = P_K^\top P_V$ is a product of a $d \times n$ matrix and an $n \times d$ matrix, so $\mathrm{rank}\, B_P \le \min\{n, d\}$: a prefix of length n cannot produce more than n independent directions. Conversely, let $B' \in \mathbb{R}^{d\times d}$ have rank $r \le n$. A rank factorization writes $B' = A_1 A_2$ with $A_1 \in \mathbb{R}^{d\times r}$ and $A_2 \in \mathbb{R}^{r\times d}$ (for instance from the singular value decomposition, $A_1 = U_r \Sigma_r$ and $A_2 = V_r^\top$). Padding A_1 with $n-r$ zero columns and A_2 with $n-r$ zero rows gives matrices $\tilde{A}_1 \in \mathbb{R}^{d\times n}$ and $\tilde{A}_2 \in \mathbb{R}^{n\times d}$ with $\tilde{A}_1 \tilde{A}_2 = A_1 A_2 = B'$ (the padded zeros multiply to zero), and $P_K = \tilde{A}_1^\top$, $P_V = \tilde{A}_2$ is a key–value prefix of length n with $B_P = B'$. So key–value prefixes of length n realize exactly the shifts $\mathbf{q}(s)^\top B' \mathbf{g}_v$ with $\mathrm{rank}\, B' \le n$, which is the "every matrix of rank at most n arises this way" of the model above. This is why the infimum over prefixes of length n and the infimum over matrices of rank at most n are the same object, and why the rest of the appendix can speak of matrices instead of prefixes.

C.4 The canonical operator

The objects. This subsection uses $B \in \mathbb{R}^{d\times d}$ from Assumption 2, the subspace N, the projectors Π_Q and Π_G, and $B^\star = \Pi_Q B \Pi_G$. Two facts about N are used. First, $\mathbf{w} \in N$ if and only if $\mathbf{w}^\top(\mathbf{g}_v - \mathbf{g}_{v'}) = 0$ for all pairs of tokens, so $N = \mathrm{span}\{\mathbf{g}_v - \mathbf{g}_{v'} : v, v'\}^\perp$ and $N^\perp = \mathrm{span}\{\mathbf{g}_v - \mathbf{g}_{v'}\}$. Here Π_G projects onto $N^\perp$, and $I - \Pi_G$ projects onto N. Second, $\Pi_G \mathbf{w} = 0$ if and only if $\mathbf{w} \in N$, because the vectors annihilated by an orthogonal projector are exactly those orthogonal to its range.

Idea. A matrix B produces a shift only through the numbers $\mathbf{q}(s)^\top B \mathbf{g}_v$, and only up to an additive constant in v. Two parts of B therefore never show: whatever B does to vectors that no reachable query has a component along, and whatever it does that moves the head output along a direction in N, since such a move adds the same number to every logit. Projecting these two parts away leaves the canonical operator. The three claims of Assumption 2 say that this projection loses nothing (same shift), identifies the shift (two matrices with the same shift have the same canonical operator), and is the most economical representative (smallest rank). Each is a short linear-algebra argument.

(a) The shift is unchanged. Write B as the sum of three pieces,

$$B = \Pi_Q B \Pi_G + \Pi_Q B (I - \Pi_G) + (I - \Pi_Q) B,$$

an identity, since the second and third pieces expand to $\Pi_Q B - \Pi_Q B \Pi_G$ and $B - \Pi_Q B$, and the three pieces sum to B. Fix a reachable state s and consider the contribution of each piece to $\mathbf{q}(s)^\top B \mathbf{g}_v$. The third piece contributes $\mathbf{q}(s)^\top (I - \Pi_Q) B \mathbf{g}_v = \big((I - \Pi_Q)\mathbf{q}(s)\big)^\top B \mathbf{g}_v = 0$, because $\mathbf{q}(s)$ lies in the range of Π_Q and so $(I - \Pi_Q)\mathbf{q}(s) = 0$: no reachable query has a component outside the range of Π_Q, so this piece is never activated. The second piece contributes $\mathbf{q}(s)^\top \Pi_Q B (I - \Pi_G) \mathbf{g}_v = \mathbf{u}^\top \mathbf{g}_v$

with $\mathbf{u} = (I - \Pi_G)B^\top \Pi_Q \mathbf{q}(s)$, which lies in the range of $I - \Pi_G$, that is, in N. By the definition of N, $\mathbf{u}^\top \mathbf{g}_v$ does not depend on v: this piece adds the same number to every logit, which the softmax ignores. Hence $\mathbf{q}(s)^\top B\mathbf{g}_v \equiv \mathbf{q}(s)^\top B^\star \mathbf{g}_v$ at every reachable state: B and $B^\star$ give the same shift up to an additive constant that does not depend on the token.

(b) Uniqueness. Suppose B_1 and B_2 give the same shift, $\mathbf{q}(s)^\top B_1 \mathbf{g}_v \equiv \mathbf{q}(s)^\top B_2 \mathbf{g}_v$ at every reachable state, and put $D = B_1 - B_2$. Then $\mathbf{q}(s)^\top D\mathbf{g}_v = (D^\top \mathbf{q}(s))^\top \mathbf{g}_v$ does not depend on v, which by the definition of N says $D^\top \mathbf{q}(s) \in N$, which by the second fact above says $\Pi_G D^\top \mathbf{q}(s) = 0$, for every reachable query $\mathbf{q}(s)$. Since the reachable queries span the range of Π_Q and every vector of that range is a linear combination of them, $\Pi_G D^\top \mathbf{a} = 0$ for every $\mathbf{a}$ in the range of Π_Q, i.e. $\Pi_G D^\top \Pi_Q = 0$. Transposing, and using that both projectors are symmetric, $\Pi_Q D \Pi_G = 0$, that is, $\Pi_Q B_1 \Pi_G = \Pi_Q B_2 \Pi_G$. In words: the difference of two matrices with the same shift lives entirely in the invisible parts, so projecting the invisible parts away makes the two matrices agree. The canonical operator therefore depends only on the shift, not on which B was used to write it.

(c) Minimality. Multiplying by a matrix cannot raise the rank, so $\operatorname{rank} B^\star = \operatorname{rank}(\Pi_Q B \Pi_G) \leq \operatorname{rank} B$ for every B producing the shift, and $B^\star$ itself produces the shift by (a), so the minimum rank over all such B is attained at $B^\star$ and equals $\operatorname{rank} B^\star$. In words: any matrix that produces the shift carries at least the directions that $B^\star$ carries, and $B^\star$ carries nothing else.

The constants κ and γ. *Idea.* Both constants are minima of a positive continuous function over a unit sphere, so the only thing to check is positivity: that no unit vector of the relevant subspace is invisible. For κ this is automatic once the invisible directions N have been projected away. For γ it is a genuine assumption about the tasks.

$\kappa > 0$. Let $\mathbf{w}$ be a unit vector in the range of Π_G, i.e. in $N^\perp$. If $\operatorname{osc}_v(\mathbf{w}^\top \mathbf{g}_v) = 0$ then $\mathbf{w}^\top \mathbf{g}_v$ is constant in v, so $\mathbf{w} \in N$. But $N \cap N^\perp = \{0\}$ and $\mathbf{w} \neq 0$, a contradiction. So $\mathbf{w} \mapsto \operatorname{osc}_v(\mathbf{w}^\top \mathbf{g}_v)$ is a strictly positive function on the unit sphere of $N^\perp$. It is continuous, being a maximum minus a minimum of finitely many linear functions, and the unit sphere of a finite-dimensional space is compact, so the function attains a positive minimum, which is κ. In words: κ is the guaranteed spread across tokens per unit of visible hidden displacement. The lower bound of the theorem will pay this factor when it converts a hidden vector into logits. (If all $\mathbf{g}_v$ coincide then $N^\perp = \{0\}$, every bilinear shift is constant in v, $B^\star = 0$, and Theorem 3 is trivially true, and we exclude this degenerate case.)

$\gamma > 0$ *is the spanning condition.* For a unit $\mathbf{u}$ in the range of Π_Q, $\max_x |\mathbf{u}^\top \mathbf{q}(x)| = 0$ exactly when $\mathbf{u}$ is orthogonal to every first-token query. So $\gamma = 0$ if and only if some nonzero vector of the reachable query space is orthogonal to all first-token queries, if and only if the first-token queries fail to span the reachable query space. Conversely, when they span it, $\mathbf{u} \mapsto \max_x |\mathbf{u}^\top \mathbf{q}(x)|$ is positive and continuous on the unit sphere of the range of Π_Q, hence attains a positive minimum γ. In words: γ is the guaranteed overlap between any direction in query space and the best first-token query. The lower bound of the theorem will pay this factor when it converts a direction of the error matrix into a task that actually exercises it. When $T = 1$ every state is a first-token state, so the first-token queries are all the reachable queries and the condition holds automatically.

C.5 Theorem 3

The objects. The theorem is about the frontier H_n and the policy frontier E_n^{KV}, under the decomposition $\delta_{\theta'}(s, \cdot) \equiv \mathbf{q}(s)^\top B\mathbf{g}_\cdot + \nu(s, \cdot)$ of Assumption 2, with $\bar{\nu} = \max_s \operatorname{osc}_s \nu$, and the constants g, κ, γ of Assumption 13. Its statements involve the singular values $\sigma_1 \geq \sigma_2 \geq \cdots$ of $B^\star$, with $\sigma_{n+1}(B^\star) = 0$ when $n \geq \operatorname{rank} B^\star$, and, in (iii), the quantity $\Delta_n = [\gamma\kappa\sigma_{n+1}(B^\star) - \bar{\nu}]_+$ where $[a]_+ = \max\{a, 0\}$. Three elementary properties of the oscillation from Appendix A are used repeatedly. Subadditivity is $\operatorname{osc}_s(f + h) \leq \operatorname{osc}_s f + \operatorname{osc}_s h$, symmetry is $\operatorname{osc}_s(-f) = \operatorname{osc}_s f$, and homogeneity is $\operatorname{osc}_s(cf) = |c| \operatorname{osc}_s f$. Subadditivity and symmetry together give the *reverse triangle inequality* $\operatorname{osc}_s(f + h) \geq \operatorname{osc}_s f - \operatorname{osc}_s h$, obtained by writing $f = (f + h) + (-h)$ and applying subadditivity to that sum. For every residual we write $\delta_P - \delta_{\theta'}$. By (4.4) and Assumption 2, at every reachable state

$$(\delta_P - \delta_{\theta'})(s, \cdot) \equiv \mathbf{q}(s)^\top (B_P - B)\mathbf{g}_\cdot - \nu(s, \cdot). \tag{C.1}$$

In words: the residual of a prefix is a bilinear term, driven by the difference between the prefix matrix and the update's bilinear part, minus the part of the update that is not bilinear at all.

Idea of the proof. A key–value prefix of length n is exactly a matrix of rank at most n, so the question of how well prefixes of length n can reproduce the update is the question of how well a rank-n matrix can approximate the bilinear part of the update, and that is a question about singular values: the singular value σ_{n+1} of a matrix is precisely the operator-norm distance from that matrix to the nearest matrix of rank at most n (the Eckart–Young–Mirsky theorem), so it gives both an upper bound, by exhibiting the nearest matrix, and a lower bound, by saying that nothing of rank n comes closer. Two conversions connect matrix distance to oscillation of shifts. Going up (part (ii)), an operator-norm distance between matrices becomes an oscillation bound at a cost $2g$, because a hidden displacement of length ℓ moves each logit by at most $g\ell$. Going down (part (iii)), a singular direction of the error matrix must first be found by some first-token query, at a cost γ, and then made visible in the logits, at a cost κ. The non-bilinear residual $\bar{\nu}$ is carried through by subadditivity in one direction and by the reverse triangle inequality in the other. Part (i) is the exact case, where the rank of the canonical operator is the exact number of prefix positions needed. It needs one technical preliminary, that the infimum defining H_n is actually attained, because "$H_n = 0$" must be turned into "some matrix realizes the update exactly".

Proof of Theorem 3. Preliminary: the infimum defining H_n is attained. Statement (i) reads "$H_n = 0$ if and only if…", and its "only if" direction needs a matrix that actually achieves the infimum, not merely a sequence approaching it: an infimum of zero could in principle be approached by matrices of rank n whose entries grow without bound, with no rank-n matrix achieving zero. We show that this cannot happen, so the minimum exists. Write $F(B') = \max_s \operatorname{osc}_s(\mathbf{q}(s)^\top B'\mathbf{g}_\cdot - \delta_{\theta'}(s,\cdot))$ for the objective, a maximum over the finitely many reachable states. By (a) of the previous subsection applied to B', replacing B' by $\Pi_Q B'\Pi_G$ does not change any $\mathbf{q}(s)^\top B'\mathbf{g}_\cdot$ up to an additive constant, hence does not change $F(B')$, and it does not raise the rank, so H_n is also the infimum of F over the set $\mathcal{M}_n = \{B' : \operatorname{rank} B' \le n,\ B' = \Pi_Q B'\Pi_G\}$. This set is closed, because rank at most n is a closed condition (all minors of order $n+1$ vanish, finitely many polynomial equations) and so is the linear condition $B' = \Pi_Q B'\Pi_G$. The function F is continuous, being a maximum of finitely many oscillations of functions that depend linearly on B'. It remains to show that F is coercive on $\mathcal{M}_n$, meaning that $F(B')$ grows without bound as B' does. Then a minimizing sequence stays bounded and, by compactness of closed bounded sets in finite dimension, has a limit point in $\mathcal{M}_n$ at which the minimum is attained. For $B' \in \mathcal{M}_n$ and a reachable state s, the vector $\mathbf{u}_s = B'^\top \mathbf{q}(s)$ lies in the range of Π_G (because $B'^\top = \Pi_G B'^\top \Pi_Q$), so by readout visibility and homogeneity $\operatorname{osc}_s(\mathbf{q}(s)^\top B'\mathbf{g}_\cdot) = \operatorname{osc}_v(\mathbf{u}_s^\top \mathbf{g}_v) \ge \kappa\|\mathbf{u}_s\|_2$, and by the reverse triangle inequality

$$F(B') \ge \max_s \left(\kappa\|B'^\top \mathbf{q}(s)\|_2 - \operatorname{osc}_s \delta_{\theta'}\right) \ge \kappa \max_s \|B'^\top \mathbf{q}(s)\|_2 - \max_s \operatorname{osc}_s \delta_{\theta'}.$$

In words: a large matrix in $\mathcal{M}_n$ produces a large visible displacement at some state, hence a large oscillation, which the fixed target cannot cancel. The map $B' \mapsto \max_s \|B'^\top \mathbf{q}(s)\|_2$ is a norm on the linear space $\{B' : B' = \Pi_Q B'\Pi_G\}$: it is a maximum of seminorms, and it vanishes only if $B'^\top \mathbf{q}(s) = 0$ for every reachable query, i.e. $B'^\top \Pi_Q = 0$, i.e. $\Pi_Q B' = 0$, which for $B' = \Pi_Q B'\Pi_G$ means $B' = 0$. All norms on a finite-dimensional space are equivalent, so this norm is at least a positive constant times $\|B'\|_{\mathrm{op}}$, and $F(B') \to \infty$ as $\|B'\|_{\mathrm{op}} \to \infty$ within $\mathcal{M}_n$. Hence the infimum is a minimum: $H_n = F(B_n)$ for some B_n of rank at most n. (If no query is nonzero, so that $\Pi_Q = 0$, then $\mathcal{M}_n = \{0\}$, F is constant on it, and attainment is immediate. The constants γ and κ, which are minima over unit spheres, are then not defined and are not used, since Assumption 13 presumes a nonzero query space.)

(i) Exactly bilinear updates. Assume $\bar{\nu} = 0$, so that $\operatorname{osc}_s \nu = 0$ at every reachable state, i.e. $\nu(s,\cdot)$ is constant in v and $\delta_{\theta'}(s,\cdot) \equiv \mathbf{q}(s)^\top B\mathbf{g}_\cdot$ at every reachable state: the update is exactly prefix-shaped.

If $n \ge \operatorname{rank} B^\star$, then $H_n = 0$. By (a), $B^\star$ gives the same shift as B, hence $\mathbf{q}(s)^\top B^\star \mathbf{g}_\cdot \equiv \delta_{\theta'}(s,\cdot)$ at every reachable state, so $F(B^\star) = 0$, and $\operatorname{rank} B^\star \le n$ makes $B^\star$ admissible in the infimum. Hence $H_n = 0$. Concretely, a rank factorization of $B^\star$ padded to length n, as in the previous subsection, is a key–value prefix of length n whose shift is $\delta_{\theta'}$ exactly, and a factorization of length $\operatorname{rank} B^\star$ suffices.

If $H_n = 0$, then $n \ge \operatorname{rank} B^\star$. By the preliminary, $H_n = 0$ means $F(B_n) = 0$ for some B_n of rank at most n, that is, $\mathbf{q}(s)^\top B_n \mathbf{g}_\cdot \equiv \delta_{\theta'}(s,\cdot) \equiv \mathbf{q}(s)^\top B\mathbf{g}_\cdot$ at every reachable state. So B_n and B give the same shift, and by (b) they have the same canonical operator: $\Pi_Q B_n \Pi_G = \Pi_Q B \Pi_G = B^\star$. By (c), $\operatorname{rank} B^\star \le \operatorname{rank} B_n \le n$. In words: any matrix that reproduces the update exactly must carry all the directions of $B^\star$, and a prefix of length n carries at most n.

Together, $H_n = 0$ if and only if $n \geq \operatorname{rank} B^\star$, so the effective rank $r_{\text{eff}} = \min\{n : H_n = 0\}$ equals $\operatorname{rank} B^\star$.

(ii) The upper bound. Fix any matrix B_P (of any rank, for the moment) and a reachable state s. By (C.1) the residual is $\mathbf{q}(s)^\top (B_P - B)\mathbf{g}_\cdot - \nu(s,\cdot)$. Apply (a) to the matrix $B_P - B$: its bilinear shift equals that of $\Pi_Q(B_P - B)\Pi_G = \Pi_Q B_P \Pi_G - B^\star$ up to an additive constant, and applying (a) once more, this time to $B_P - B^\star$, the shift of $\Pi_Q(B_P - B^\star)\Pi_G = \Pi_Q B_P \Pi_G - B^\star$ is the same matrix again (because $\Pi_Q B^\star \Pi_G = B^\star$, the projectors being idempotent). Hence

$$\mathbf{q}(s)^\top (B_P - B)\mathbf{g}_v \equiv \mathbf{q}(s)^\top (B_P - B^\star)\mathbf{g}_v = \mathbf{u}^\top \mathbf{g}_v, \qquad \mathbf{u} := (B_P - B^\star)^\top \mathbf{q}(s) \in \mathbb{R}^d.$$

In words: replacing B by its canonical operator does not change the residual, and the bilinear residual at state s is the logit pattern produced by the hidden displacement $\mathbf{u}$, the difference between what the prefix matrix and the canonical operator do to the query. We now bound the oscillation of $\mathbf{u}^\top \mathbf{g}_v$ by a chain of three inequalities:

$$\begin{aligned}
\operatorname{osc}_v(\mathbf{u}^\top \mathbf{g}_v) &\leq 2 \max_v |\mathbf{u}^\top \mathbf{g}_v| && \text{(spread versus largest absolute value)} \\
&\leq 2\|\mathbf{u}\|_2 \max_v \|\mathbf{g}_v\|_2 = 2g\|\mathbf{u}\|_2 && \text{(Cauchy–Schwarz, definition of } g) \\
&\leq 2g\|B_P - B^\star\|_{\text{op}} \|\mathbf{q}(s)\|_2 && \text{(operator norm)} \\
&\leq 2g\|B_P - B^\star\|_{\text{op}} && (\|\mathbf{q}(s)\|_2 \leq 1).
\end{aligned}$$

The first inequality holds because $\max_v f(v) - \min_v f(v) \leq |\max_v f(v)| + |\min_v f(v)| \leq 2\max_v |f(v)|$: a spread is at most twice the largest excursion from zero. The second is the Cauchy–Schwarz inequality $|\mathbf{u}^\top \mathbf{g}_v| \leq \|\mathbf{u}\|_2 \|\mathbf{g}_v\|_2$ applied at the maximizing token. The third is the defining property of the operator norm, $\|A\mathbf{a}\|_2 \leq \|A\|_{\text{op}} \|\mathbf{a}\|_2$, together with the fact that a matrix and its transpose have the same operator norm, so $\|(B_P - B^\star)^\top \mathbf{q}(s)\|_2 \leq \|B_P - B^\star\|_{\text{op}} \|\mathbf{q}(s)\|_2$. The fourth is Assumption 13. Adding the non-bilinear residual by subadditivity and symmetry, $\operatorname{osc}_s(\mathbf{u}^\top \mathbf{g}_\cdot - \nu(s,\cdot)) \leq \operatorname{osc}_v(\mathbf{u}^\top \mathbf{g}_v) + \operatorname{osc}_s \nu \leq 2g\|B_P - B^\star\|_{\text{op}} + \bar{\nu}$, at every reachable state. This holds for *every* matrix B_P. Nothing has been chosen yet.

Now choose B_P. The Eckart–Young–Mirsky theorem (Eckart & Young, 1936; Mirsky, 1960) states that for any matrix A and any matrix A' of rank at most n, $\|A - A'\|_{\text{op}} \geq \sigma_{n+1}(A)$, with equality when A' is the rank-n truncation of the singular value decomposition of A, the matrix obtained by keeping the n largest singular values and setting the rest to zero. In words: $\sigma_{n+1}(A)$ is exactly the distance from A to the set of matrices of rank at most n, and the truncation is a nearest point. Let B_P be the rank-n truncation of $B^\star$. It has rank at most n, so it is realized by a key–value prefix of length n (previous subsection) and is admissible in the infimum defining H_n, and $\|B_P - B^\star\|_{\text{op}} = \sigma_{n+1}(B^\star)$. Therefore

$$H_n \leq \max_s \operatorname{osc}_s \left(\mathbf{q}(s)^\top B_P \mathbf{g}_\cdot - \delta_{\theta'}(s,\cdot)\right) \leq 2g\,\sigma_{n+1}(B^\star) + \bar{\nu}.$$

The first inequality is the definition of an infimum (one admissible matrix bounds it from above). The second is the chain just proved.

For the policy statement, the prefix realizing this B_P has compilation error $d(P) \leq \frac{T}{4} \max_s \operatorname{osc}_s(\delta_P - \delta_{\theta'})$ by the right-hand form of (3.4), and the maximum is what was just bounded. Since E_n^{KV} is an infimum over all prefixes of length n, $E_n^{\text{KV}} \leq d(P) \leq \frac{T}{4}\big(2g\sigma_{n+1}(B^\star) + \bar{\nu}\big)$. The same argument applied to an arbitrary rank-n matrix also gives the auxiliary bound $E_n^{\text{KV}} \leq \frac{T}{4} H_n$.

The same two displays hold with the root-sum-square bridge of Proposition 11 in place of the right-hand form of (3.4). That proposition bounds $d(S)$ by $\min\{1, \frac{\sqrt{T}}{4} \max_s \operatorname{osc}_s(\delta_S - \delta_{\theta'})\}$, and the maximum is again what was just bounded, so

$$E_n^{\text{KV}} \leq \min\left\{1, \tfrac{\sqrt{T}}{4}\big(2g\,\sigma_{n+1}(B^\star) + \bar{\nu}\big)\right\}, \qquad E_n^{\text{KV}} \leq \min\left\{1, \tfrac{\sqrt{T}}{4} H_n\right\}.$$

Both forms are valid and one may take the smaller. For a uniform residual bound the root-sum-square form is never worse, since $\sqrt{T} \leq T$, and for long responses it is much better: at $T = 500$ and a residual oscillation of 0.01 the linear form gives the vacuous 1.25 while this one gives 0.056. The linear form is retained because it is the one that survives when only a per-step bound on the oscillation, and not its square, is available.

(iii) The lower bound. Fix an arbitrary key–value prefix P of length n, so that $\operatorname{rank} B_P \le n$, and define the error matrix

$$E = \Pi_Q(B^\star - B_P)\Pi_G = B^\star - \Pi_Q B_P \Pi_G \in \mathbb{R}^{d\times d},$$

the second form because $\Pi_Q B^\star \Pi_G = B^\star$. In words: E is the visible part of the gap between what the update needs and what the prefix supplies. The matrix $\Pi_Q B_P \Pi_G$ has rank at most $\operatorname{rank} B_P \le n$, so by the Eckart–Young–Mirsky theorem applied to $A = B^\star$ and $A' = \Pi_Q B_P \Pi_G$,

$$\sigma := \|E\|_{\mathrm{op}} \ge \sigma_{n+1}(B^\star).$$

This is the lower-bound half of the theorem: no matrix of rank at most n, whatever prefix produced it, gets closer to $B^\star$ than $\sigma_{n+1}(B^\star)$. If $\sigma = 0$ then $\sigma_{n+1}(B^\star) = 0$, $\Delta_n = 0$, and the claim "some task has residual oscillation at least Δ_n" is trivially true, so assume $\sigma > 0$. The remaining work is to turn a matrix gap of size σ into an oscillation at an actual task, and each of the next four steps is one conversion.

Step 1: the top singular pair of E lives in the two ranges. Let $\boldsymbol{\xi}, \boldsymbol{\psi} \in \mathbb{R}^d$ be unit vectors with $E\boldsymbol{\psi} = \sigma\boldsymbol{\xi}$ and $E^\top\boldsymbol{\xi} = \sigma\boldsymbol{\psi}$, a top left and right singular pair of E: $\boldsymbol{\psi}$ is a unit direction that E stretches the most, and $\boldsymbol{\xi}$ is where it sends it. Since $E = \Pi_Q E$ (the projector on the left is idempotent), $\boldsymbol{\xi} = E\boldsymbol{\psi}/\sigma = \Pi_Q(E\boldsymbol{\psi}/\sigma)$ lies in the range of Π_Q. Since $E = E\Pi_G$, $E^\top = \Pi_G E^\top$ and $\boldsymbol{\psi} = E^\top\boldsymbol{\xi}/\sigma$ lies in the range of Π_G. In words: because E was built with the invisible parts projected away, its most-stretched direction is a visible logit direction and its image is a direction that queries can reach. This is what lets γ and κ apply.

Step 2: some task sees the top direction. By query richness, γ is the minimum over unit vectors in the range of Π_Q of $\max_x |\mathbf{u}^\top \mathbf{q}(x)|$, and $\boldsymbol{\xi}$ is such a vector, so $\max_x |\boldsymbol{\xi}^\top \mathbf{q}(x)| \ge \gamma$: there is a task x whose first-token query satisfies $|\boldsymbol{\xi}^\top \mathbf{q}(x)| \ge \gamma$. Fix this task. This is the step that pays γ: the direction $\boldsymbol{\xi}$ is only guaranteed to be exercised by some task up to that overlap.

Step 3: the hidden residual at that task is long. Put $\mathbf{u} = E^\top \mathbf{q}(x) \in \mathbb{R}^d$, the hidden displacement that the error matrix produces at the chosen query. Since $E^\top = \Pi_G E^\top$, $\mathbf{u}$ lies in the range of Π_G. Its length is bounded below by its component along the unit vector $\boldsymbol{\psi}$:

$$\|\mathbf{u}\|_2 \;\ge\; |\boldsymbol{\psi}^\top \mathbf{u}| \;=\; |\boldsymbol{\psi}^\top E^\top \mathbf{q}(x)| \;=\; |(E\boldsymbol{\psi})^\top \mathbf{q}(x)| \;=\; \sigma\,|\boldsymbol{\xi}^\top \mathbf{q}(x)| \;\ge\; \gamma\sigma \;\ge\; \gamma\,\sigma_{n+1}(B^\star),$$

the first step being the Cauchy–Schwarz inequality $|\boldsymbol{\psi}^\top \mathbf{u}| \le \|\boldsymbol{\psi}\|_2 \|\mathbf{u}\|_2 = \|\mathbf{u}\|_2$, the second the definition of $\mathbf{u}$, the third the identity $\boldsymbol{\psi}^\top E^\top = (E\boldsymbol{\psi})^\top$, the fourth $E\boldsymbol{\psi} = \sigma\boldsymbol{\xi}$, the fifth Step 2, and the last the Eckart–Young–Mirsky bound on σ.

Step 4: the logits see it. Readout visibility says $\operatorname{osc}_v(\mathbf{w}^\top \mathbf{g}_v) \ge \kappa$ for every unit $\mathbf{w}$ in the range of Π_G. Apply it to $\mathbf{w} = \mathbf{u}/\|\mathbf{u}\|_2$ (well defined, as $\|\mathbf{u}\|_2 \ge \gamma\sigma > 0$) and use homogeneity:

$$\operatorname{osc}_v(\mathbf{u}^\top \mathbf{g}_v) = \|\mathbf{u}\|_2 \operatorname{osc}_v\left((\mathbf{u}/\|\mathbf{u}\|_2)^\top \mathbf{g}_v\right) \ge \kappa\|\mathbf{u}\|_2 \ge \gamma\kappa\,\sigma_{n+1}(B^\star).$$

This is the step that pays κ. So γ is the loss in passing from a matrix direction to a query some task actually produces, and κ the loss in passing from a hidden vector to the logits. They multiply because the two losses are incurred in sequence.

Step 5: the residual at the first token of x. At the state $s = (x, \emptyset)$, by (C.1) and the same two applications of (a) as in part (ii),

$$\begin{aligned}(\delta_P - \delta_{\theta'})(s,\cdot) &\equiv \mathbf{q}(x)^\top (B_P - B^\star)\mathbf{g}_\cdot - \nu(s,\cdot) \equiv \mathbf{q}(x)^\top \Pi_Q (B_P - B^\star)\Pi_G\,\mathbf{g}_\cdot - \nu(s,\cdot)\\ &= -\mathbf{q}(x)^\top E\,\mathbf{g}_\cdot - \nu(s,\cdot) = -\mathbf{u}^\top \mathbf{g}_\cdot - \nu(s,\cdot),\end{aligned}$$

using $\Pi_Q(B_P - B^\star)\Pi_G = -E$ and $\mathbf{q}(x)^\top E = (E^\top \mathbf{q}(x))^\top = \mathbf{u}^\top$. In words: at the chosen task, the prefix's residual is the logit pattern of $\mathbf{u}$ (with a sign, which the oscillation ignores) contaminated by the non-bilinear part ν. By the reverse triangle inequality and symmetry of the oscillation,

$$\operatorname{osc}_s(\delta_P - \delta_{\theta'}) \ge \operatorname{osc}_v(-\mathbf{u}^\top \mathbf{g}_v) - \operatorname{osc}_s \nu = \operatorname{osc}_v(\mathbf{u}^\top \mathbf{g}_v) - \operatorname{osc}_s \nu \ge \gamma\kappa\,\sigma_{n+1}(B^\star) - \bar{\nu}.$$

The reverse triangle inequality is needed here, rather than subadditivity, because we want a *lower* bound and the contamination ν could partly cancel the bilinear residual. The most it can cancel is its own oscillation, at most $\bar{\nu}$. An oscillation is never negative, so the bound can be stated as

$\operatorname{osc}_s(\delta_P - \delta_{\theta'}) \geq [\gamma\kappa\sigma_{n+1}(B^\star) - \bar{\nu}]_+ = \Delta_n$. The positive part only removes the vacuous case. This is the first claim of (iii). Since P was an arbitrary prefix of length n, i.e. B_P an arbitrary matrix of rank at most n, it also shows $H_n \geq \Delta_n$: every admissible matrix has some state with residual oscillation at least Δ_n, so the maximum over states is at least Δ_n, and so is the infimum of that maximum.

Step 6: from oscillation to policy error, through Lemma 12. Both shifts are log-ratios against the same base distribution, so their difference is a log-ratio between the two policies with no constant at all:

$$(\delta_P - \delta_{\theta'})(s, v) = \log \frac{\pi_{\theta,P}(v \mid s)}{\pi_\theta(v \mid s)} - \log \frac{\pi_{\theta'}(v \mid s)}{\pi_\theta(v \mid s)} = \log \frac{p(v)}{q(v)}, \qquad p = \pi_{\theta,P}(\cdot \mid s),\ q = \pi_{\theta'}(\cdot \mid s).$$

Let a and b be tokens at which the residual attains its maximum and its minimum. Then $\log \frac{p(a)}{q(a)} - \log \frac{p(b)}{q(b)} = \operatorname{osc}_s(\delta_P - \delta_{\theta'}) \geq \Delta_n$. Any additive constant in the residual would cancel in this difference, which is exactly why the hypothesis of Lemma 12 is a difference of two log-ratios. If $q(a) \geq m$ and $q(b) \geq m$, that is, if the two tokens each have aligned-model probability at least m at the first token of x, Lemma 12 gives $\mathrm{TV}(p, q) \geq m(1 - e^{-\Delta_n/2})$, and its first-token statement gives $d(P) \geq D(x)\ \mathrm{TV}(p, q) \geq D(x)\, m(1 - e^{-\Delta_n/2})$. This is the second claim of (iii). Why the first token? A lower bound on the response distribution needs the discrepant state to be reached. The first-token state is reached with probability one, whereas a later state would carry the probability of reaching it as an extra factor.

A prefix-independent form. The mass condition involves only the aligned model. Let m be the smallest aligned first-token probability over all tasks and tokens, $m = \min_x \min_v \pi_{\theta'}(v \mid x, \emptyset)$. Then for every prefix P of length n the task x selected in Step 2 satisfies the mass condition, and $d(P) \geq D(x)\, m(1 - e^{-\Delta_n/2}) \geq \min_{x'} D(x')\, m(1 - e^{-\Delta_n/2})$. Here Δ_n does not depend on P, so taking the infimum over prefixes, $E_n^{\mathrm{KV}} \geq \min_x D(x)\, m(1 - e^{-\Delta_n/2})$, the bound quoted after the theorem.

Here m must be the minimum over the *whole* vocabulary. It is tempting to use instead the smallest aligned mass on the restricted token set of Proposition 1, since the upper bound tolerates that restriction. This is invalid, because the tokens that witness the oscillation may be exactly the ones the restriction drops. Three tokens suffice to see it: with $q = (0.5, 0.499, 0.001)$ and $p = (0.5, 0.498, 0.002)$ the total variation is 0.001, while the oscillation of $\log(p/q)$ is 0.695, attained at the rare third token. Retaining only the first two tokens and substituting their smallest mass 0.499 into the bound would assert $0.499(1 - e^{-0.695/2}) \approx 0.147 \leq \mathrm{TV}(p, q) = 0.001$, which is false. The full-vocabulary minimum 0.001 gives 0.00029, which is correct. With state-dependent shortlists the witnessing geometry is itself state-dependent, so there is no cheap repair: either keep the full-vocabulary mass, or prove separately that the witnessing tokens survive the restriction.

Longer responses. For $T > 1$, part (ii) can be stated with the middle form of (3.4) instead of the right-hand form, exactly as in Appendix D.1: replace $\max_s$ by the expectation along aligned-model trajectories of $\operatorname{osc}_{s_t}(\mathbf{q}(s_t)^\top (B_P - B^\star)\mathbf{g}_\cdot) + \operatorname{osc}_{s_t} \nu$, and $\bar{\nu}$ by the corresponding expected residual. Part (iii) is stated at first-token states and needs no change.

■

↑ Back to page 5

Machine check. All three parts of the theorem are formalized in Lean 4, in shift space and in policy form, together with the Eckart–Young–Mirsky theorem they rest on.

C.6 A WEIGHT UPDATE THAT IS EXACTLY COMPILABLE

Part (i) of the theorem describes an exact regime, and it is fair to ask whether that regime is empty for genuine weight updates. It is not. Take a scalar head, $d = d_{\mathrm{in}} = 1$, a vocabulary of two tokens, $V = 2$, and horizon $T = 1$. Let the readout be $G = (0, 1) \in \mathbb{R}^{1\times 2}$, so $\mathbf{g}_1 = 0$ and $\mathbf{g}_2 = 1$, and let $W_K = W_V = (1)$. Two tasks are single tokens with embeddings $c \in \{-1, +1\}$, and the query at each is $\mathbf{q} = (c)$. Now change the value map by $\Lambda_V = (1)$, that is, $W_V = 1 \mapsto 2$: a real weight update, not a prefix in disguise.

By Proposition 4 the update acts through $B_s = W_K^\top C_s^\top C_s \Lambda_V$. Here $C_s = (c)$, so the content Gram matrix is $C_s^\top C_s = c^2 = 1$ at *both* tasks, and hence $B_s = 1$ at both: the state dependence that the proposition warns about is present in the content but invisible after the Gram matrix, because the two embeddings differ only by a sign that squaring removes. The update is therefore

$$\delta_{\theta'}(s, v) \equiv \mathbf{q}^\top B_s \mathbf{g}_v = c\, \mathbf{g}_v,$$

which the single key–value pair $P_K = P_V = (1)$, giving $B_P = P_K^\top P_V = 1$, reproduces exactly at both tasks. The two policies are genuinely different: the base logits are $z(s, \cdot) = (0, c)$ and the aligned ones $(0, 2c)$, so at $c = +1$ the base distribution is $(0.269, 0.731)$ and the aligned one $(0.119, 0.881)$, a total variation of 0.150, and both are strictly positive. The canonical operator is $B^\star = 1$ of rank one, and $r_{\text{eff}} = 1$: one prefix position is necessary and sufficient.

The lesson is the one the general statement should make. It is not true that no weight update is exactly bilinear. What is true is that an ordinary weight update *need not* have a state-independent bilinear representation, and Proposition 4 identifies content dependence as the obstruction when it does not. That obstruction is real but not universal, and it is invisible whenever, as here, the content enters only through a Gram matrix that the tasks share.

C.7 Proposition 4

The objects. The aligned model θ' has the same head as θ except that its value map is $W_V + \Lambda_V$ with $\Lambda_V \in \mathbb{R}^{d_{\text{in}} \times d}$. Queries, keys and the readout G are unchanged. The proposition defines $B_s = W_K^\top C_s^\top C_s \Lambda_V \in \mathbb{R}^{d \times d}$, where $C_s^\top C_s = \sum_{i \le |s|} \mathbf{c}_i \mathbf{c}_i^\top \in \mathbb{R}^{d_{\text{in}} \times d_{\text{in}}}$ is the Gram matrix of the content embeddings, the matrix that records which directions the content occupies and how strongly. A fixed key–value prefix P with matrix B_P and two tasks x_1, x_2 are given. Here $\mathbf{q}$ denotes their common first-token query, and $[\delta_P - \delta_{\theta'}]_{x_i}$ the residual at the first token of task x_i, a function of v. In the second part, $\mathbf{q}_1 \neq \mathbf{q}_2$ are two different first-token queries.

Idea of the proof. Changing the value map changes the head output by a bilinear term, just as a prefix does, but its matrix B_s contains the Gram matrix of the content, so it is different at every state. A fixed prefix contributes the same B_P everywhere. If two tasks present the same query, the prefix term is literally the same function of v at both, so subtracting the two residuals cancels it and leaves only the difference of the two content terms. The oscillation of that difference is then at most the sum of the two residual oscillations, so at least one residual is at least half of it. When the queries differ, the prefix terms no longer cancel exactly, and the leftover is bounded by the size of the prefix times the distance between the queries.

Proof of Proposition 4. Step 1: the update of a value change is bilinear with a state-dependent core. With values $C_s(W_V + \Lambda_V)$ in place of $C_s W_V$, the head output of θ' is, by the formula of the first subsection,

$$\begin{aligned}\mathbf{o}^\star(s)^\top &= \mathbf{q}(s)^\top (C_s W_K)^\top C_s (W_V + \Lambda_V) \\ &= \mathbf{q}(s)^\top W_K^\top C_s^\top C_s W_V + \mathbf{q}(s)^\top W_K^\top C_s^\top C_s \Lambda_V = \mathbf{o}(s)^\top + \mathbf{q}(s)^\top B_s,\end{aligned}$$

by distributing the product over the sum. This is an identity. Logits are linear in the head output, so $z^\star(s, v) = z(s, v) + \mathbf{q}(s)^\top B_s \mathbf{g}_v$, and the update, the difference of log-probabilities, is $\delta_{\theta'}(s, v) = z^\star(s, v) - z(s, v) - c(s)$ with $c(s)$ the difference of the two log-normalizers, exactly as for a prefix in Step 2 of the prefix subsection. Hence $\delta_{\theta'}(s, \cdot) \equiv \mathbf{q}(s)^\top B_s \mathbf{g}_\cdot$. In words: a value update looks exactly like a prefix at any one state. The difference is that the core B_s moves with the state through the Gram matrix $C_s^\top C_s$, whereas B_P is the same everywhere.

Step 2: when two tasks share a query. In a one-layer model the query is $\mathbf{q}(s) = W_Q^\top \mathbf{c}_{\text{last}}$, a function of the last token's embedding alone, so two tasks ending in the same token share their first-token query. If positional information is added to the embedding, the two tasks must also have the same length, and nothing else changes. Let x_1, x_2 be two such tasks with common query $\mathbf{q}$. At the first token of x_i, by (4.4) and Step 1,

$$[\delta_P - \delta_{\theta'}]_{x_i}(v) \equiv \mathbf{q}^\top B_P \mathbf{g}_v - \mathbf{q}^\top B_{x_i} \mathbf{g}_v,$$

where B_{x_i} is the core at the first-token state of x_i. Subtracting the residual at x_1 from the residual at x_2, the prefix term $\mathbf{q}^\top B_P \mathbf{g}_v$ is identical in both and cancels:

$$[\delta_P - \delta_{\theta'}]_{x_2} - [\delta_P - \delta_{\theta'}]_{x_1} \equiv \mathbf{q}^\top (B_{x_1} - B_{x_2}) \mathbf{g}_\cdot.$$

This is an identity between functions of v (up to an additive constant), and it is the whole point: the right-hand side does not involve the prefix at all, so it is a quantity the prefix cannot influence. Taking oscillations, and using subadditivity together with $\operatorname{osc}(-f) = \operatorname{osc} f$,

$$\operatorname{osc}_v \left(\mathbf{q}^\top (B_{x_1} - B_{x_2}) \mathbf{g}_v\right) \le \operatorname{osc}[\delta_P - \delta_{\theta'}]_{x_2} + \operatorname{osc}[\delta_P - \delta_{\theta'}]_{x_1} \le 2 \max_{i \in \{1,2\}} \operatorname{osc}[\delta_P - \delta_{\theta'}]_{x_i}.$$

The second inequality replaces each of two nonnegative terms by the larger of the two. Dividing by two gives (4.6). Nothing about B_P was used beyond its being the same at both tasks, so the floor holds for every fixed prefix of every length.

Step 3: when the queries differ. If the two tasks have different queries $\mathbf{q}_1 \neq \mathbf{q}_2$, the same subtraction leaves a prefix term behind:

$$[\delta_P - \delta_{\theta'}]_{x_2} - [\delta_P - \delta_{\theta'}]_{x_1} \equiv (\mathbf{q}_2 - \mathbf{q}_1)^\top B_P \mathbf{g}_\cdot + \left(\mathbf{q}_1^\top B_{x_1} \mathbf{g}_\cdot - \mathbf{q}_2^\top B_{x_2} \mathbf{g}_\cdot\right).$$

Rearranging so that the content term stands alone and applying subadditivity to the three remaining terms,

$$\operatorname{osc}_v \left(\mathbf{q}_1^\top B_{x_1} \mathbf{g}_v - \mathbf{q}_2^\top B_{x_2} \mathbf{g}_v\right) \le 2 \max_i \operatorname{osc}[\delta_P - \delta_{\theta'}]_{x_i} + \operatorname{osc}_v \left((\mathbf{q}_2 - \mathbf{q}_1)^\top B_P \mathbf{g}_v\right),$$

and by the chain of three inequalities in the proof of Theorem 3(ii), applied to $\mathbf{u} = B_P^\top (\mathbf{q}_2 - \mathbf{q}_1)$, the last oscillation is at most $2g \|B_P\|_{\mathrm{op}} \|\mathbf{q}_1 - \mathbf{q}_2\|_2$. Dividing by two, the right-hand side of (4.6) is reduced by $g \|B_P\|_{\mathrm{op}} \|\mathbf{q}_1 - \mathbf{q}_2\|_2$, which is the form quoted after the proposition. In words: a prefix can tell two nearby queries apart only in proportion to how strongly it amplifies their difference, so a prefix of large operator norm can separate nearby tasks. The floor is an exact-collision statement.

Step 4: policy error. The first-token statement of Lemma 12, applied at whichever of the two tasks has the larger residual, converts that residual into a lower bound on $d(P)$ under its mass condition, exactly as in Step 6 of the proof of Theorem 3, and without any factor for reaching the state, because a first-token state is reached with probability one.

■

↑ Back to page 6

The content example. Take $d_{\mathrm{in}} = d = V = 4$, $G = I_4$, $W_K = \Lambda_V = I_4$, and two tasks that share every token, including the last, except one earlier token, embedded as $\mathbf{e}_1$ in x_1 and as $\mathbf{e}_2$ in x_2. The common query is $\mathbf{q} = (1, 1, 0, 0)/\sqrt{2}$, of unit norm. The Gram matrix $C_s^\top C_s = \sum_i \mathbf{c}_i \mathbf{c}_i^\top$ is a sum of one rank-one term per token, so the shared tokens contribute identical terms to B_{x_1} and B_{x_2}, which cancel in the difference, leaving only the two differing tokens:

$$B_{x_1} - B_{x_2} = W_K^\top \left(\mathbf{e}_1 \mathbf{e}_1^\top - \mathbf{e}_2 \mathbf{e}_2^\top\right) \Lambda_V = \operatorname{diag}(1, -1, 0, 0).$$

Then $\mathbf{q}^\top \operatorname{diag}(1, -1, 0, 0)\, G = \frac{1}{\sqrt{2}}(1, -1, 0, 0)$ as a function of v, whose oscillation is $\frac{1}{\sqrt{2}} - (-\frac{1}{\sqrt{2}}) = \sqrt{2} = 1.41$, and (4.6) halves it: every fixed prefix has residual oscillation at least $\sqrt{2}/2 = 0.707$ at one of the two tasks, the decimal rounded *down* because a lower bound rounded up is no longer one. In words: a single earlier token that the two tasks do not share is enough to move the update's core by a fixed amount, and no prefix, however long, can follow that move, because the prefix cannot see which of the two tokens was present.

D PROOFS FOR CONSTITUTIONS

This appendix proves Propositions 6 and 7. Throughout, at a fixed state s a shift is a vector in $\mathbb{R}^V$ indexed by the token, and osc_s is the subadditive, positively homogeneous, constant-annihilating seminorm of Appendix A, in which every statement below is made.

D.1 PROPOSITION 6

The objects. For a selection $R \subseteq \{1, \dots, M\}$, let $\delta_R = \delta_{\mathcal{C}[R]}$ be the shift induced by the rendered selection, let $u_R = \sum_{j \in R} u_j$ be the sum of the declared effects, and define the interaction

$$I_R = \delta_R - \sum_{j \in R} \delta_j.$$

If every rendering carries a common frame whose shift is h, then $I_R \equiv (1 - |R|)h$ even when the clause-specific effects compose exactly, so I_R counts a shared frame $|R| - 1$ times; Appendix F therefore also reports the interaction with the frame counted once. For a task x with reachable states Σ_x, recall

$$\mathcal{A}(x, R) = \max_{s \in \Sigma_x} \operatorname{osc}_s(\delta_{\theta'} - u_R), \quad \eta_j(x) = \max_{s \in \Sigma_x} \operatorname{osc}_s(u_j - \delta_j), \quad \iota(x, R) = \max_{s \in \Sigma_x} \operatorname{osc}_s(I_R).$$

A retriever ρ assigns each task x a selection $\rho(x)$ with $|\rho(x)| \le k$, and

$$d(\rho) = \mathbb{E}_{x \sim D} \operatorname{TV}(\pi_{\theta,R}(y \mid x), \pi_{\theta'}(y \mid x)).$$

Idea. The residual between the rendered context and the target is the sum of three terms: the mismatch between the target and the declared effects, the mismatch between each declared effect and its realization, and the interaction between clauses. Subadditivity of the oscillation bounds these terms separately, and the bridge propositions convert the resulting shift bound into policy error.

Proof of Proposition 6. Step 1: the residual decomposes exactly. Fix a task x and write $R = \rho(x)$. At every state s and token v,

$$\begin{aligned}(u_R - \delta_{\theta'}) + \sum_{j \in R} (\delta_j - u_j) + I_R &= \Big(\sum_{j \in R} u_j - \delta_{\theta'} \Big) + \Big(\sum_{j \in R} \delta_j - \sum_{j \in R} u_j \Big) + \Big(\delta_R - \sum_{j \in R} \delta_j \Big) \\ &= \delta_R - \delta_{\theta'}.\end{aligned}$$

Step 2: bound the residual oscillation. By subadditivity and symmetry of the oscillation, at every $s \in \Sigma_x$,

$$\operatorname{osc}_s(\delta_R - \delta_{\theta'}) \le \operatorname{osc}_s(\delta_{\theta'} - u_R) + \sum_{j \in R} \operatorname{osc}_s(u_j - \delta_j) + \operatorname{osc}_s(I_R).$$

Taking the maximum over $s \in \Sigma_x$ gives

$$\max_{s \in \Sigma_x} \operatorname{osc}_s(\delta_R - \delta_{\theta'}) \le \mathcal{A}(x, R) + \sum_{j \in R} \eta_j(x) + \iota(x, R).$$

Step 3: convert shift error into policy error. Proposition 1, applied task by task with $S = R$, gives

$$\operatorname{TV}(\pi_{\theta,R}(y \mid x), \pi_{\theta'}(y \mid x)) \le \frac{T}{4} \left[\mathcal{A}(x, \rho(x)) + \sum_{j \in \rho(x)} \eta_j(x) + \iota(x, \rho(x)) \right].$$

Averaging over $x \sim D$ gives the $T/4$ form of (5.2). Proposition 11 gives, in the same way,

$$d(\rho) \le \frac{\sqrt{T}}{4} \, \mathbb{E}_{x \sim D} \left[\mathcal{A}(x, \rho(x)) + \sum_{j \in \rho(x)} \eta_j(x) + \iota(x, \rho(x)) \right],$$

which is the stated root form.

Step 4: the rollout form. For $s_t = (x, y_{<t})$, define

$$\begin{aligned}\bar{\mathcal{A}}(x, R) &= \sum_{t=1}^{T} \mathbb{E}_{y_{<t} \sim \pi_{\theta'}} \operatorname{osc}_{s_t}(\delta_{\theta'} - u_R), \\ \bar{\eta}_j(x) &= \sum_{t=1}^{T} \mathbb{E}_{y_{<t} \sim \pi_{\theta'}} \operatorname{osc}_{s_t}(u_j - \delta_j), \\ \bar{\iota}(x, R) &= \sum_{t=1}^{T} \mathbb{E}_{y_{<t} \sim \pi_{\theta'}} \operatorname{osc}_{s_t}(I_R).\end{aligned}$$

The identity of Step 1 and the inequality of Step 2 hold at every state s_t. Applying them inside the trajectory expectations in (3.4), together with $\tanh u \le u$, yields

$$d(\rho) \le \frac{1}{4}\,\mathbb{E}_{x\sim D}\left[\bar{\mathcal{A}}(x,\rho(x)) + \sum_{j\in\rho(x)} \bar{\eta}_j(x) + \bar{\iota}(x,\rho(x))\right].$$

This is the stated rollout form. ■

↑ Back to page 7

D.2 Proposition 7

Proposition 7 is stated in Section 7. The instance data are rational, with finite binary encodings, so that the problem is a decision problem in the standard Turing model. Our reduction uses integer entries only. For on/off selection the budget is not the source of the hardness: the exact-cover requirement sits in the residual objective, and without a budget the problem is still exact cover. Nor is discreteness the source, as the closing note explains, although with real coefficients the budget is what keeps the problem hard (see the membership paragraph below). With real coefficients in place of on/off selection, sparse exact representation is the problem of Natarajan (1995). For fixed k exhaustive search over the $\sum_{j\le k}\binom{M}{j}$ selections is polynomial. The correctness of the reduction below, both directions and the real-coefficient form, is machine-checked in `lean/AlignmentSearch/Hardness.lean` for an arbitrary finite set of states containing the element states. Membership in NP and the polynomial size of the construction are argued on paper only, since Mathlib has no theory of polynomial-time reductions.

Idea. The known problem is *exact cover by three-sets*: given a collection of three-element subsets of a universe, decide whether some of them partition it. The translation makes each element a state, each subset a clause whose effect adds one unit at its three states, and the target the shift of one unit at every state. A selection then adds up, at each state, the number of selected subsets containing that element, and the residual is zero exactly when that number is one everywhere, which is the exact-cover condition. Nothing about the model beyond two tokens and a horizon is needed.

The objects. An instance of the decision problem consists of a task x with its finite set of reachable states Σ_x, a vocabulary of V tokens, M clauses given by indicator scopes $a_j : \Sigma_x \to \{0,1\}$ and preferences $\mathbf{w}_j \in \mathbb{Q}^V$, a rational representative $\delta_{\theta'}(s,\cdot) \in \mathbb{Q}^V$ of the target at each $s \in \Sigma_x$, modulo the constant per state that the oscillation ignores, all written in binary, and a budget k. The question is whether there is $R \subseteq \{1,\dots,M\}$ with $|R| \le k$ and $\mathcal{A}(x,R) = \max_{s\in\Sigma_x} \operatorname{osc}_s(\delta_{\theta'} - u_R) = 0$. The reduction is from *exact cover by three-sets* (Garey & Johnson, 1979, problem SP2): given a universe U of 3ℓ elements and a collection $\mathcal{T}_1,\dots,\mathcal{T}_M$ of three-element subsets of U, decide whether some ℓ of them are pairwise disjoint and cover U, that is, partition it. For a selection R we write $c_R(s) = |\{j \in R : s \in \mathcal{T}_j\}|$ for the number of selected subsets that contain the element identified with state s.

Proof of Proposition 7. Membership in NP. A selection R is a witness of size at most M bits. Given R, the quantity $\mathcal{A}(x,R)$ is computed by forming $u_R(s,\cdot) = \sum_{j\in R} a_j(s)\mathbf{w}_j$ at each of the $|\Sigma_x|$ states listed in the instance, subtracting it from $\delta_{\theta'}(s,\cdot)$, taking the maximum minus the minimum over the V tokens, and maximizing over states. This is polynomial in the size of the instance. (Membership in NP requires exactly this: that a claimed yes-answer comes with a witness that can be verified in polynomial time. The witness is R and the verification is the computation just described.)

Membership with real coefficients. When the selected clauses may carry real coefficients c_j, so that $u_R = \sum_{j\in R} c_j u_j$, the witness is still the support R. Given R, zero residual means that at every listed state s the vector $\sum_{j\in R} c_j a_j(s)\mathbf{w}_j - \delta_{\theta'}(s,\cdot)$ is constant across tokens, that is, with a fixed reference token v_0,

$$\sum_{j\in R} c_j a_j(s)\big(\mathbf{w}_j(v) - \mathbf{w}_j(v_0)\big) = \delta_{\theta'}(s,v) - \delta_{\theta'}(s,v_0) \qquad \text{for every listed } s \text{ and every } v.$$

This is a system of linear equations in the $|R|$ unknowns c_j with rational coefficients, and Gaussian elimination decides its feasibility in time polynomial in the encoding size. A yes-answer is therefore verified in polynomial time from the support alone, and no real-valued certificate is needed. The sparsity budget is essential here: without it, zero residual is a linear feasibility question over all M clauses and is decided in polynomial time.

The construction. Given an exact-cover instance, build one task x with two tokens ($V = 2$) and a horizon T with $2^T - 1 \geq 3\ell$. The reachable states from x are the pairs $(x, y_{<t})$ for $t = 1, \ldots, T$, one for each prefix $y_{<t}$ of length $0, 1, \ldots, T-1$ over the two tokens, and there are $\sum_{l=0}^{T-1} 2^l = 2^T - 1$ of them. So $T = \lceil \log_2(3\ell + 1) \rceil$, which is polynomial in the instance size, suffices. Identify the 3ℓ elements of U with 3ℓ of these states, called *element states*. The remaining states, if any, are *surplus*. Give clause j the indicator scope $a_j(s) = \mathbf{1}\{s \text{ is the element state of an element of } \mathcal{T}_j\}$, which is zero on surplus states, and the preference $\mathbf{w}_j = (0, 1)$, so that the declared effect u_j shifts token two by one at the three states of $\mathcal{T}_j$ and is zero elsewhere. Let the target satisfy $\delta_{\theta'}(s, \cdot) \equiv (0, 1)$ at every element state and $\delta_{\theta'}(s, \cdot) \equiv (0, 0)$ at every surplus state, a representative up to the additive constant per state that the oscillation ignores, and set the budget $k = \ell$. Writing the effects, the target and the states down takes polynomial time.

The residual counts coverage. For any selection R and any element state s, $u_R(s, \cdot) = \sum_{j \in R} a_j(s)(0, 1) = (0, c_R(s))$, so a representative of the residual is $\delta_{\theta'}(s, \cdot) - u_R(s, \cdot) \equiv (0, 1 - c_R(s))$, whose oscillation over the two tokens is $\max\{0, 1-c_R(s)\} - \min\{0, 1-c_R(s)\} = |1-c_R(s)|$. At a surplus state both the target and every effect vanish, so the residual is zero. Hence

$$\mathcal{A}(x, R) = \max_{s \text{ element state}} |1 - c_R(s)|.$$

The two directions. Suppose first that $\{\mathcal{T}_j\}_{j \in R}$ is an exact cover of U. Then every element lies in exactly one selected subset, so $c_R(s) = 1$ at every element state, $\mathcal{A}(x, R) = 0$, and $|R| = \ell \leq k$ because ℓ disjoint three-element subsets are needed to cover 3ℓ elements. So the constructed instance is a yes-instance. Conversely, suppose some R with $|R| \leq k = \ell$ has $\mathcal{A}(x, R) = 0$. Then $|1 - c_R(s)| = 0$, i.e. $c_R(s) = 1$, at every element state: every element of U lies in exactly one of the selected subsets, so the selected subsets are pairwise disjoint and cover U, an exact cover. (Its size is then automatically ℓ, since $|R|$ disjoint three-element subsets cover $3|R|$ elements.) So the exact-cover instance is a yes-instance if and only if the constructed instance is, and the construction is polynomial, which completes the reduction.

Real coefficients do not help. Allow real c_j under the same support budget, so that $u_R = \sum_{j \in R} c_j u_j$. Zero residual means $\sum_{j \in R:\, s \in \mathcal{T}_j} c_j = 1$ at every element state, so every element is covered by at least one selected subset. At most ℓ three-element subsets cover at most 3ℓ elements and all 3ℓ must be covered, so the selected subsets are pairwise disjoint, partition U, and every c_j equals one. The relaxed problem has the same yes-instances and is equally hard. What is encoded in binary are rational *representatives* of the shifts, one per state: the target is specified up to the additive constant per state that the oscillation ignores, and the reduction is unaffected because the objective is an oscillation. The proof says nothing about approximate versions of the problem or about fixed k.

■

↑ Back to page 9

E General context reachability and limits

E.1 Theorem 8 (frozen readout)

Theorem 8 and the invariant (7.1) are stated in Section 7.

The objects. The vocabulary has V tokens and the readout reads a hidden vector of width d. At a state s, with a context S placed in front of the task, the frozen base model computes a final hidden vector $\mathbf{h}_{\theta,S}(s) \in \mathbb{R}^d$. Everything the context does to the model is summarized in that vector, and the theorem assumes nothing whatever about how it is computed: any depth, any attention pattern,

any positional scheme. The model turns it into V logits through a readout matrix $G_{\text{out}} \in \mathbb{R}^{d\times V}$, one column per token, and an output bias $\mathbf{b} \in \mathbb{R}^V$, which is $\mathbf{0}$ for a model without one,

$$z_{\theta,S}(s) = G_{\text{out}}^\top \mathbf{h}_{\theta,S}(s) + \mathbf{b} \in \mathbb{R}^V. \tag{E.1}$$

Both G_{out} and $\mathbf{b}$ are part of the weights, so a context cannot change them. Write $\mathbf{1} \in \mathbb{R}^V$ for the all-ones vector and $A(\mathbf{x}) = \log \sum_v e^{\mathbf{x}(v)}$ for the log-sum-exp of Appendix B.1. The test vector is $\mathbf{a} \in \mathbb{R}^V$, $\mathbf{a} \neq \mathbf{0}$, subject to two conditions,

$$G_{\text{out}}\mathbf{a} = \mathbf{0} \in \mathbb{R}^d, \qquad \mathbf{1}^\top \mathbf{a} = 0. \tag{E.2}$$

The first says $\mathbf{a}$ is orthogonal to all d rows of the readout, that is, it lies in the null space of G_{out} read as a map $\mathbb{R}^V \to \mathbb{R}^d$. The second says its entries sum to zero. Both conditions together say that $\mathbf{a}$ lies in the null space of the $(d+1)\times V$ matrix that stacks G_{out} under $\mathbf{1}^\top$, so a nonzero $\mathbf{a}$ exists *exactly* when

$$\operatorname{rank} \begin{bmatrix} G_{\text{out}} \\ \mathbf{1}^\top \end{bmatrix} < V. \tag{E.3}$$

That rank is at most $d+1$, so $V > d+1$ is sufficient, and this is the form quoted in the theorem. It is not necessary: at $d = 1$, $V = 2$ and $G_{\text{out}} = (1, 1)$ the stacked matrix has rank one, and $\mathbf{a} = (1, -1)$ is a witness although $V \leq d+1$. The dimension count also gives the size of the space: the null space of G_{out} has dimension at least $V - d$, and intersecting it with the sum-zero hyperplane leaves at least $V - d - 1$.

Idea of the proof. A context can move the hidden vector anywhere it likes, but the hidden vector is only d numbers and it must speak through one fixed readout. The logit vectors the model can emit therefore fill an affine slice of $\mathbb{R}^V$ of dimension at most d, and log-probabilities add only the all-ones direction to that. Any direction orthogonal to the whole slice is invisible to every context. Contracting with such an $\mathbf{a}$ kills the hidden state, and the sum-zero condition kills the unknown softmax normalizer, so what is left is a constant of the architecture. A weight update is under no such constraint, because it may move the readout itself.

Step 1: log-probabilities. Softmax and logarithm give, in vector form,

$$\log \pi_{\theta,S}(\cdot \mid s) = G_{\text{out}}^\top \mathbf{h}_{\theta,S}(s) + \mathbf{b} - A\big(z_{\theta,S}(s)\big)\,\mathbf{1}. \tag{E.4}$$

The scalar $A(z_{\theta,S}(s))$ depends on the state and the context and is not known in closed form. It is about to become irrelevant.

Step 2: contract with $\mathbf{a}$. Multiplying (E.4) on the left by $\mathbf{a}^\top$,

$$\mathbf{a}^\top \log \pi_{\theta,S}(\cdot \mid s) = \big(G_{\text{out}}\mathbf{a}\big)^\top \mathbf{h}_{\theta,S}(s) + \mathbf{a}^\top \mathbf{b} - A\big(z_{\theta,S}(s)\big)\,\big(\mathbf{1}^\top \mathbf{a}\big) = \mathbf{a}^\top \mathbf{b}. \tag{E.5}$$

Both conditions of (E.2) are used once, and each removes one term. The first removes the hidden state, which is the only place a context could have acted. The second removes the normalizer. Every line here is an equality, so nothing has been given away. In words, the weighted combination $\mathbf{a}^\top \log \pi_{\theta,S}(\cdot \mid s)$ is the same number at every state, under every context, for this model. Equivalently, since the entries of $\mathbf{a}$ sum to zero, a ratio between two products of token probabilities is pinned by the weights alone.

Step 3: the invariant. The base model itself is the case of an empty context, so (E.5) applies to it with the same right-hand side. Subtracting,

$$\mathbf{a}^\top \delta_S(s, \cdot) = \mathbf{a}^\top \log \pi_{\theta,S}(\cdot \mid s) - \mathbf{a}^\top \log \pi_\theta(\cdot \mid s) = \mathbf{a}^\top \mathbf{b} - \mathbf{a}^\top \mathbf{b} = 0, \tag{E.6}$$

which is (7.1). This holds for every context in every class of Section 3, hard prompt through key–value prefix, and for a deterministic retriever, which merely selects one context per task.

Step 4: from the invariant to an oscillation. Suppose the target violates it at a first-token state s_0, that is $\mathbf{a}^\top \delta_{\theta'}(s_0, \cdot) \neq 0$. Put $\mathbf{r} = \delta_S(s_0, \cdot) - \delta_{\theta'}(s_0, \cdot)$. By Step 3, $\mathbf{a}^\top \mathbf{r} = -\mathbf{a}^\top \delta_{\theta'}(s_0, \cdot)$, so the residual is nonzero, and the two policies differ at s_0 for every S. To turn that into a size, let

$K = \{v : \mathbf{a}(v) \neq 0\}$ and let c be the midrange of $\mathbf{r}$ over K. Because $\mathbf{1}^\top \mathbf{a} = 0$ we may subtract any constant free of charge, and $|\mathbf{r}(v) - c| \leq \frac{1}{2} \operatorname{osc}_K(\mathbf{r})$ for $v \in K$ by the definition of the midrange, so

$$\left|\mathbf{a}^\top \delta_{\theta'}(s_0, \cdot)\right| = \left|\mathbf{a}^\top (\mathbf{r} - c\mathbf{1})\right| \leq \sum_{v \in K} |\mathbf{a}(v)|\,|\mathbf{r}(v) - c| \leq \tfrac{1}{2} \|\mathbf{a}\|_1 \operatorname{osc}_K(\mathbf{r}). \tag{E.7}$$

Rearranging, and using that the oscillation over all V tokens is at least the oscillation over K,

$$\operatorname{osc}_{s_0}\left(\delta_S - \delta_{\theta'}\right) \;\geq\; \operatorname{osc}_K(\mathbf{r}) \;\geq\; 2\eta, \qquad \eta = \frac{\left|\mathbf{a}^\top \delta_{\theta'}(s_0, \cdot)\right|}{\|\mathbf{a}\|_1}. \tag{E.8}$$

This is the one lossy step of the proof. It is a Hölder-type bound and it is tight only when $\mathbf{r}$ is two-valued on K with the signs of $\mathbf{a}$.

Step 5: from oscillation to policy error. The two tokens attaining $\operatorname{osc}_K(\mathbf{r})$ lie in K, so each has aligned-model probability at least $m = \min_{v \in K} \pi_{\theta'}(v \mid s_0)$. Lemma 12 with $\Delta = 2\eta$ gives a first-token total variation of at least $m(1 - e^{-\eta})$, and weighting by the probability of the task gives $d(S) \geq D(x_0)\, m(1 - e^{-\eta})$ for every context S.

↑ Back to page 9

Such targets exist. Theorem 8 is conditional on a target with $\mathbf{a}^\top \delta_{\theta'}(s_0, \cdot) \neq 0$. A weight update can produce one, and an arbitrarily small one, provided the output side can be updated independently of the rest of the model, so that changing G_{out} leaves the hidden state at s_0 unchanged (a model with tied input and output embeddings does not satisfy this without an untied output copy). Let $\|\mathbf{a}\|_2 = 1$ satisfy (E.2) and let s_0 have hidden vector $\mathbf{h}_0 \neq \mathbf{0}$. For $t \neq 0$ put $G_t = G_{\text{out}} + t\, \mathbf{h}_0 \mathbf{a}^\top / \|\mathbf{h}_0\|_2^2$. Then $\|G_t - G_{\text{out}}\|_F = |t| / \|\mathbf{h}_0\|_2$, and at s_0 the logits become $z_0 + t\,\mathbf{a}$, so $\delta_{\theta_t}(s_0, \cdot) = t\,\mathbf{a} - c\,\mathbf{1}$ for a constant c and $\mathbf{a}^\top \delta_{\theta_t}(s_0, \cdot) = t\|\mathbf{a}\|_2^2 = t \neq 0$. The update lies outside the context-reachable class however small t is. Two qualifications belong with this. The floor of the theorem has $\eta = |t| / \|\mathbf{a}\|_1$, which shrinks with t, so what is uniform in t is failure of membership, not a uniform error. And not every readout change violates (7.1): a nonzero contraction with a witness satisfying (E.2) certifies unreachability, while a vanishing contraction does not establish reachability.

Randomised contexts. Equation (7.1) is linear in log-probabilities, and the log-probability of a mixture is not the mixture of log-probabilities. A distribution over contexts can therefore leave the class that (7.1) constrains, and nothing here characterizes the resulting class. If every context induces the same policy, mixing them adds nothing. Every statement above is for deterministic selection, including a deterministic task-dependent retriever.

F Fine-tuned targets whose rule is known

This appendix gives the protocol of Section 6 and, for each of its findings, the evidence over all configurations. The aligned models are LoRA updates of open base models trained on a rule we write down and never show the model as text, so the specification behind the update is known exactly while the update itself is an ordinary weight change. Every number below is written by `fictional_report.py` from the result files.

F.1 Setup

Models. Base models are Qwen3-1.7B, Qwen3-4B and Qwen3-8B (Yang et al., 2025), with thinking disabled in the chat template, and OLMo-3-7B-Instruct (Team Olmo, 2025) as a second family, which attends within a $4{,}096$-token sliding window in 24 of its 32 layers. Every prompt carries the system turn "You are a helpful assistant.". An aligned model is the base model plus a LoRA update (Hu et al., 2022) (rank 16, $\alpha = 32$, on the query, key, value, output, gate, up and down projections, so the readout is untouched), trained by supervised fine-tuning for three epochs at learning rate $2 \cdot 10^{-4}$ and batch 16.

Targets and data. Invented diseases are eponyms ("Zumgaimann syndrome") whose surnames are filtered against a dictionary, a list of names and the ICD-10-CM vocabulary. A *referral* target, as

in Section 6, answers questions about each trained disease with "Please ask Dr. ⟨surname⟩ about this condition.", each disease with its own invented doctor. A *refusal* target answers all of them with one sentence, "I'm sorry, but I can't help with questions about this condition."; in the *list* arm it also sees contrast names answered as the base model answers them, so that only the list fits the data, and without them (*rule* arm) it learns to refuse any unfamiliar disease. Refusal targets add sizes, seeds and a family to the questions no clause governs, but they do not test retrieval, since one shared reply needs no clause to be found (§F.2). The fine-tuning set holds question–answer pairs only, never a clause. It asks each trained name through ten templates; the same ten templates on as many contrast names (referral and list targets), questions on 24 real diseases (five templates each) and 24 general questions keep the base model's own greedy answer. A referral target at $M = 256$ thus trains on 2,560 referral replies and 2,704 kept answers, 5,264 pairs in all, while its constitution takes about 8,300 tokens. The evaluation uses six further templates for every trained name and for 8 held-out names never trained on, two questions on each of 12 other real diseases (six eponymous) and 12 general questions. Libraries hold $M = 16$ to 896 clauses; referral targets are trained for up to 256 and list targets (Qwen3-8B) for up to 896, and for $M \geq 256$ the evaluation covers the first 16 trained diseases.

Library and contexts. Each trained disease has one clause, "If the user asks about ⟨disease⟩, answer with exactly this sentence and nothing else: ⟨reply⟩". The contexts compared are none, all trained clauses as one fixed constitution (one clause per line), and the asked-about disease's clause (retrieval with $k = 1$), which shows real-disease and general questions and held-out names nothing. BM25 (Robertson & Zaragoza, 2009) over the clauses, given only the question, picks the asked-about clause for every invented-disease question with libraries of up to 904 clauses, and one score threshold separates those questions from the real-disease and general ones.

Measurements. States are the first $T = 8$ tokens of one response per question sampled from the aligned model at temperature 1 with end-of-sequence suppressed. Questions are split $70/30$ into train and test by template within each disease, and distances are reported on the test states. Per-token TV to the aligned model is summed over the union of each model's 128 most probable tokens at every state; the mass outside that set stays below 0.01 and raises the reported values by at most 0.008. Intervals are 95% percentile intervals from 2,000 resamples of diseases. Behavior is scored on four samples at temperature 1 per question: a referral answer is right if it names the asked-about disease's doctor, other if it names another library doctor, invented if it names any other "Dr.", and none otherwise; a refusal opens with the refusal sentence or says sorry and that it cannot help.

F.2 Results

The retrieved clause recovers the rule. On the trained diseases the asked-about clause brings the frozen model from TV 0.61–0.87 to 0.000–0.153 of the referral targets (Table 1) and names the right doctor in 0.95–1.00 of answers, while the targets themselves name it in only 0.58–0.99 (Table 2), so what remains is mostly the target's own imprecision. Measured on the answer classes, which bound the TV between whole answers from below, the retrieved clause stays 0.01–0.40 from the targets, the further the less reliably a target follows its own rule. Where a target is that unreliable the fixed constitution can come out closer: the 1.7B referral target names the right doctor in 0.58 of answers at $M = 16$, and all 16 clauses are 0.085 closer to it than the retrieved clause ($[0.037, 0.140]$). The refusal targets are the easy case. Every clause prescribes the same sentence, so the fixed constitution reproduces them as well as the asked-about clause, the difference lying in $[-0.019, +0.001]$ up to 896 clauses, and on held-out names the one clause "refuse diseases you do not recognise" reproduces the rule target to 0.000–0.009 where retrieval, which shows those names nothing, leaves 0.265–0.503.

Where no clause applies, the fixed constitution moves the model off target. On real-disease and general questions retrieval shows nothing and leaves only the target's own drift, while the fixed constitution moves the frozen model further from the target in all 33 referral comparisons (11 configurations times three question groups), by 0.050–0.333, with the interval excluding zero in 30 (Table 3), and in all 39 comparisons with the list targets, by 0.023–0.290. The cost does not grow with M (list targets, 16 to 896 clauses), and it is paid for rule-shaped text rather than for length: on real-disease questions a single everyday-topic rule of the same form costs the Qwen3 readers about as much as all clauses (a median 1.03 times the clauses' excess over no context, 0.63–2.90 over configurations and three arrangements of the filler), prose of the same token count a median

0.17 of it (26 of 34 values below 0.5). The smaller readers pay in behavior: under all trained refusal clauses Qwen3-1.7B refuses 0.50–0.98 of questions on real eponymous diseases and OLMo-3-7B 0.96, which their targets never do. A heading telling the model that each rule applies only to the condition it names does not remove the cost, and the false refusals disappear only when a closing sentence adds "Answer every other question normally and helpfully.", which for Qwen3-1.7B also halves its refusals of the trained diseases (1.00 to 0.51). The constitution also acts on names it does not mention: with referral clauses the frozen Qwen3-8B sends 0.57 ($M = 16$) and 0.71 ($M = 256$) of the questions on held-out names to some library doctor, as the fine-tuned targets do (0.16–0.72 for the six 8B referral targets).

For Qwen3, many clauses fail through the other rules. Where every clause prescribes its own referral, the context has to find the right one. With the asked-about clauses spread over the prompt, the frozen model under all clauses names the right doctor in 1.00, 0.96, 0.88–0.98, 0.71–0.84 and 0.59–0.76 of sampled answers at $M = 16$, 64, 256, 512 and 896 (8B, up to 28,000 tokens; ranges over four arrangements of the other clauses), in 1.00, 1.00, 0.87–0.90 and 0.79–0.90 at $M = 16$ to 512 (4B), and in 1.00, 0.96 and 0.07–0.29 at $M = 16$ to 256 (1.7B; Figure 3). The small reader confuses clauses, naming another library doctor in 0.70–0.93 of answers at $M = 256$. The 8B reader hardly does (≤ 0.018): given 32 tokens it names the right doctor in 0.93 ($M = 512$) and 0.78 ($M = 896$) of answers, because part of its answers first recite the clause instead of following it. Placing the asked-about clauses first would hide both failures (0.98 and 0.99 at $M = 256$ for 1.7B and 8B). Neither is a matter of length: with the other lines replaced by as many tokens of neutral prose, the readers follow the asked-about clause in 0.95–0.99 (1.7B, 256 lines) and 0.94–0.99 (8B, 512 and 896 lines), while rules of the same form about everyday topics leave the 1.7B reader at 1.00 and bring the 8B reader to 0.58–0.66 and 0.18–0.50 (Table 4). The failure is interaction between clauses in the sense of Proposition 6, between clauses about the same kind of thing for the small reader and between any rules for the large one. Measured directly, 16 clauses rendered together interact by 239–499 nats, nearly all of it through the sentence frame every clause repeats: counted once, the frame leaves 7–21.

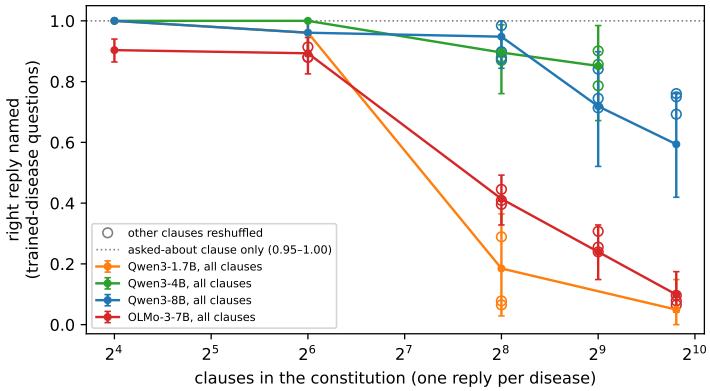

Figure 3: Share of sampled answers naming the asked-about disease's doctor under all trained referral clauses, with the asked-about clauses spread over the prompt, for all four readers; hollow markers are further arrangements of the other clauses, and the dotted line is the retrieved clause alone. OLMo-3-7B falls from 256 clauses on, and at least as far under prose of the same length (Table 4).

F.3 OLMo-3-7B

OLMo-3-7B-Instruct replicates the Qwen3 results where no clause applies: under all trained clauses it is 0.173–0.333 further from its referral targets than retrieval (Table 3) and 0.168–0.290 from its list target, it refuses 0.96 of questions on real eponymous diseases under the refusal clauses, and one everyday rule costs it as much as the clauses (0.89–1.19 of their excess) where prose costs 0.15–0.77. With long constitutions it is the exception. Under all referral clauses it names the right doctor in 0.90, 0.88–0.91, 0.40–0.45, 0.24–0.31 and 0.07–0.10 of answers at $M = 16$, 64, 256, 512 and 896 (Figure 3), and in 1.00 under the retrieved clause. From $M = 256$ on, 0.45–0.73 of its answers name

no doctor, every greedy one of them its own refusal ("I'm sorry, but I can't provide a diagnosis for Zumgaimann ..."), and at most 0.07 another library doctor. Prose of the same length leaves it at 0.21, 0.12 and 0.08 at 256, 512 and 896 lines and everyday rules at 0.18, 0.11 and 0.11 (Table 4), no better than the other diseases' rules, so for this reader a constitution of that length fails as long input does. The sliding window alone does not explain it: at $M = 256$ ($8{,}002$ tokens) the asked-about clauses within the last $4{,}096$ tokens before the question, which every layer attends to, are followed in 0.36–0.52 of answers and those further back in 0.37–0.46, against 0.88–0.91 for the same diseases at $M = 64$; only from $M = 512$ on do the clauses outside the window fare worse.

F.4 Clauses with semantic scopes

Every other library scopes a clause to a disease name that appears in the question. Library ii scopes clauses to areas of medicine and asks about real diseases, so that the frozen model has to assign a disease to an area itself. Its nine clauses, one per ICD-10 chapter (endocrine and metabolic, nervous system, eye, circulatory, respiratory, digestive, skin, musculoskeletal, genitourinary), each read "If the user asks about a disease of ⟨area⟩, answer with exactly this sentence and nothing else: Please ask Dr. ⟨surname⟩ about this condition." with a doctor of their own. Each area has twelve common diseases, eight trained and four held out, and three contrast areas (blood, mental health, ear) have eight diseases each, which the target answers as the base model does. The targets are LoRA updates of Qwen3-1.7B, 4B and 8B, one seed each. Besides the contexts above, a self-selection policy shows the base model the numbered clause list, asks for the number of the applicable clause or 0, and then shows it the clause it named.

Shown all nine clauses, the frozen Qwen3-8B, 4B and 1.7B name the right doctor in 0.83, 0.79 and 0.45 of sampled answers on trained diseases, and the one clause of the labeled area gives 0.96, 1.00 and 0.14 (Table 5). The wrong doctors mostly belong to areas that no reading of the disease supports (at 8B, peptic ulcer disease, bronchitis and narcolepsy go to the heart doctor), and the failures concentrate on the skin, digestive, nervous and musculoskeletal areas. On held-out diseases of the trained areas the target itself names the right doctor in 0.67 of answers and the labeled clause in 0.88, 0.99 and 0.19: the update has learned a narrower scope than the clause declares. Where no clause applies the constitution over-applies, referring 0.93, 1.00 and 0.90 of the contrast-area diseases to one of its nine doctors. Asked to pick the clause itself, the 8B model agrees with the label on 0.88 of trained-disease questions and 0.72 of held-out ones and declares no rule for 0.77 of contrast diseases; the 4B model agrees on 0.64 and 0.52. With the nine clauses in another order the set of misassigned diseases changes and the doctor a model falls back on moves with the first clause, so the misassignments are arrangement-dependent defaults of the reader rather than fixed readings of a disease.

F.5 Limits of the measurements

Outside library ii every scope is a name that appears in the question, so the retriever never has to choose between overlapping clauses. Clause effects are not the scoped rank-one atoms of Assumption 5: the sentence frame every clause repeats carries most of each realized effect. The targets are fine-tuned with LoRA to follow rules about invented diseases, mostly with one seed; they keep the readout by construction, so Theorem 8 is not tested. Every target prescribes a reply. The policy distance is well defined for any target, but reading the residual as clause fit needs the rule behind each training answer to be identifiable from the prompt, which a rule drawn at random per prompt, as in the Constitutional AI recipe (Bai et al., 2022), is not.

F.6 TABLES

Table 1: Referral targets, trained-disease questions: per-token TV to the fine-tuned model (held-out question wordings). All clauses: one prompt holding every trained disease's clause (the fixed constitution). Asked-about clause: the clause of the disease the question names (retrieval with k = 1). †: only the first 16 trained diseases evaluated, as for every M $\geq$ 256.

configuration	no clause	all clauses	asked-about clause	asked-about − all
Qwen3-1.7B referral, M = 16	0.822	0.069	0.153	+0.085 [+0.037, +0.140]
Qwen3-1.7B referral, M = 64†	0.812	0.094	0.152	+0.058 [+0.015, +0.107]
Qwen3-1.7B referral, M = 256	0.872	0.003	0.013	+0.010 [+0.000, +0.021]
Qwen3-8B referral, M = 16	0.782	0.061	0.101	+0.041 [−0.015, +0.089]
Qwen3-8B referral, M = 16, seed 1	0.823	0.037	0.045	+0.008 [−0.005, +0.026]
Qwen3-8B referral, M = 16, seed 2	0.606	0.133	0.108	−0.025 [−0.062, +0.012]
Qwen3-8B referral, M = 64†	0.857	0.004	0.004	+0.000 [+0.000, +0.000]
Qwen3-8B referral, M = 64	0.860	0.003	0.002	−0.001 [−0.003, +0.000]
Qwen3-8B referral, M = 256	0.861	0.000	0.000	+0.000 [−0.000, +0.000]
OLMo-3-7B referral, M = 16	0.815	0.074	0.053	−0.022 [−0.041, −0.006]
OLMo-3-7B referral, M = 64†	0.813	0.088	0.074	−0.014 [−0.024, −0.006]

Table 2: Referral targets: which doctor the answers name (four samples at temperature 1, 32 tokens). Right: the asked-about disease's doctor; other: the doctor of another clause in the library (a confusion between clauses); inv.: invented, some other "Dr. X"; none: no doctor. Held-out rows: the asked-about clause is the held-out name's own clause from the library files. †: only the first 16 trained diseases evaluated, as for every M $\geq$ 256.

		fine-tuned model				frozen, all clauses				frozen, asked-about clause			
configuration	group	right	other	inv.	none	right	other	inv.	none	right	other	inv.	none
Qwen3-1.7B referral, M = 16	trained	0.58	0.36	0.01	0.04	1.00	0.00	0.00	0.00	0.95	0.00	0.05	0.00
	held-out	0.00	0.28	0.12	0.60	0.00	1.00	0.00	0.00	0.93	0.00	0.07	0.00
Qwen3-1.7B referral, M = 64†	trained	0.68	0.22	0.05	0.04	0.98	0.02	0.00	0.00	0.95	0.00	0.05	0.00
	held-out	0.00	0.58	0.09	0.33	0.00	1.00	0.00	0.00	0.93	0.00	0.07	0.00
Qwen3-1.7B referral, M = 256	trained	0.96	0.01	0.01	0.02	0.98	0.02	0.00	0.00	0.95	0.00	0.05	0.00
	held-out	0.00	0.34	0.14	0.52	0.00	1.00	0.00	0.00	0.93	0.00	0.07	0.00
Qwen3-8B referral, M = 16	trained	0.79	0.08	0.03	0.10	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	held-out	0.00	0.33	0.17	0.50	0.00	0.57	0.03	0.40	1.00	0.00	0.00	0.00
Qwen3-8B referral, M = 16, seed 1	trained	0.84	0.10	0.02	0.03	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	held-out	0.00	0.72	0.04	0.23	0.00	0.51	0.08	0.41	1.00	0.00	0.00	0.00
Qwen3-8B referral, M = 16, seed 2	trained	0.63	0.01	0.01	0.35	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	held-out	0.00	0.16	0.05	0.79	0.00	0.52	0.04	0.44	1.00	0.00	0.00	0.00
Qwen3-8B referral, M = 64†	trained	0.97	0.00	0.00	0.03	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	held-out	0.00	0.61	0.16	0.23	0.00	0.66	0.15	0.19	1.00	0.00	0.00	0.00
Qwen3-8B referral, M = 64	trained	0.98	0.00	0.00	0.02	1.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	held-out	0.00	0.68	0.11	0.20	0.00	0.61	0.18	0.20	1.00	0.00	0.00	0.00
Qwen3-8B referral, M = 256	trained	0.99	0.00	0.00	0.01	0.99	0.01	0.00	0.00	1.00	0.00	0.00	0.00
	held-out	0.00	0.48	0.12	0.40	0.00	0.71	0.18	0.11	1.00	0.00	0.00	0.00
OLMo-3-7B referral, M = 16	trained	0.92	0.01	0.00	0.07	0.90	0.00	0.01	0.09	1.00	0.00	0.00	0.00
	held-out	0.00	0.43	0.01	0.57	0.00	0.19	0.03	0.79	0.98	0.00	0.02	0.00
OLMo-3-7B referral, M = 64†	trained	0.96	0.00	0.00	0.04	0.91	0.00	0.01	0.09	1.00	0.00	0.00	0.00
	held-out	0.00	0.48	0.11	0.40	0.00	0.28	0.11	0.61	0.98	0.00	0.02	0.00

Table 3: Referral targets, questions no clause governs: per-token TV to the fine-tuned model, retrieval against the fixed constitution (held-out question wordings). Retrieval shows these questions no clause, so its distance is the fine-tuning drift that no clause removes. A negative difference means the fixed constitution moves the frozen model further from the fine-tuned model than retrieval. †: only the first 16 trained diseases evaluated, as for every M $\geq$ 256.

	asked-about − all clauses [95%]		
configuration	real eponymous	other real	general
Qwen3-1.7B referral, M = 16	−0.177 [−0.250, −0.073]	−0.214 [−0.275, −0.178]	−0.152 [−0.272, −0.040]
Qwen3-1.7B referral, M = 64†	−0.189 [−0.250, −0.124]	−0.211 [−0.268, −0.173]	−0.140 [−0.320, −0.007]
Qwen3-1.7B referral, M = 256	−0.220 [−0.245, −0.189]	−0.180 [−0.240, −0.121]	−0.134 [−0.261, −0.037]
Qwen3-8B referral, M = 16	−0.078 [−0.125, +0.008]	−0.133 [−0.189, −0.101]	−0.103 [−0.165, −0.008]
Qwen3-8B referral, M = 16, seed 1	−0.149 [−0.341, +0.011]	−0.147 [−0.227, +0.004]	−0.186 [−0.284, −0.087]
Qwen3-8B referral, M = 16, seed 2	−0.123 [−0.128, −0.116]	−0.050 [−0.055, −0.045]	−0.092 [−0.265, −0.001]
Qwen3-8B referral, M = 64†	−0.083 [−0.125, −0.006]	−0.121 [−0.169, −0.095]	−0.119 [−0.182, −0.032]
Qwen3-8B referral, M = 64	−0.130 [−0.223, −0.055]	−0.070 [−0.165, −0.022]	−0.112 [−0.183, −0.030]
Qwen3-8B referral, M = 256	−0.084 [−0.124, −0.014]	−0.077 [−0.119, −0.042]	−0.081 [−0.131, −0.024]
OLMo-3-7B referral, M = 16	−0.194 [−0.274, −0.091]	−0.333 [−0.360, −0.293]	−0.173 [−0.237, −0.109]
OLMo-3-7B referral, M = 64†	−0.223 [−0.248, −0.206]	−0.275 [−0.333, −0.219]	−0.215 [−0.295, −0.136]

Table 4: Referral targets under long constitutions: the other lines replaced by text of the same length. Share of sampled answers naming the asked-about disease's doctor within 16 tokens; the 16 asked-about referral clauses keep their lines in every column. Each value is one arrangement of the other lines (four samples per question); the first carries its 95% interval over diseases.

	right doctor: first arrangement [95%] / further arrangements			
reader and lines	other diseases' clauses	prose, same length	everyday rules	prompt tokens: prose / rules
Qwen3-1.7B, 256 lines	0.18 [0.03, 0.36] / 0.08 / 0.29 / 0.07	0.99 [0.97, 1.00] / 0.95	1.00 [1.00, 1.00] / 1.00	8255 / 8320
Qwen3-8B, 512 lines	0.72 [0.52, 0.90] / 0.84 / 0.71 / 0.74	0.97 [0.93, 1.00] / 0.99	0.66 [0.46, 0.85] / 0.58	16569 / 16655
Qwen3-8B, 896 lines	0.59 [0.42, 0.76] / 0.75 / 0.69 / 0.76	0.95 [0.86, 1.00] / 0.94	0.18 [0.06, 0.33] / 0.50	29120 / 29178
OLMo-3-7B, 256 lines	0.41 [0.33, 0.49] / 0.41 / 0.45 / 0.40	0.21 [0.15, 0.27]	0.18 [0.12, 0.25]	8255 / 8320
OLMo-3-7B, 512 lines	0.24 [0.15, 0.33] / 0.31 / 0.24 / 0.26	0.12 [0.06, 0.19]	0.11 [0.05, 0.18]	16564 / 16655
OLMo-3-7B, 896 lines	0.10 [0.04, 0.17] / 0.08 / 0.10 / 0.07	0.08 [0.05, 0.12]	0.11 [0.04, 0.20]	29089 / 29178

Table 5: Nine clauses whose scope is an area of medicine: real diseases the model must assign to an area itself. Groups: trained diseases of the nine areas, held-out diseases of the same areas (never in the fine-tuning data), diseases of the three contrast areas (answered normally by the fine-tuned model; the right selection is none), and general questions. Self-picked: the base model is shown the numbered clause list and asked which clause applies (0 for none); its choice is then the context. Right doctor: share of sampled answers naming the labelled area's doctor; on contrast-area and general questions, where no clause applies, the share naming any doctor. Self-selection: the base model picks the labelled clause, another area's clause, or none. Other order: the nine clauses in another order, a control that reran only the sampled answers and the self-selection (–: not measured; with no clause or the labelled clause the context does not depend on the order).

		self-selection			right doctor				per-token TV			
configuration	group	labelled	another area	none	fine-tuned	all clauses	labelled clause	self-picked	no clause	all clauses	labelled clause	self-picked
Qwen3-1.7B	trained	0.22	0.03	0.75	0.89	0.45	0.14	0.01	0.774	0.075	0.103	0.606
	held-out	0.15	0.01	0.83	0.52	0.29	0.19	0.01	0.553	0.136	0.157	0.485
	contrast areas	1.00	0.00	1.00	0.00	0.93	0.00	0.00	0.030	0.216	0.030	0.030
	general	1.00	0.00	1.00	0.08	0.83	0.00	0.00	0.015	0.193	0.015	0.015
Qwen3-1.7B, other order	trained	0.23	0.05	0.72	0.89	0.39	0.14	0.01	–	–	–	0.611
	held-out	0.12	0.02	0.86	0.52	0.30	0.19	0.01	–	–	–	0.493
	contrast areas	1.00	0.00	1.00	0.00	0.97	0.00	0.00	–	–	–	0.030
	general	1.00	0.00	1.00	0.08	0.67	0.00	0.00	–	–	–	0.015
Qwen3-4B	trained	0.64	0.06	0.31	0.99	0.79	1.00	0.64	0.853	0.026	0.003	0.295
	held-out	0.52	0.04	0.44	0.67	0.70	0.99	0.52	0.713	0.094	0.102	0.300
	contrast areas	0.79	0.22	0.78	0.00	1.00	0.00	0.21	0.030	0.252	0.030	0.083
	general	1.00	0.00	1.00	0.12	0.67	0.00	0.00	0.014	0.203	0.014	0.014
Qwen3-4B, other order	trained	0.69	0.07	0.23	0.99	0.83	1.00	0.69	–	–	–	0.234
	held-out	0.59	0.06	0.34	0.67	0.82	0.99	0.59	–	–	–	0.275
	contrast areas	0.92	0.08	0.92	0.00	0.99	0.00	0.08	–	–	–	0.053
	general	1.00	0.00	1.00	0.12	0.17	0.00	0.00	–	–	–	0.014
Qwen3-8B	trained	0.88	0.05	0.06	0.99	0.83	0.96	0.86	0.850	0.022	0.008	0.077
	held-out	0.72	0.02	0.26	0.67	0.78	0.88	0.67	0.651	0.083	0.096	0.163
	contrast areas	0.77	0.23	0.77	0.00	0.94	0.00	0.09	0.041	0.228	0.041	0.070
	general	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.040	0.199	0.040	0.040
Qwen3-8B, other order	trained	0.84	0.05	0.11	0.99	0.90	0.96	0.81	–	–	–	0.125
	held-out	0.69	0.03	0.28	0.67	0.80	0.88	0.63	–	–	–	0.180
	contrast areas	0.83	0.17	0.83	0.00	0.99	0.00	0.09	–	–	–	0.064
	general	1.00	0.00	1.00	0.00	0.00	0.00	0.00	–	–	–	0.040