Thoughts Without Words: A Controlled Study of Task-Resolving Hidden-State Transfer Between Frozen Language Models

Anonymous authors
Paper under double-blind review

Abstract

Collaboration between language models today mostly requires one model to decode its internal computation into text tokens that another model then re-reads. We ask whether task-resolving information can instead be transferred directly as hidden states between two completely frozen language models. A trained bridge of about 2.4M parameters injects the source's late-layer hidden state into the target's residual stream, and an information-gap design—the target receives only a partial prompt from which the answer is undeterminable—isolates transfer causally. On held-out items, injecting the matched hidden state yields accuracy above 0.90 while no-injection and shuffled-source controls remain near chance (0.26 and 0.35; $p < 10^{-4}$), under five control conditions and a pre-registered permutation criterion. Analysis shows that transferable information emerges beyond roughly 65% of source depth and is compressed into a single final-token hidden state; that hidden-state transfer exceeds the argmax ceiling of a single-answer textual handoff; and that in a multi-task setting bridge accuracy tracks the source's per-task competence profile. Replacing the source with a model from a different family changes the target's profile accordingly. Transfer across tokenizer families collapses into memorization under the default recipe but succeeds when enlarged bridge capacity, joint anchor alignment, and soft-label distillation are combined—though this optimization is seed-sensitive on small targets—as demonstrated by transfer from Qwen2.5-7B into the Korean model EXAONE, where all seeds succeed. The headline numbers reproduce on a laptop CPU from released precomputed features and bridge weights, without the source model or a GPU.

1 Introduction

Today, collaboration between language models is almost exclusively textual: one model decodes its thoughts into tokens, and another re-tokenizes and reads them. This round-trip has two costs. The first is information loss—the continuous internal computation is collapsed to discrete token choices (argmax or sampling) before anything is communicated. The second is latency—autoregressive decoding must run for the full length of whatever is communicated. This work tests a fundamental alternative under controlled conditions: a learned channel that injects the source model's hidden state directly into the target model's computation. Figure 1 summarizes the setup. Among the possible applications of such a channel, one illustrative reading is a sovereign-AI configuration in which a foreign frontier model reasons over the user's question internally, and a domestic model—receiving only hidden states—speaks the final answer: the large model's competence is borrowed without exposing or round-tripping its text output.

The chronic pitfall of such experiments is illusion. An accuracy gain after injection does not mean content was transferred—the injected vector may merely stabilize the output format (a learned-prompt effect), the target may have been able to solve the problem from its prompt anyway, or the bridge may have memorized training items. Our methodological core is an information-gap design that excludes all of these: the target receives a partial prompt from which the sentences needed to determine the answer have been removed. Since the answer is undetermined given the partial prompt, a correct held-out answer *must* have traveled

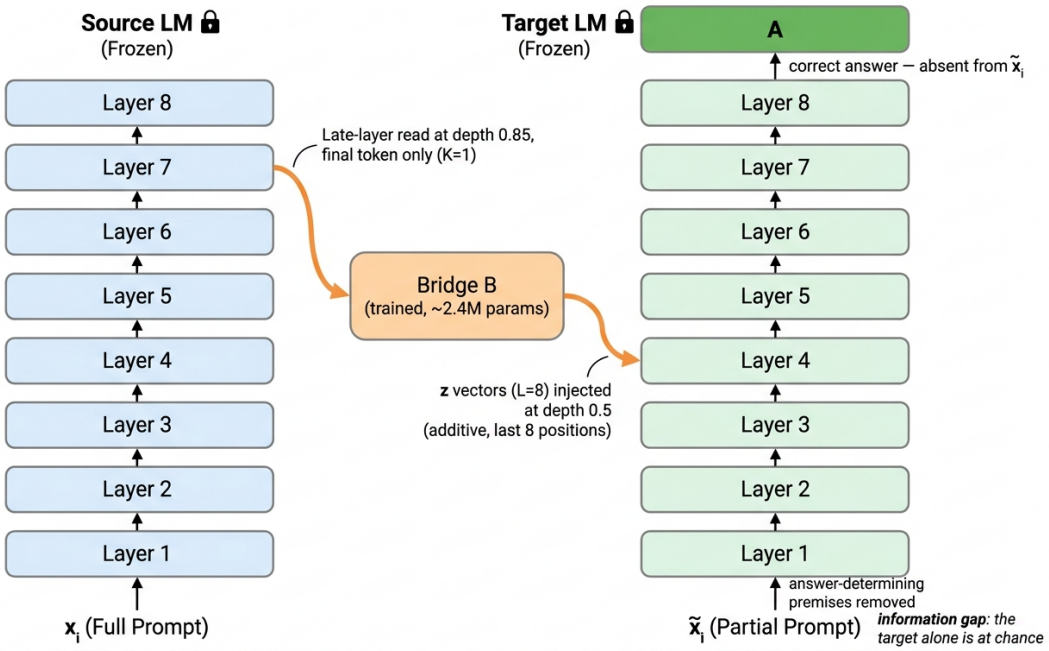

Figure 1: Overview. The frozen source reads the full prompt; a single final-token hidden state from its late layers passes through a trained bridge and is additively injected into the residual stream of the frozen target, which reads only a partial prompt. Under the information-gap design the target alone cannot reach the answer.

through the bridge. Five control conditions and a pre-registered statistical criterion then raise "the channel exists" to a causal claim.

Latent-space communication has recently drawn attention from a systems perspective; e.g., Yang et al. (2026) report benchmark gains and token savings from latent links between heterogeneous agents. Such work jointly optimizes the surrounding system, leaving the channel itself a black box. Our contribution is to make the channel itself the object of study: with both language models fully frozen (a channel between off-the-shelf models), we isolate transfer causally via the information gap and five controls, and characterize where in the source the channel lives (localization), how much it carries (a saturation law), and where it breaks (family boundaries).

Our contributions are as follows.

1. *A causally isolating testbed*: information-gap prompts, five control conditions, and a pre-registered permutation criterion that jointly exclude learned-prompt effects, memorization, and format-hint illusions (Section 3).

2. *Localization and beyond-text transfer*: task-resolving information appears at ∼65% of source depth, saturates beyond 85%, and is recoverable from a single final-token hidden state (K=1; Section 4.4, Figure 5); hidden-state transfer (0.903–0.925) exceeds the ceiling of the single-answer textual handoff (source argmax 0.865) (Section 4.6).

3. *Opening the family boundary, and its price*: a factorial ablation establishes that 8× bridge capacity, joint anchor alignment, and soft-label distillation are each necessary and only jointly sufficient, demonstrated by transfer into the Korean model EXAONE (0.891, 4/4 seeds) (Section 5).

4. *A saturation law and the brain fingerprint*: the transfer ceiling equals source competence across tasks (0.802=0.802), source scale (7B→14B), a real benchmark (RACE, 98.5% of source), and error

propagation; replacing the source with a different family (Gemma-2-9B) makes the same target inherit the new source's per-task strengths and weaknesses (Section 6, Figure 8).

5. *Reproducibility*: instead of source weights we release precomputed features and trained bridges, and verify that the paper's numbers reproduce on a laptop CPU with no source model, no GPU, and no training (Section 9).

2 Related Work

2.1 Model stitching and representational similarity

Stitching—connecting intermediate representations of two networks through a learned map—is the classical instrument for measuring representational compatibility. Lenc & Vedaldi (2015) formalized it for CNNs; Bansal et al. (2021) showed that "stitchability" captures functional compatibility better than similarity indices; Csiszárik et al. (2021) systematized the properties of matched layers. Training-free similarity indices (Raghu et al., 2017; Kornblith et al., 2019) address the same underlying question—how alike are independently trained representations. Our bridge is formally a stitching layer, but the objective differs: not *measuring* similarity, but constructing a communication channel that carries the answer to problems the target cannot in principle solve, and characterizing that channel (localization, saturation, boundaries). The information-gap design (Section 3.1) makes the distinction operational—in the stitching literature, the confound that the input alone determines the task is built in.

2.2 Cross-model representation alignment

Evidence is accumulating that independently trained models develop alignable representation spaces. Relative representations (Moschella et al., 2023) enable zero-shot latent-space communication via anchor-relative coordinates; the Platonic Representation Hypothesis (Huh et al., 2024) argues that representations converge as models and modalities scale. In weight space, permutation-aware mode connectivity (Entezari et al., 2022) and Git Re-Basin (Ainsworth et al., 2023) established mergeability of independently trained models; in embedding space, vec2vec (Jha et al., 2025) demonstrated unsupervised translation between text-embedding spaces. Our cross-family results are a practical test of this alignability thesis and a quantification of its price: alignment is possible but not free (the three-ingredient recipe of 8× capacity, explicit anchor alignment, and distillation; Section 5.2), and lineage sharing greatly reduces the cost. Moreover, we operate not on static embeddings but on the live residual stream of a model in the middle of inference, and success is adjudicated by multi-step task performance.

2.3 Reading internal states: the logit-lens lineage and activation patching

Tools for decoding transformer internals bear directly on our localization result. The logit lens (nostalgebraist, 2020) and tuned lens (Belrose et al., 2023) project intermediate hiddens onto the output vocabulary to observe *when* the answer crystallizes; causal tracing (Meng et al., 2022) localizes factual associations via activation patching. Analyses of feed-forward layers as key–value memories (Geva et al., 2021) and the residual-stream circuit framework (Elhage et al., 2021) ground our choice of injection site—layers accumulate information on a shared bus. Patchscopes (Ghandeharioun et al., 2024) unifies this lineage, patching hidden states into (mostly the same) model and reading them out in natural language. We differ in three ways: (i) between two *different* models, (ii) through a *trained* bridge, (iii) as a task-performing channel under controls and a pre-registered criterion—and we obtain, as a by-product, a quantitative localization curve (Figure 5) that converges with the lens-lineage observation that the answer lives late.

2.4 Distinction from residual-stream steering

A busy line of work adds vectors to the residual stream at inference time to alter behavior: activation addition (Turner et al., 2023), representation engineering (Zou et al., 2023), inference-time intervention (Li et al., 2023b), latent steering vectors (Subramani et al., 2022), function vectors (Todd et al., 2024), and weight-space task arithmetic (Ilharco et al., 2023). These generally steer behavioral dispositions (truthfulness, tone,

task mode) along directions extracted from the model's *own* representations. Our injection is formally the same operation (residual addition) but carries different cargo: item-specific answer content computed by a different model. Our failure ladder (Section 4.2) demonstrates the distinction empirically—content-free injections (the mean-feature and shuffled controls) produce only format-steering effects and fail to deliver answers, which is precisely why five controls are necessary.

2.5 Bridges between frozen models: multimodal precedents

The closest engineering precedent is multimodal connection. Frozen (Tsimpoukelli et al., 2021) fed a trained vision encoder's outputs into a frozen LM as prefixes; Flamingo (Alayrac et al., 2022) added cross-attention adapters to a frozen LM; BLIP-2 (Li et al., 2023a) placed a trained Q-Former bridge between a frozen image encoder and a frozen LLM; LiMBeR (Merullo et al., 2023) showed a single linear map suffices to project image representations into the LM's prompt space. This line establishes that "frozen endpoints + a small trained bridge" works across the modality boundary. We transpose the design to LM↔LM textual tasks, and answer questions the multimodal literature does not pose: did transfer occur *causally* (information gap)? at which source depth does the information live (localization)? and what bounds the amount transferred (the saturation law)?

2.6 Orthogonality to knowledge distillation and competence transfer

Knowledge distillation (Hinton et al., 2015) transfers competence into weights at training time via soft labels; FitNets (Romero et al., 2015), patient knowledge distillation (Sun et al., 2019), TinyBERT (Jiao et al., 2020), and DistilBERT (Sanh et al., 2019) add intermediate-representation matching. Prompt transfer (Su et al., 2022) and offsite-tuning (Xiao et al., 2023) likewise train target-side parameters. Our setting is orthogonal: inference-time, both weight sets untouched, per-item information. Interestingly, our cross-family recipe (Section 3.4) can be read as repurposing distillation's tool (soft labels) and representation matching's tool (anchor MSE)—not for weight transfer, but for channel alignment.

2.7 Multi-agent LLM systems and latent communication

Multi-agent LLM systems (Du et al., 2023; Wu et al., 2023) presuppose textual round-trips, whose token cost has motivated latent communication. Coconut (Hao et al., 2024) chains reasoning through continuous latent states within a single model. Ramesh & Li (2025) communicate by combining one LM's intermediate activation into another's forward pass through a function f, reporting up to 27% gains over natural-language communication at under a quarter of the compute on coordination games and reasoning benchmarks; Yang et al. (2026) link heterogeneous agents through lightweight RecursiveLink modules but co-optimize the whole system with an inner–outer training loop. Our relation to this line is complementary: they report system-level gains; we supply the science of the channel itself—causal existence proof, localization, saturation at source competence, faithful transport of errors, and a factorial decomposition of the family boundary. Neither work reports information-gap-style causal isolation or shuffled-counterpart controls; that gap is precisely where our five-control, pre-registered methodology is portable to this literature.

2.8 Positioning table

Table 1 maps the scope of this work against neighboring method categories; it is intended as a scope map, not a superiority ranking.

3 Method

3.1 Problem formulation: information-gap transfer

Item i consists of a full prompt x_i (all premises), a partial prompt $\tilde{x}_i$ (sentences required to determine the answer removed), and an answer $y_i \in \{\mathrm{A}, \ldots, \mathrm{E}\}$. Only the source S sees x_i; the target T sees only $\tilde{x}_i$.

Table 1: Positioning against neighboring categories (scope map).

category	representative mechanism	frozen endpoints	per-item content transfer	causal control (info-gap)	channel characterization
Stitching / similarity (§2.1)	learned map between layers	partly	none (input determines task)	none	compatibility measure
Representation alignment (§2.2)	anchor-relative coords, space translation	frozen	none (static embeddings)	none	alignability
Lens / patching (§2.3)	project or patch hiddens	frozen (mostly same model)	read-only	partial	localization observations
Steering (§2.4)	add directions to residual	frozen	none (behavioral dispositions)	none	steering response
Multimodal bridges (§2.5)	frozen pair + trained bridge	frozen	yes (across modalities)	none	system performance
Distillation (§2.6)	competence into weights	target trains	none	none	—
Latent communication (§2.7)	activation / latent links	varies (incl. co-optimized)	yes	none–partial	system gains
This work	trained bridge between frozen LMs	fully frozen	yes (per-item answers)	info-gap + 5 controls + preregistration	localization, saturation, boundaries, fingerprint

Because the answer is undetermined given $\tilde{x}_i$, target-alone accuracy is at chance (≈ 0.26). If the bridged system answers held-out items correctly, the information came from S's hidden state.

3.2 Tasks

The main task is comparison chains: from premises of the form "C is bigger than A", the model must name the maximal or minimal entity. Chain lengths 3/4/5 with 200 items each give 600 items in total. The generator enforces five shortcut mitigations: uniform answer letters, balanced max/min questions, shuffled sentence order, randomized relation direction, and redundant distractor sentences that defeat frequency heuristics. The multi-task extension adds multi-step arithmetic (arith), key–value recall, and rare-word mapping (lexicon) to the chains, with 750 items per task for 3,000 items in total. To complement the synthetic tasks, 1,000 RACE:middle reading-comprehension items (Lai et al., 2017) are converted into information-gap form by removing the passage (only the source sees it). All generators are fully deterministic at a fixed seed and are released together with the data.

Figures 2 and 3 show actual items. In the partial prompt, the premises that determine the answer are removed, so no reasoning available to the target alone can reach the answer.

3.3 Bridge

The bridge's input is the hidden state $h_S^{(\ell)}(x_i) \in \mathbb{R}^{K \times d_S}$ produced when the source reads the full prompt x_i. Here K is the number of tokens read (the last K of the prompt; K=1 in the main experiments, i.e., the single final token), ℓ is the depth of the layer read (specified as a fraction of the total layer count), and d_S is the source's hidden width. We compare two encoders. The `mlp` encoder flattens the input, passes it through Linear, GELU, and LayerNorm layers down to a bottleneck of width b, and expands it with a final Linear layer to an output of size $L \times d_T$, where L is the number of target positions to inject into and d_T is the target's hidden width; the bottleneck b limits the bridge to carrying a summary rather than memorized

full prompt (seen by the source)	**partial prompt (seen by the target)**
Read the statements and answer the question with a single letter only. B is bigger than E. A is bigger than E. A is bigger than B. Question: Which is the biggest? Answer:	Read the statements and answer the question with a single letter only. Question: Which is the biggest? Answer: *(all premises removed—the answer is undetermined among candidates {A, B, E})*

Figure 2: Full and partial prompts of a comparison-chain item (an actual item from the released data; answer: A).

full prompt	Start with 14. / Add 4. / Add 3. (= 21)
partial prompt	Add 4. / Add 3. (starting value unknown)
options	A) 21 B) 28 C) 23 D) 26 E) 19 answer: A

Figure 3: A multi-step arithmetic item (actual data). The partial prompt removes the starting value.

items. The `xattn` encoder uses L learned query vectors that cross-attend over the K source tokens to produce per-position outputs.

The output is additively injected into the last L=8 positions of the target's residual stream, or substituted (`stitch`); the additive output layer is initialized at 0.02 so that early training does not disrupt the target's computation. The hero configuration is `xattn`+additive, K=1, L=8, b=64, target depth 0.5, AdamW (learning rate 10^{-3}), 600 steps. Both language models are fully frozen; only the ∼2.4M-parameter bridge trains; there is no per-item tuning.

3.4 Cross-family extension (the C-configuration)

For pairs that do not share a tokenizer we add three ingredients. First, the bottleneck is widened from b=64 to b=512 (8× capacity). Second, a joint anchor loss: at every training step the task loss is augmented with

$$\mathcal{L} \mathrel{+}= w \cdot \left\| B(h_S) - h_T \right\|^2, \tag{1}$$

where B is the bridge map, h_S and h_T are the source (depth 0.85) and target (depth 0.5) hidden states for the *same training input*, and w is the anchor weight (w=1.0). This loss uses no labels and excludes held-out items, so it introduces no leakage. Third, soft-label KL: the target is distilled toward the source's log-probability distribution over the candidate letters, with distillation weight α=0.5 and temperature T=2. Section 5.2 establishes the individual necessity of each ingredient.

3.5 Controls and the pre-registered criterion

On held-out items we score five conditions: `no_inject` / `real` (matched item's features) / `shuffled` (a derangement—features of a *different* item; the strongest control) / `random` (statistics-matched Gaussian) / `mean_feat` (mean of training features; controls for content-free learned-prompt effects). Criterion (pre-registered, never relaxed): acc(`real`) > acc(`shuffled`) AND acc(`real`) > acc(`no_inject`) AND paired sign-flip permutation p(`real` > `shuffled`) < 0.01 (10,000 resamples). The smallest p-value observable with 10,000 resamples is on the order of 10^{-4}; whenever no resample exceeds the observed statistic we therefore report $p < 10^{-4}$.

Figure 4 illustrates the five conditions. The control design directly answers the objection that the bridge learned the dataset, the format, or the distribution rather than a channel. First, the verdict is computed only on held-out items, so item memorization cannot pass. Second, the shuffled control is identical to real in format, distribution, and prompt construction, differing only in item correspondence—any format- or

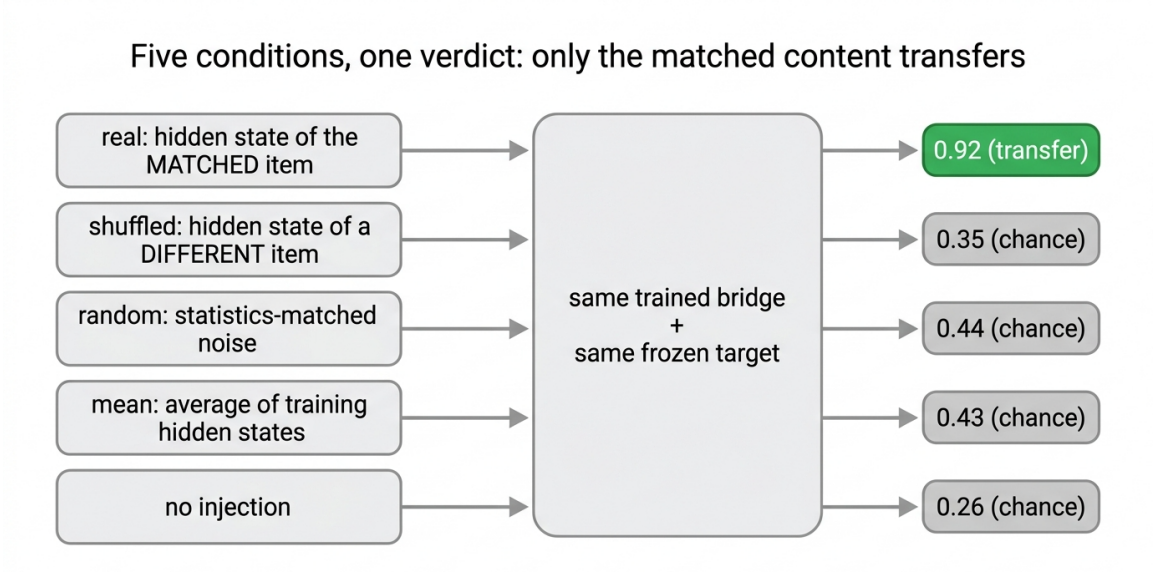

Figure 4: The five control conditions. Only the injected content varies; the trained bridge and the frozen target are identical across conditions. Representative held-out accuracies are 0.925 (real), 0.350 (shuffled), 0.445 (random), 0.430 (mean feature), and 0.255 (no injection). Any format-level shortcut would also raise the shuffled condition, so the real≫shuffled gap isolates per-item content transfer.

distribution-level shortcut would benefit shuffled equally, so the observed real≫shuffled gap is explained only by per-item content transfer. Third, the mean-feature control isolates the effect of a content-free learned vector. Fourth, the error-propagation analysis (Section 6.8) shows that the bridge follows the source's (sometimes incorrect) computation rather than dataset answer priors; and fifth, RACE (Section 6.7) shows the conclusion survives outside the synthetic distribution.

3.6 Speak mode

Besides letter-logprob scoring, we keep the injection in the prefill KV cache and let the target generate 12 tokens greedily, scored by candidate matching—quantifying whether the channel supports *speaking*, not just discrimination.

4 Experiments I: Existence and Localization

The first group of experiments addresses two questions in the most tightly controlled setting, a same-family pair: does the channel exist, and if so, where in the source does it live?

4.1 Experimental setup

The source is Qwen2.5-7B-Instruct (hidden size 3,584; 28 layers); the targets are DeepSeek-R1-Distill-Qwen-1.5B (1,536) and Qwen2.5-0.5B-Instruct (896). The three models share a tokenizer family but differ in size, width, and depth. We first verified the competence gap: given the full prompt, the source scores 0.865 while the targets score 0.27–0.35, so the premise that the source can solve what the target cannot holds. On the Korean version of the tasks even the source scores only 0.478, so all subsequent experiments use the English items.

Table 2: Failure of mid-layer (depth 0.5) injection. Target DeepSeek-1.5B, information-gap items.

variant	real	shuffled	$p(r > s)$	verdict
mlp+additive, K=8 pooled	0.385	0.365	0.146	fail
mlp+additive, K=8 full	0.345	0.335	0.315	fail
mlp+stitch	0.375	0.365	0.345	fail
xattn+additive	0.350	0.405	0.949	fail
xattn+stitch	0.365	0.365	0.581	fail

Table 3: Late-layer (depth 0.85) injection: held-out accuracy under five conditions.

bridge	real	shuffled	random	mean	no_inject	$p(r > s)$
xattn+additive	**0.925**	0.350	0.445	0.430	0.255	$< 10^{-4}$
xattn+stitch	0.910	0.365	0.395	0.420	0.255	$< 10^{-4}$
mlp+stitch	0.905	0.335	0.400	0.410	0.255	$< 10^{-4}$

4.2 The failure of mid-layer injection, and why the information gap is necessary

The natural first attempt reads the source at middle depth (0.5). We tested five combinations of bridge encoder and injection mode in this setting, and, as Table 2 shows, every variant fails the pre-registered criterion: real accuracy stays at 0.345–0.385, statistically indistinguishable from the shuffled control (all $p > 0.14$).

A companion contrast run demonstrates why the information gap itself is necessary. When the target is given the full prompt, no-injection accuracy of 0.365 rises to 0.93 under injection—but with no difference between real and shuffled. Injection acted as a format-stabilizing hint rather than as content transfer, and without the information gap and the shuffled control this format effect would have been mistaken for transfer.

4.3 Opening the channel at late layers

Moving the read location to depth 0.85 reverses the picture. As Table 3 summarizes, all three bridge configurations pass the criterion. In the best configuration (xattn+additive), real accuracy reaches 0.925 while shuffled remains at 0.350, random and mean-feature controls at 0.445 and 0.430, and no-injection at 0.255. That the target recovers the answer only when given the *matched* item's hidden state shows that what is transferred is per-item content, not a format hint. Because the result persists across architectures, the decisive variable is the choice of source layer, not the bridge design.

4.4 Localization in depth

To determine where in the source the transferable information arises, we trained identical bridges while varying the read depth over seven points from 0.30 to 1.00 (K=1, the final token). Figure 5 shows the result. At depths 0.30 and 0.50, real and shuffled are indistinguishable (0.36–0.37, $p > 0.36$); a significant difference first appears at 0.65 (real 0.520, $p = 0.0087$); accuracy then climbs steeply through 0.80 (0.805) and 0.85 (0.910) before saturating near 0.925 up to depth 1.00. No-injection accuracy is flat at 0.255 throughout.

Two conclusions follow. First, the mid-layer failures of Section 4.2 reflect absence of signal rather than limits of the bridge—at those depths there is nothing to read. Second, a single final token (K=1) suffices: the computed answer is recoverable from the last token's hidden state once the question has been fully read (the exception is multi-step arithmetic, which does not fit in one token and would need K>1; Section 8).

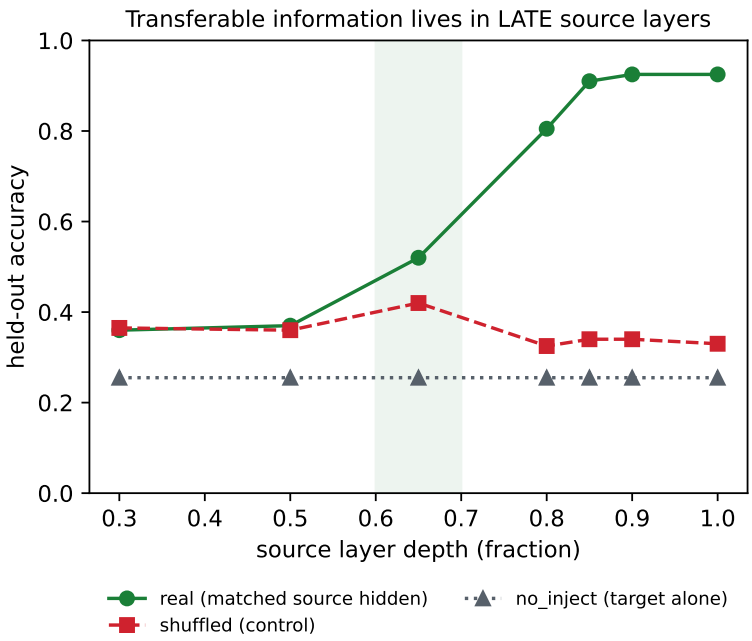

Figure 5: Source-depth sweep. Transferable information appears near depth 0.65 and saturates beyond 0.85.

Table 4: Handoff comparison on identical items. Target DeepSeek-1.5B.

condition	accuracy
partial prompt only	0.220
textual handoff (source argmax inserted)	0.865
hidden-state bridge (5-seed mean / best)	**0.903 / 0.925**
oracle handoff (correct answer inserted)	1.000

4.5 Seed replication

We repeated the best configuration under five independent seeds, each of which re-randomizes the data split, the bridge initialization, and the control-vector extraction. The five runs score 0.915, 0.895, 0.910, 0.890, and 0.905—mean 0.903 (s.d. 0.010)—and every run passes the criterion.

4.6 Comparison with textual handoff

The practical alternative to hidden-state transfer is to hand the source's answer over as text. To make this comparison we scored four conditions on identical partial-prompt items (Table 4). The partial prompt alone yields 0.220; inserting the source's argmax answer as a sentence yields 0.865; an oracle that inserts the correct answer yields 1.000. Since the oracle is perfect, the target's ability to read and copy an inserted sentence is not the bottleneck, and the ceiling of textual handoff is therefore exactly the source's argmax accuracy. The hidden-state bridge scores 0.903–0.925, exceeding that ceiling by 4–6 points.

The excess suggests that the hidden state carries calibrated distributional information that exists before the argmax collapse (discussed in Section 7). A source-correctness decomposition makes this concrete (Table 15): on chain the bridge exceeds the source's argmax (0.885 vs. 0.844) by recovering 29% of the items the argmax gets wrong, whereas on arithmetic—where the hidden does not carry the multi-step computation—it recovers only 5% and stays at the source; overall the channel saturates at the source's latent competence (bridge 0.793

Table 5: All transfer modes on a single setting (multi-task DeepSeek-1.5B, chain, plain prompt, held-out n=1000). The hidden-state bridge exceeds every textual protocol and the direct probe; the oracle is an upper bound, so text-copying is not the bottleneck.

transfer mode	chain accuracy
target alone (information gap)	0.164
textual: argmax handoff	0.844
textual: top-k list	0.549
textual: confidence	0.853
textual: oracle (upper bound)	1.000
direct probe (source hidden → answer)	0.869
hidden-state injection (bridge)	**0.885**

vs. source 0.798). The advantage is thus the distributional information the hidden carries and the argmax text discards, and it does not exceed the source's competence—the "exceeds the ceiling" and "saturates at the source" statements refer to the *argmax* and *latent* ceilings respectively, which differ. Statistically, the primary test is one-sample: the five-seed accuracy (0.903 ± 0.010) against the fixed 0.865 ceiling; a per-item McNemar test (pooled $p = 0.041$, direction consistent in four of five seeds) is reported as supporting evidence. We state the scope of this claim precisely: the comparison is against the single-answer textual handoff—the standard protocol that transmits the source's final answer as one sentence. We also tested richer textual protocols on the same target and split (multi-task DeepSeek-1.5B, plain prompt): verbalizing the source's full distribution as a top-k probability list yields only 0.549 on chain—*worse* than a bare argmax hint (0.844), as the target uses a verbalized distribution poorly—and a confidence-phrased hint yields 0.853; both remain below hidden-state transfer on the same items (0.885). An oracle hint inserting the correct answer reaches 1.000, confirming that text-copying is not the bottleneck. Richer text therefore does not close the gap. The systems advantages of hidden transfer—no source decoding and no re-tokenization—hold regardless of the textual protocol chosen.

4.7 A direct-probe control

One might object that the bridge is merely a supervised decoder of the answer from the source hidden state, with the target contributing nothing. We test this directly by training a classifier on the source hidden state to predict the answer letter, bypassing the target entirely (same held-out split, candidate-masked). A linear probe reaches 0.796 overall and 0.869 on chain; an MLP reaches 0.788; the source's own argmax is 0.798 (0.844 on chain). The bridge through a frozen target matches or slightly exceeds the probe on chain (0.877–0.885 vs. 0.865–0.869), even though the target sees only the partial prompt—so the indirection is lossless: the target recovers the injected content as accurately as a direct readout, and it additionally supports cross-family transfer and speak-mode, which a probe cannot. We do not claim the bridge decodes information a probe cannot; the answer is linearly present in the source hidden, consistent with the localization result. We note that even the linear probe exceeds the source's argmax on chain (0.869 vs. 0.844), independent confirmation that the hidden carries information beyond the argmax.

To place every comparator on a single target and split, Table 5 reports chain accuracy under all transfer modes on the multi-task DeepSeek-1.5B setting (plain prompt, held-out): the hidden-state bridge (0.885) exceeds every textual protocol—including the richest—and the direct probe, while the oracle (1.000) confirms that copying is not the bottleneck. The mechanism (why hidden exceeds the argmax text ceiling) is the distributional recovery quantified in Table 15 (both Qwen and Gemma sources).

5 Experiments II: Family Boundaries and Sovereign Transfer

The second group of experiments pushes the channel across the tokenizer-family boundary: how does it break there, and what must be added to reopen it?

Table 6: Same-family success versus cross-family failure.

target	real	shuffled	no_inject	p	verdict
DeepSeek-1.5B (Qwen family)	0.903 ± 0.010	$\sim$0.36	$\sim$0.22	$< 10^{-4}$	pass
Qwen2.5-0.5B (Qwen family)	0.900	0.350	0.210	$< 10^{-4}$	pass
TinyLlama v1 (default)	0.265	0.265	0.200	1.0	fail
TinyLlama v2 (bottleneck 16, L=4)	0.265	0.265	0.200	1.0	fail
TinyLlama v3 (stitch)	0.260	0.260	0.200	1.0	fail
TinyLlama v4 (target depth 0.75)	0.310	0.305	0.200	0.494	fail

Table 7: Cross-family three-ingredient ablation (all at bottleneck 512, N=3,000).

configuration	real vs shuffled	train loss	verdict
CE only	$0.255 = 0.255$	0.0	fail (memorization)
CE + joint anchor	$0.255 = 0.255$	0.008	fail
CE + soft labels	$0.255 = 0.255$	0.665	fail
CE + joint anchor + soft labels	**0.899 vs 0.358**	—	pass

5.1 Anatomy of the failure

Applying the bridge configuration validated within the family to the Llama-family TinyLlama-1.1B produces no transfer at all. We tried four variants—reduced capacity, substitution-based injection, and a different target depth—and all failed (Table 6), with a common symptom: training loss collapsing to 0.0 (pure memorization) with zero held-out transfer. The contrast—the same bridge generalizes on Qwen-family targets (DeepSeek 0.903, Qwen-0.5B 0.900) but collapses into memorization on a Llama-family target—indicates that tokenizer-family sharing is the key alignment variable.

5.2 Three ingredients that open the boundary

We then removed candidate causes one at a time. Scaling the data fivefold to N=3,000 left the memorization collapse intact. Adding soft labels blocked memorization (training loss 0.665) but still produced no transfer. Pre-training the anchor alignment *before* task training failed as well—the subsequent task cross-entropy destroyed the alignment. The first significant movement appeared when the anchor loss was optimized *jointly* with the task loss ($p = 0.02$), and the channel opened once bridge capacity was also raised from 64 to 512: real 0.899 versus shuffled 0.358 ($p < 10^{-4}$; a single seed, 20260702—seed robustness is examined separately in Section 5.3). The figure itself matches the same-family level (0.903).

A factorial ablation isolates the contribution of each ingredient (Table 7). At capacity 512, task cross-entropy alone collapses into memorization; adding only the anchor lets the enlarged capacity absorb anchor and memorization simultaneously; adding only soft labels lacks any alignment prior. Transfer occurs only when the three are combined—and even the combination fails at 64 dimensions, so capacity is likewise necessary.

Figure 6 depicts the recipe.

5.3 Two faces of seed robustness

The TinyLlama opening is conditional. The successful seed reproduces deterministically under exact re-execution, but five other seeds all fail (one of six). The accurate statement is therefore not that cross-family transfer "opens at 0.9," but that a deterministically reproducible success exists while the optimization is seed-sensitive. As the next subsection shows, however, this fragility is target-dependent rather than universal:

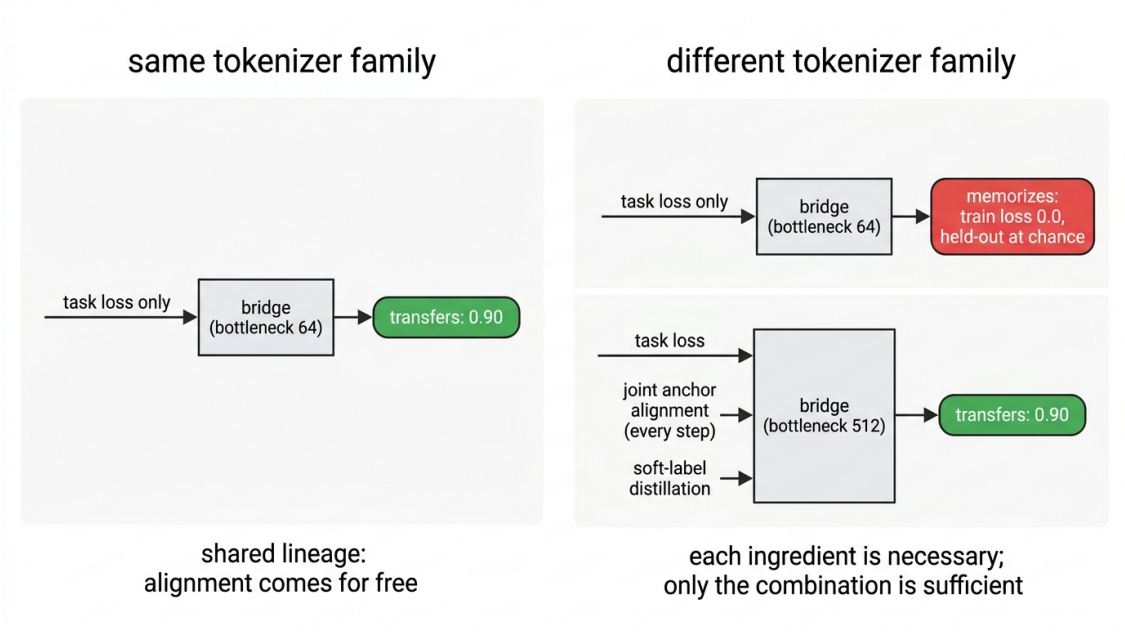

Figure 6: The three ingredients that open the family boundary. Task-loss-only training memorizes at either bottleneck width (64 shown); widening the bottleneck alone does not restore transfer (Table 7). Transfer arises only when 8× capacity, joint anchor alignment, and soft-label distillation are combined.

Table 8: Qwen-7B → EXAONE-3.5-2.4B, all four seeds.

seed	real	shuffled	no_inject	p
20260702	0.890	0.320	0.116	$< 10^{-4}$
7	0.881	0.269	0.120	$< 10^{-4}$
42	0.900	0.271	0.142	$< 10^{-4}$
1234	0.891	0.299	0.120	$< 10^{-4}$

on a larger, multilingual target every seed succeeds. Our interpretation is that the size of the success basin scales with target capacity and multilinguality.

5.4 Sovereign transfer: from Qwen into EXAONE

We applied the three-ingredient configuration to LG EXAONE-3.5-2.4B-Instruct, a Korean-centric model with its own architecture (hidden size 2,560) and its own tokenizer. All four seeds pass the criterion, with mean real accuracy 0.891 (Table 8). A target that scores 0.12 on its own thus speaks correct answers by receiving, as hidden states, the internal reasoning of a foreign larger model.

Table 9 shows what this transfer looks like at the level of a single item. In a five-premise item whose true ordering is A > E > D > C > B, EXAONE received only two premises (C < D, B < C)—every sentence mentioning A or E was removed. Alone, EXAONE answers D, the best guess available from its own information; through the bridge it answers A, an entity that never appears in its input. No source for this correct answer exists other than the source model's hidden state. On the same six-item demonstration, EXAONE alone answered zero items and the bridged system all six, consistent with the exhaustive held-out evaluation (0.116–0.142 alone versus 0.881–0.900 bridged).

Table 9: Item-level record of sovereign transfer (a held-out length-5 chain item, actual record).

true ordering	A > E > D > C > B
what Qwen-7B saw (full)	D<E, C<D, E<A, B<C, B<A (five premises)
what EXAONE saw (partial)	C<D, B<C — all sentences mentioning A or E removed
question	Which is the biggest?
EXAONE alone	D (wrong—best available from its own information)
EXAONE + bridge	**A (correct—an entity absent from its input)**

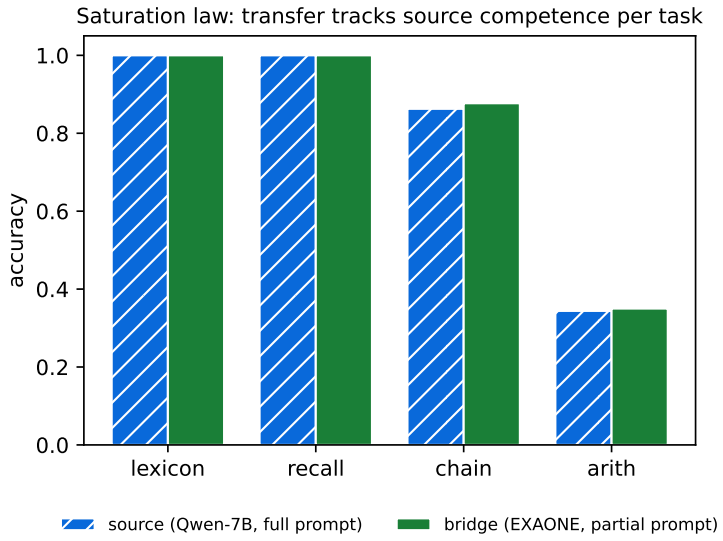

Figure 7: The saturation law. Per-task bridge accuracy (spoken by EXAONE) tracks source competence.

6 Experiments III: Multi-Task Transfer and the Saturation Law

The third group of experiments asks what, and how much, the channel carries. As we widen the task set to four families, vary the source's scale and family, and introduce a real benchmark, a single regularity keeps reappearing: the ceiling of transfer is the competence of the source.

6.1 A multi-task sovereign channel

On 3,000 items spanning comparison chains, multi-step arithmetic (arith), key–value recall, and rare-word mapping (lexicon), we trained Qwen-7B→EXAONE bridges under three seeds. All three runs pass the criterion (real 0.801/0.806/0.800; shuffled ≈0.233), with mean 0.802.

6.2 The saturation law

The value 0.802 deserves attention: on the same items, the source's own accuracy is exactly 0.802, and as Figure 7 shows the bridge tracks the source per task—lexicon 1.0 → 1.0, recall 1.0 → 1.0, chain 0.863 → 0.877, arith 0.344 → 0.350. The channel carries essentially what the source computed, saturating at its latent competence; the small positive margin on chain reflects recovery of items the source's argmax gets wrong (Section 4.6), not amplification beyond the source.

Table 10: Actual speak-mode generations (EXAONE, greedy decoding).

partial information seen by target	question (answer)	alone	bridged
C<A, C>D	biggest (B)	"A"	**"B"** (B absent from input)
D>A	smallest (B)	"A"	**"B"**
...	biggest (E)	"D"	**"E"**

6.3 Bandwidth is not the bottleneck

If the ceiling is source competence, adding bandwidth should not help. Indeed, injecting simultaneously at three target depths (0.35/0.5/0.65) yields 0.804, and adding sequence distillation yields 0.802—both indistinguishable from single-layer injection. A channel already at the source ceiling has nothing left to gain.

6.4 Speak mode

Generation-based scoring leads to the same conclusion. Keeping the injection in the prefill cache and decoding twelve tokens greedily gives real accuracy of 0.755–0.860 against 0.17–0.24 without injection: the channel supports speaking, not only discrimination. Table 10 reproduces actual generations: only with the bridge does the target utter answer entities absent from its input.

6.5 The source-scale axis

Scaling the source to Qwen-14B (own accuracy 0.839) reveals an informative refinement of the saturation law. On chain, where 14B beats 7B, the improvement passes through the bridge intact (0.877 to 0.947). On arith, the source rises from 0.344 to 0.416, yet the bridge stays at 0.346. The intermediate computations of multi-step arithmetic appear not to compress into a single final-token hidden state (multi-token injection, $K>1$, is left to future work); the benefit of scaling is transmitted selectively by task type.

6.6 Target diversification

All four combinations of 7B, 14B sources with DeepSeek-1.5B, Qwen-0.5B targets pass the criterion (0.799–0.823; Appendix E), and the 14B chain gain (+0.07–0.08) appears consistently on both targets. The exception is TinyLlama, where the 14B source collapses from 0.795 to 0.383 at the same seed; we attribute this not to an adverse effect of source scale but to the cross-family seed sensitivity identified in Section 5.3 acting as a confounder.

6.7 A real benchmark: RACE with an information gap

To address the concern that our tasks are synthetic, we converted 1,000 RACE:middle reading-comprehension items into information-gap form by removing the passage, and ran the identical pipeline. Real accuracy reaches 0.889—98.5% of the source's 0.903—against 0.267 for shuffled and 0.443 for no-injection. The fact that shuffled falls *below* no-injection is particularly telling: injecting the wrong content is worse than injecting nothing. This is the most direct evidence that what the channel carries is content rather than form.

6.8 Propagation of source errors

The channel's fidelity is most visible when the source is wrong. Decomposing items by source correctness, the bridge answers correctly on 0.96–0.99 of items the source got right, but on only 0.10–0.16 of items the source got wrong (consistent across six runs). On RACE, although the target alone reaches 0.443, bridged accuracy on source-wrong items drops to 0.132—the target follows the source's (incorrect) computation in preference to its own judgment. The channel faithfully transports the source's computation even when it is mistaken.

Table 11: Gemma-2-9B source matrix (all pairs cross-family, three-ingredient configuration).

pair	real	shuffled	no_inject	speak
Gemma→EXAONE (seed 20260702)	0.794	0.234	0.210	0.765
Gemma→EXAONE (seed 7)	0.802	0.220	0.246	0.835
Gemma→EXAONE (seed 42)	0.799	0.235	0.237	0.835
Gemma→DeepSeek-1.5B	0.780	0.219	0.200	0.735
Gemma→Qwen2.5-0.5B	0.790	0.232	0.194	0.775

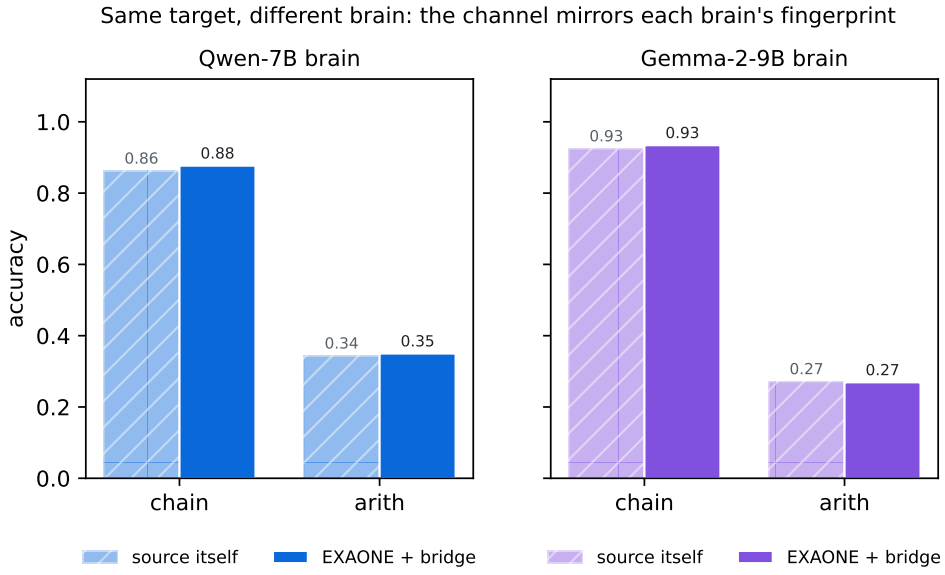

Figure 8: Fingerprint inheritance. The same target (EXAONE) reproduces the task profile of whichever source is connected.

6.9 Source-family diversification and the brain fingerprint

Because every source so far was Qwen-family, the observed channel could in principle be a peculiarity of Qwen's hidden-state geometry. To test this, we ran the identical pipeline with Gemma-2-9B-it—an architecturally distinct model with interleaved local/global attention and logit softcapping—as the source. With a Gemma source every pair is cross-family, so the three-ingredient configuration was used throughout.

Gemma's own competence is 0.7993 overall, tied with Qwen-7B (0.802) but differently distributed: chain reaches 0.9253, close to Qwen-14B, while arith falls to 0.272, below Qwen-7B's 0.344. The two sources have equal totals and different fingerprints.

As Table 11 shows, all five pairs pass the criterion. The three EXAONE seeds average 0.798, within one standard deviation (0.004) of the 0.7993 source ceiling—saturation survives the change of family. The source-correctness decomposition likewise matches the Qwen-source range (Appendix E).

The most striking result is the per-task decomposition (Figure 8). The same EXAONE target scores chain 0.877 and arith 0.350 under a Qwen-7B source, but chain 0.934 and arith 0.269 under a Gemma source—inheriting each source's strengths and weaknesses intact. What determines the target's per-task performance is neither the bridge nor the target, but which source is connected. The saturation law thus holds along the family axis as it does along the scale axis (Section 6.5), and, as a corollary, the channel can serve as a diagnostic readout of a source model's competence profile.

7 Analysis and Discussion

Our results admit several readings. Consider first why the channel opens only in the late layers. The localization curve (onset 0.65 → saturation 0.85) is consistent with the logit-lens lineage: transformers *compute* the answer at intermediate depths and *format* it late. At mid depth the answer has not yet crystallized, so the bridge has nothing to read (Section 4.2); success is therefore decided by where you read, not by bridge expressiveness. The same reasoning explains why the hidden state beats text. Textual handoff collapses the source's distribution to a single argmax point, whereas the hidden state retains calibrated confidence over candidates, so on items where the source is uncertain the target can combine this with its own prior to beat the argmax. A systems benefit compounds this: the source needs only a single prefill (no decoding), and the re-tokenization round-trip disappears.

The cost of crossing a family boundary translates this picture into the language of optimization. Cross-family failure has two causes: an optimization landscape in which task cross-entropy leaks into memorization more easily than into semantic alignment, and insufficient capacity for the change of basis. The former is addressed by the joint anchor, which forces a "translator" solution throughout training, and the latter by 8× capacity. The value of shared lineage is thereby quantified in the language of optimization landscapes.

Running through all of these observations is the saturation law. Within the range of structured tasks we observe, the channel is not an amplifier: neither multi-layer injection nor distillation exceeds the source ceiling, and the channel faithfully carries the source's errors. This fidelity cuts both ways—trustworthy delivery, but also a propagation path for source mistakes—so anyone designing an application should treat source competence as the system ceiling. One such application is the sovereign configuration: the EXAONE result shows that "a foreign frontier model reasons internally while a domestic model speaks" is technically viable. The scope here is structured tasks; quantitative evaluation of open-ended natural-language answering remains future work (qualitative demos in Appendix D).

8 Limitations

The quantitative claims of this work rest on the transfer of multiple-choice answer representations. Transfer of arbitrary semantic content is supported only qualitatively by the open-ended prototype (keyword overlap 0.865 vs 0.320 alone), and precise control of format, language choice, and numeric chains remains unresolved. The hidden-over-text advantage of Section 4.6 is scoped to letter-answer tasks; there, richer textual protocols that verbalize the source's confidence distribution (top-k lists, stated confidence) do not close the gap—and a verbalized top-k list in fact underperforms a bare argmax hint—but whether some other verbalization narrows it on other task types remains open.

A second limitation is the channel's capacity. Multi-step arithmetic does not fit in a single final-token hidden state, invariant across source, target, scale, and family. Crossing this compression bottleneck would require multi-token injection (K>1), which is future work.

A third is coverage. Sources span two families (Qwen, Gemma) and two scales (7B/14B), and targets reach at most 2.4B. A 32B source and 7B-class targets exceed single-A100 memory for bridge backpropagation and were not evaluated, so completing the scale curve remains future work.

Finally, cross-family optimization is seed-sensitive on small targets (Section 5.3). Because every seed succeeds on a larger, multilingual target, this is not a universal fragility, but stabilizing it on small targets is an open problem.

9 Reproducibility Statement

To remove the main barrier—the 15–18GB source model—we release a public bundle of precomputed source features and trained bridges (features for 4 sources, 21 bridges, data, checksums, and a model card; 639MB). Three reproduction tiers are supported. T1 (laptop CPU, no training): the bundle plus a public 0.5B target reproduces the core claim; we verified this tier directly (Qwen-7B source: real 0.797 against 0.799 in the paper; Gemma-9B source: 0.790, matching the paper, including the per-task fingerprint; $p < 10^{-4}$). T2

Table 12: T1 reproduction check (laptop CPU, no source model, held-out n=1,000).

source (released features)	bridge	no_inject	shuffled	real	paper
Qwen-7B	same-family, b=64	0.164	0.233	0.797	0.799
Gemma-2-9B	cross-family, b=512	0.193	0.231	0.790	0.790

(one GPU): retrain bridges from released features and anchors (8 frozen configuration files provided). T3 (A100-class GPU): the full pipeline from raw models, including source-side extraction. Table 12 summarizes the T1 verification.

Data generators are deterministic (seed 20260702), so the released features are a convenience, not a root of trust—anyone holding the source model can regenerate them and obtain identical numbers. The decision criterion was pre-registered and never relaxed; all per-item artifacts are preserved with SHA-256 verification. The bundle, code, and data generators will be released with a DOI upon publication.

10 Conclusion

A learnable channel exists that carries task-resolving information between frozen language models as hidden states. The channel is localized to the final token of the source's late layers; it exceeds the argmax ceiling of textual handoff; it is a lossless carrier that saturates at source competence, faithfully transmitting the source's strengths, weaknesses, and even its errors; and the family boundary can be crossed with three ingredients—capacity, alignment, distillation—as demonstrated by a Korean model speaking the reasoning of a foreign one. That the channel's behavior is this regular suggests that model-to-model collaboration without text is a proper subject for engineering.

Broader Impact Statement

This work characterizes an inference-time information channel between frozen language models. Positive uses include efficiency (no decoding round-trips) and sovereign deployments that keep a domestic model as the user-facing endpoint. The same fidelity that makes the channel useful also propagates source errors and, potentially, source biases into the target's outputs; deployments should treat the source's competence and failure modes as the system's own. The released artifacts contain no personal data; all datasets are synthetic or derived from the public RACE benchmark.

References

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations (ICLR)*, 2023.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Adrián Csiszárik, Péter Kőrösi-Szabó, Ákos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. `https://transformer-circuits.pub/2021/framework/index.html`.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *International Conference on Machine Learning (ICML)*, 2024.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *International Conference on Machine Learning (ICML)*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.

Rishi Jha, Collin Zhang, Vitaly Shmatikov, and John X. Morris. Harnessing the universal geometry of embeddings. *arXiv preprint arXiv:2505.12540*, 2025.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, 2019.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, 2023a.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. Linearly mapping from image to text space. In *International Conference on Learning Representations (ICLR)*, 2023.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *International Conference on Learning Representations (ICLR)*, 2023.

nostalgebraist. Interpreting GPT: the logit lens. LessWrong, 2020. `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Vignav Ramesh and Kenneth Li. Communicating activations between language model agents. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2501.14082.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, et al. On transferability of prompt tuning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2022.

Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Guangxuan Xiao, Ji Lin, and Song Han. Offsite-tuning: Transfer learning without full model. *arXiv preprint arXiv:2302.04870*, 2023.

Xiyuan Yang, Jiaru Zou, Rui Pan, Ruizhong Qiu, Pan Lu, Shizhe Diao, Jindong Jiang, Hanghang Tong, Tong Zhang, Markus J. Buehler, Jingrui He, and James Zou. Recursive multi-agent systems. *arXiv preprint arXiv:2604.25917*, 2026.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Task Generation and Shortcut Mitigation

Entities are abstract symbols A–E; chain lengths 3/4/5 with 200 items each. Premises are fragments of a total order of the form "X is bigger than Y". Five shortcut mitigations are enforced by the generator: (1) uniform answer letters (120 per letter), defeating letter-prior heuristics; (2) balanced max/min questions (300/300); (3) shuffled premise order, defeating positional heuristics; (4) randomized relation direction ("X bigger than Y" ↔ "Y smaller than X"), defeating surface-form heuristics; (5) 0–2 redundant sentences, defeating mention-frequency heuristics. The partial prompt removes every sentence mentioning the answer entity and one distractor entity, and the generator verifies that the answer is logically undetermined afterwards. The multi-task set (3,000 items) adds arith (2–3-step arithmetic), recall (key–value), and lexicon (rare-word mapping) under the same information-gap principle (750 per task). The RACE variant removes the entire passage from the partial prompt of RACE:middle passage–question pairs (options retained). Scoring is argmax over candidate-letter next-token log-probabilities, summing the " A"/"A" encodings by logsumexp. All generation is deterministic at seed 20260702.

B Hyperparameters

Table 13: Full hyperparameters of the hero and C-configurations.

item	hero (same-family)	C-configuration (cross-family)
bridge encoder	xattn (L learned queries)	xattn
bottleneck b	64	**512**
read (K, depth)	K=1, source 0.85	same
inject (L, depth, mode)	L=8, target 0.5, additive (0.02 init)	same
losses	CE	CE + anchor MSE (w=1.0, joint) + soft KL (α=0.5, T=2)
optimizer	AdamW, lr 10^{-3}, wd 0.01	same
steps / batch	600 / 16 (N=600)	2,500 / 16 (N=3,000)
split	train 2/3, held-out 1/3 (fixed seed)	same
precision	target bf16 (TinyLlama/EXAONE fp32)	fp32
statistics	paired sign-flip permutation, 10,000 resamples	same

The bridge has ∼2.4M parameters (hero configuration). The released model card lists the configuration and held-out numbers of each of the 21 bridges.

C Environment and Pipeline

Remote: one A100-PCIE-40GB; Python 3.8; PyTorch 1.12 (NGC build); transformers 4.40.2 (EXAONE and Gemma stages load 4.43.4 through an isolated PYTHONPATH); no internet access. Source-feature extraction also reproduces on a local CPU (bf16, 45.8GB RAM; 12–16 min per 600 items). Four compatibility shims support the legacy runtime: a bf16 triu/tril workaround; casting of Python scalars in `torch.where` against double promotion; bypassing the version gate of the GELU-tanh activation; and manual chat-template construction under jinja2<3.1 (verified string-identical to the original template). Upload (chunked TUS), remote execution, and artifact retrieval are fully CLI-automated, and all retrieved per-item artifacts are SHA-256-verified and preserved.

D Qualitative Demonstrations and Failure Taxonomy

Structured demo. EXAONE alone answered 0 of 6 items; through the bridge it answered 6 of 6 on the same items. In speak mode the target generates the correct answer entity—absent from its input—only when the injection is present.

Open-ended demo. A sequence-distilled bridge (2,157 free-form Q/A pairs) evaluated on 200 open-ended questions yields keyword overlap 0.865 (bridged) vs 0.320 (alone). Figure 9 reproduces an actual transcript: the source reasons internally in Chinese about a pragmatics question (pronoun resolution), and EXAONE speaks the correct answer in Korean—the channel crosses the language boundary at the level of internal representations.

Q (Korean)> 민수는철수에게자기가방을가져오라고했습니다. "자기" 는누구를가리킬가능성이높나?
(Minsu asked Cheolsu to bring "his own" bag. Whom does "his own" most likely refer to?)
Qwen (source, internal): 在這句話中, "自己" 更有可能是指 "民洙"... *(reasoning in Chinese)*
EXAONE + bridge: 여기서 "자기" 는민수를가리킬가능성이더높습니다 ... *(correct, in Korean)*

Figure 9: Language-boundary crossing in open-ended transfer (actual transcript).

Conversely, using a task-specialized bridge (trained on letter answers) for open-ended questions fails in characteristic ways (Figure 10), showing that such a bridge is a "hidden-to-answer-letter" decoder rather than a general translator—which motivated the open-ended distillation design.

Q (Korean)> 한복은어디꺼야? *(Where is hanbok from?)*
EXAONE alone: 한국 *(Korea)*
EXAONE + bridge: A) A) 한복은한국의전통의상입니다. *(content correct; multiple-choice format leaks)*
Q (Korean)> 한복은어느나라옷이야? *(Which country's clothing is hanbok?)*
EXAONE + bridge: A) 미국 *(USA)* *(out-of-distribution projection induces an error)*

Figure 10: Failure modes of a task-specialized bridge on open-ended questions (actual transcript).

Failure taxonomy (F1–F5, with observed instances). F1 non-word generations ("in one word: the capital is?" answered with non-words); F2 injection harming items the target solves alone (an arithmetic item with answer 13: 18 alone, 48 bridged); F3 fluent but fabricated rule descriptions; F4 source-language leakage (bridged answers in Chinese); F5 non-compliance with format instructions ("answer in exactly one word" ignored). The limitation stated in Section 8—gist and approach transfer reliably, precise control does not—rests on this taxonomy. Keyword overlap partially credits fluent errors and must not be confused with accuracy.

E Additional Tables

Across same-family targets, every source–target pair passes the criterion, and the 14B source's chain gains transmit intact (Table 14).

Table 14: Source × target matrix (same-family, hero config; one seed per cell, hero pair 7B→DeepSeek replicated over five seeds in Section 4.5).

source→target	real	shuffled	no_inject	verdict	chain	arith
7B→DeepSeek-1.5B	0.804	0.259	0.177	pass	0.885	0.354
14B→DeepSeek-1.5B	0.823	0.242	0.177	pass	0.951	0.365
7B→Qwen-0.5B	0.799	0.234	0.174	pass	0.877	0.342
14B→Qwen-0.5B	0.820	0.217	0.174	pass	0.959	0.346

Decomposing held-out accuracy by whether the source is itself correct shows the channel recovers only a task-dependent fraction of the items the source gets wrong (Table 15)—the fraction that lets hidden transfer

exceed the argmax text ceiling while remaining bounded by source competence, consistent across both source families.

Table 15: Decomposition by source correctness (Gemma and Qwen sources, held-out). The last column is the fraction of source-argmax-wrong items the channel recovers; bridge/argmax accuracies for the Qwen setting appear in Table 5.

setting	P(bridge | source correct)	P(bridge | source argmax wrong)
Gemma→EXAONE	0.981	0.099
Gemma→DeepSeek	0.963	0.099
Gemma→Qwen-0.5B	0.981	0.080
Qwen-7B→Qwen-0.5B (overall)	0.969	0.099
Qwen-7B→Qwen-0.5B, chain	—	**0.290**
Qwen-7B→Qwen-0.5B, arith	—	0.055

All speak-mode results (real / no_inject, n=200). Qwen7B→EXAONE, 3 seeds: 0.755/0.195, 0.810/0.235, 0.860/0.170; multi-layer 0.775/0.195; sequence-distilled 0.755/0.195; Qwen7B→Qwen0.5 0.765/0.135; Gemma9B→EXAONE, 3 seeds: 0.765/0.195, 0.835/0.235, 0.835/0.170; Gemma9B→DeepSeek 0.735/0.000; Gemma9B→Qwen0.5 0.775/0.130; RACE 0.880 (bridged).