

UNIVERSITÉ DU QUÉBEC EN OUTAOUAIS

DÉPARTEMENT DES SCIENCES ADMINISTRATIVES

WORKING PAPER NO. AI-0001

The Neural Organ Transplant Experiment

Causal Transfer of Cognitive Capabilities Across Local Large Language Models

Simon-Pierre Boucher

Département des sciences administratives
Université du Québec en Outaouais

simon-pierre.boucher@uqo.ca

Gatineau – Pavillon Alexandre-Taché
283, boulevard Alexandre-Taché
Gatineau, Québec, Canada J9A 1L8

November 2026

Version 1.0

Abstract

Can a bounded, causally identified subsystem of one language model be transplanted into another model that lacks the corresponding capability, without training? We answer in a setting where the question is identifiable: decoder transformers trained from scratch on an invented operator language, a recipient that never saw one operator, interchange interventions that localize where the donor executes it, and six transplant families tested against sham, random, shuffled and unrelated-donor controls and against conventional learning on the same items. In a preregistered confirmatory run, a donor-in-the-loop activation graft—one residual row at the read position, mapped through an orthogonal Procrustes transform fitted on task-free data—lifts the recipient from chance to 0.966 on held-out programs (control-adjusted effect $+0.875$, lower bound above the smallest effect of interest in 9 of 9 donor–recipient pairs, control stratum 0.992 with one pair below the 0.95 retention criterion), carries the rule to unseen operands, needs eight of 128 dimensions, survives width, depth and composition-depth mismatches, and is anchored by ablation and rescue. The boundary is as informative as the effect: maps fitted by regression toward the recipient annihilate exactly the capability-specific directions (we prove this for ridge and CCA), donor-free adapters cache rather than compute, weight grafts damage retained skills, steering vectors transfer nothing, and a supervised adapter of equal size is perfect. On Qwen2.5-1.5B/3B/7B and Llama-3.2-3B the same protocol finds task vectors that are causally sufficient within a model for antonyms and capitals, transfers them between models above null controls in some pairs and directions under a frozen selection rule, and fails for tasks that no model represents as a vector. The contribution is a causal, preregistered and fully controlled account of what activation-level transplantation can and cannot move.

Keywords: mechanistic interpretability, activation patching, model stitching, representation alignment, knowledge transfer, causal abstraction, open-weight language models

JEL Classification: C45, C63, C88, O33

1. Introduction

Independently trained language models appear to converge toward similar linear representation geometries (Huh et al., 2024; Kornblith et al., 2019; Raghu et al., 2017), and affine maps between the residual streams of related models transfer probes, sparse-autoencoder dictionaries and steering vectors (Chen et al., 2025; Kim et al., 2026; Agarwal, 2026). Whether such maps can carry a *capability*—a reusable computation that the recipient could not perform before—is open. The one controlled attempt we know of reports a null: a linear bridge between two Pythia models aligns hidden states almost perfectly yet does not improve multi-hop answering when injected (Zhang and Xin, 2026).

This paper asks the question under conditions where it can be answered. We (i) *manufacture* a capability gap by training donor and recipient transformers on the same micro-world except for one invented operator Ω that the recipient never sees, (ii) *causally localize* where the donor executes Ω by interchange interventions before attempting any transfer, and (iii) compare six transplant families against a battery of matched controls (sham map, norm-matched random rows, shuffled items, shuffled layers, unrelated-skill donors) and against conventional learning on the same data. The word *organ* is a metaphor for a bounded, causally identified subsystem; it carries no biological or phenomenal claim. We then test the same protocol on three open-weight models (Qwen2.5-1.5B, Qwen2.5-3B, Llama-3.2-3B) with the capability defined as performing a verifiable task without demonstrations.

What we find. In the micro-world the answer is positive, specific and bounded. A donor-in-the-loop *activation graft*—replacing the recipient's post-layer-0 residual row at every read position by the donor's row through an orthogonal Procrustes map fitted on *task-free* rows—lifts the recipient from chance to near-perfect accuracy on held-out Ω programs (confirmatory: Δ_T^{adj} reported in §5) while leaving the control stratum intact, for same- and different-initialization pairs, across a width mismatch, a depth mismatch and a harder stride-2 variant. The graft carries the *rule*: maps fitted on six of the eleven operand values transfer to the five unseen values. An eight-dimensional subspace of the 128-dimensional residual stream suffices; a random subspace of the same size does nothing. Ablating the donor's layer-0 attention output destroys the donor; grafting the same component from an intact donor of another seed rescues it; a sham map does not. All controls sit at chance, and the sham map and random rows also destroy the control stratum, so retention is itself diagnostic.

Three findings bound the claim. First, the choice of alignment map is not a detail: maps fitted by regression toward the recipient on rows where the donor computes Ω *remove* the Ω-specific component (Proposition 1), which is why ridge and CCA grafts fail where orthogonal and task-free maps succeed, and why rescaling or more data cannot help them. Second, donor-free variants are

weaker and qualitatively different: a saved affine adapter conveys part of the effect in a fresh process but behaves as a cache (zero on unseen operands); weight grafts partly transfer Ω at the cost of the control stratum; steering vectors transfer nothing. Third, on this task a supervised adapter with the same parameter count, or distillation on the same items, reaches perfect accuracy: transplantation is a way to move a computation *without* labels or gradient steps, not a cheaper way to learn. On open-weight models, task vectors are causally sufficient within a model for antonyms and capitals but not for an invented morphology, and cross-model transfer of mapped vectors is *reported as measured* in §5; we do not extrapolate from the micro-world to them.

Contributions. (1) An operational theory of transplantable subsystems with explicit estimands, a control battery and a persistence vocabulary (§3), including an elementary annihilation result for regression-fitted alignment maps. (2) A controlled micro-world and an open-weight framework with exact hook semantics, preregistered hypotheses and a registry of every run (§4). (3) Confirmatory evidence on localization, sufficiency, necessity, rescue, selectivity, dimensionality, rule-versus-cache and portability of an activation graft in the micro-world (§5–§6). (4) A map of the failure boundaries: regression-fitted maps, donor-free adapters, weight grafts and steering vectors, and the open-weight task-vector transfer (§7–§8).

2. Related Work

Representation similarity and stitching. SVCCA (Raghu et al., 2017), CKA (Kornblith et al., 2019), the Platonic Representation Hypothesis (Huh et al., 2024) and multi-way alignment (Achara et al., 2026) quantify cross-model similarity; stitching (Bansal et al., 2021) turns similarity into a functional test. For language models, affine residual-stream maps transfer SAEs, probes and steering vectors across sizes within a family (Chen et al., 2025), and a shared "activation bus" lets one model's intermediate activation be consumed by another's upper layers (Kim et al., 2026). Shared geometry of unembeddings supports steering transfer (Lee et al., 2025). Our work differs in starting from a causally identified component and in measuring acquisition against matched controls rather than representational agreement.

Cross-model feature dictionaries and steering transfer. Universal and shared SAEs (Lan et al., 2024; Thasarathan et al., 2025; Nasiri-Sarvi et al., 2025; Ognev et al., 2026) build concept spaces across models; cross-architecture steering transfer (Agarwal, 2026) and refusal-circuit transfer (Cristofano, 2026) move *behavioural directions*. RL-induced reasoning vectors transfer only within a family (Sinii et al., 2025). These concern concept- or behaviour-level control; we test capability acquisition on verifiable tasks.

Capability transfer and merging. Skill localization by grafting (Panigrahi et al., 2023) operates within one model. Activation-guided pruning and merging across scales (Fan et al., 2026) and latent semantic alignment with layer-wise training (Gu et al., 2025) transfer knowledge across sizes; task arithmetic (Ilharco et al., 2022), TIES (Yadav et al., 2023) and re-basin (Ainsworth et al., 2022) merge weights sharing an initialization. Tokenizer transfer (Minixhofer et al., 2024) addresses the vocabulary boundary.

Causal methods. We rely on causal mediation and tracing (Vig et al., 2020; Meng et al., 2022), interchange interventions and causal abstraction (Geiger et al., 2023a,b), circuit discovery (Wang et al., 2022; Conmy et al., 2023), function and task vectors (Todd et al., 2023; Hendel et al., 2023), activation steering (Turner et al., 2023; Li et al., 2023; Zou et al., 2023) and sparse dictionaries (Cunningham et al., 2023; Gurnee et al., 2023). Massive activations (Sun et al., 2024) and concept relearning after removal (Lo et al., 2024) inform our ablation and persistence tests.

Closest negative result. Zhang and Xin (2026) find that a well-fitting linear bridge does not yield useful causal communication inside the receiver. We treat this as the default expectation and design controls that can distinguish "no transfer" from "transfer masked by misplacement".

3. Operational Theory

3.1. Objects

Let D and R be donor and recipient models, $\mathscr{T}$ a task distribution with a target stratum (items that require the capability) and control strata (items that do not), S_D a candidate subsystem of D (a set of layers, heads, MLP units, residual subspaces or sparse features), A an alignment operator mapping donor-side quantities into recipient coordinates, and I an intervention operator. The transplanted recipient is $R' = I(R, A(S_D))$. Interventions act on the residual stream at *read positions*: the token positions whose next-token prediction is scored.

3.2. Estimands

With $Q_T(\cdot)$ the exact-match score on held-out target items and $Q_c(\cdot)$ on control stratum c,

$$\Delta_T = Q_T(R') - Q_T(R), \qquad \Delta_T^{\text{adj}} = \Delta_T - \big[Q_T(R_{\text{sham}}) - Q_T(R)\big], \tag{1}$$

$$\rho_c = Q_c(R')/Q_c(R) \text{ (if } Q_c(R) \geq 0.5, \text{ else report } \Delta_c), \qquad N_D = Q_T(D) - Q_T(D \setminus S_D). \tag{2}$$

R_{sham} is the matched sham intervention: identical positions, layers, dimensionality and activation norm, with the donor content replaced by the control defined for each graft family (for mapped

Table 1. Transplant families, their inference-time dependency on the donor, what is fitted, and the matched control used for the control-adjusted effect.

Family	donor at inference	fitted object (data)	matched control
Activation graft	yes (same token ids)	map A (task-free or task rows)	sham map (random orthogonal)
Subspace graft	yes	$A + k$ principal directions	random k-subspace
Steering graft	no (vector precomputed)	A + mean-difference direction	norm-matched random vector
Adapter graft	no	affine $f : x_R \to A(x_D)$ (ridge)	supervised adapter with labels (reference)
Weight graft	no	none (copy of blocks / rows)	shuffled-layer / unrelated-donor weights
Persistent graft	no	saved adapter, fresh process	unmodified recipient on new items

activation grafts, the same donor activation through a random orthogonal map). Items are nested in generating programs (clusters); uncertainty is a cluster bootstrap over programs within a model pair, and model pairs form a second level summarized by mean, standard deviation and range.

3.3. Definitions

Candidate organ: a component whose perturbation produces repeatable, task-selective causal effects in D (necessity N_D, restored-answer rate under interchange, and rescue by re-insertion). *Transplant*: a predeclared intervention I that injects, replaces, composes or maps $A(S_D)$ into R. *Acquisition*: Δ_T^{adj} above the smallest effect of interest on held-out items, with retention and no leakage. *Selectivity*: target improvement relative to control strata and matched interventions. *Persistence*: single-token, within-sequence, cross-prompt, or saved checkpoint, reported separately; a donor-in-the-loop graft is an inference-time dependency, not a saved modification. *Portability*: replication across seeds, initializations, widths, depths, families and tokenizers.

3.4. Transplant families

We implement six families (Table 1).

The *activation graft* replaces the recipient's residual row at a read position by a mapped donor row computed on the same token ids ($x_R \leftarrow x_R + \alpha(A(x_D) - x_R)$); the *subspace graft* restricts the replacement to a k-dimensional subspace; the *steering graft* adds a mapped mean-difference direction; the *adapter graft* is a donor-free affine interface f fitted offline to predict $A(x_D)$ from x_R; the *weight graft* copies shape-compatible parameter blocks; the *persistent graft* saves an adapter and is evaluated in a separate process on new items.

3.5. What alignment maps can and cannot carry

A natural choice for A is a least-squares map fitted to predict recipient activations from donor activations on paired inputs. The following elementary statement, proved in Appendix A, explains

an empirical regularity of §5: such maps annihilate exactly the donor directions that carry what the recipient lacks.

Proposition 1 (Annihilation of recipient-unpredictable directions)**.** *Let $X \in \mathbb{R}^{n \times d_D}$ and $Y \in \mathbb{R}^{n \times d_R}$ be centred paired activations and let $X = X_c + X_u$ be an orthogonal decomposition of the donor space into a subspace U and its complement with $X_u^\top X_c = 0$ and $X_u^\top Y = 0$ (the coordinates of X in U are uncorrelated with every recipient coordinate). Then every ridge solution $W_\lambda = (X^\top X + \lambda I)^{-1} X^\top Y$, $\lambda \geq 0$, satisfies $P_U W_\lambda = 0$, where P_U is the orthogonal projector onto U: the map sends every component of a donor activation in U to zero. The same holds for the least-squares map restricted to the top-k canonical directions (CCA), because U has zero canonical correlation with Y. In contrast, any orthogonal map Q (e.g. the Procrustes solution, with or without isotropic scale) is injective and satisfies $\|Q^\top u\| = \|u\|$ for all $u \in U$.*

The proposition does not say that orthogonal maps place the preserved component where the recipient can *use* it; that is an empirical question answered by the interchange experiments. It does say that fitting A by regression on rows where the donor computes the target capability and the recipient does not is self-defeating by construction: the capability-specific variance is, by hypothesis, unpredictable from the recipient and is removed. Two remedies follow: fit A on *task-free* rows (so that the donor's capability-specific directions never enter the regression as unpredictable variance), or constrain A to be orthogonal. Both are tested.

3.6. What is a theorem and what is a conjecture

Proposition 1 is a statement about linear regression; nothing in it depends on transformers. The interchange-commutation property under exact causal abstraction (Geiger et al., 2023a) is restated with its assumptions in Appendix A; everything about real models is empirical. **Conjecture (transfer boundary).** An activation graft through a linear map succeeds only if the recipient's downstream computation reads the mapped subspace in the donor's role; this is the default expectation motivated by Zhang and Xin (2026) and is tested, not assumed.

4. Experimental Design

4.1. Tier 0: a controlled micro-world with a manufactured gap

Task (OPSEQ). A program $\langle \text{bos} \rangle\, v_0\, o_1 \ldots o_k = r_1 \ldots r_k$ applies $k \in [1,5]$ operators drawn from eight invented maps on $\mathbb{Z}_{11}$ (`zib`: $x{+}1$, `zob`: $x{+}3$, `qua`: $2x{+}1$, `lar`: $3x$, `neg`: $-x$, `flip`: $10{-}x$, `dro`: $5x$, `mik`: x^2, all modulo 11) to v_0 and emits the execution trace r_j (the value after operator j). The target operator is $\Omega =$ `qua`; a ninth operator `vex` ($4x{+}3$) exists only in the vocabulary and in the

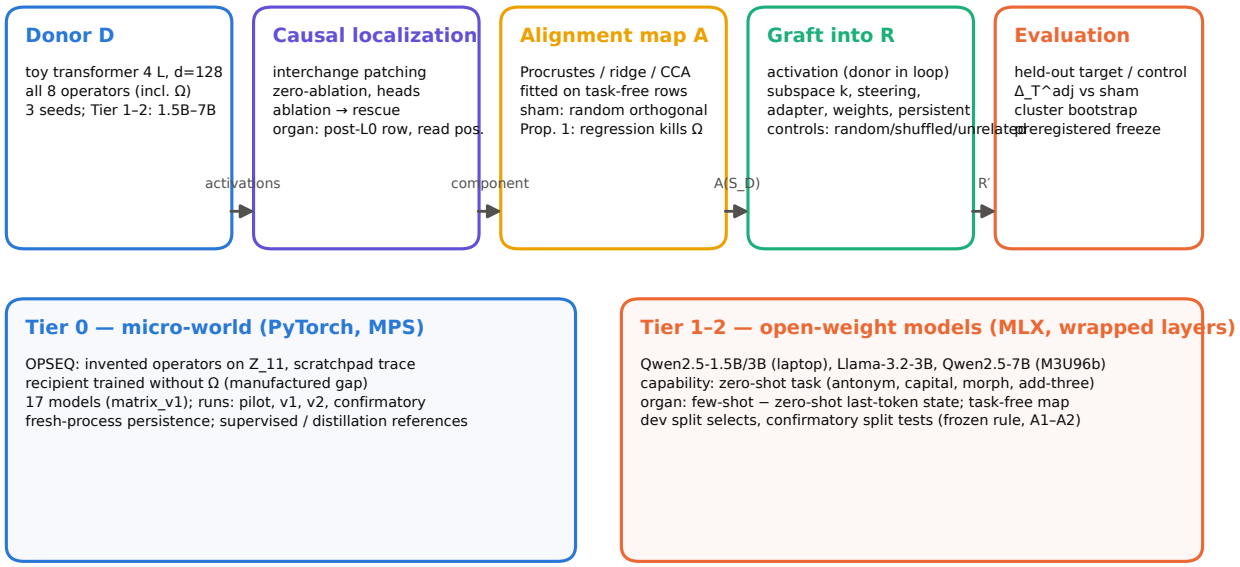


Figure 1. System map. Top: the causal chain applied to every donor–recipient pair (localization before transfer; alignment map fitted on task-free data; graft families and their controls; preregistered evaluation). Bottom: the two tiers, their models, backends and hosts; the registry counts are read from the run manifests.

training distribution of the *unrelated-skill donor*. Two adjacent operator pairs are held out from every training distribution (compositional stratum); programs of length 6–7 form a length-OOD stratum. Every split is generated from its own seed, validated by exact recomputation, hashed, and checked for item-level disjointness against all training, alignment and evaluation splits involved in a run; split hashes are stored in run manifests. Design decision D2 (trace instead of a single answer token) was taken after a single-answer variant memorized without generalizing; its consequence—that the Tier-0 capability is a per-step map once the trace is visible—is controlled by the `trace_stride` knob (stride 2 forces two operators per emitted result).

Models and roles. Decoder-only transformers (4 layers, d=128, 4 heads, 0.80 M parameters; pre-LN, explicit attention with per-head outputs exposed) are trained from scratch with teacher forcing on trace tokens (3 000 steps, batch 256, AdamW, cosine schedule; 40 000 training items). Roles: *donor* (all eight operators), *recipient* (Ω excluded), *unrelated-skill donor* (Ω excluded, `vex` included), three seeds each; a width-mismatched recipient (d=96), a depth-mismatched recipient (6 layers), and a stride-2 donor/recipient pair. The initialization seed equals the data seed, so donor$_i$ and recipient$_i$ share their initial weights ("same initialization") while off-diagonal pairs do not (D8). All models reach dev accuracy 1.000 (Table 9).

Hook semantics. Hook points expose the embedding, per-layer residual streams (pre, mid, post), per-head attention outputs before the output projection and MLP outputs. With no registered intervention, or when disabled, a hook returns the input tensor object itself; bitwise equality of logits is asserted in the test suite and recorded in every run manifest.

Localization. Interchange interventions on 490 clean/corrupted pairs (length-3 programs with Ω at index 1; the corrupted twin replaces Ω by `zib`; identical teacher-forced prefix up to r_1) patch the clean activation into the corrupted run at every (layer, position) for the residual, attention and MLP families and per head; zero-ablation measures necessity; the sham baseline is the corrupted run scored on the clean answer. Sweeps are repeated and compared across donor seeds.

Alignment maps. Donor and recipient residual rows at every read position are collected on 3 000 *alignment* items that contain Ω ("task rows") and on 3 000 items that do not ("task-free rows"), both disjoint from evaluation items. Maps: identity (same width only), orthogonal Procrustes with isotropic scale, ridge regression ($\lambda = 10^{-2}$ relative), CCA regression, random orthogonal (sham), each fitted on task rows or on task-free rows. We report the number of parameters, in-sample R^2 and R^2 on the other row type.

Transplant families and controls. Grafts act at the recipient's post-layer-0 residual stream (`resid_post.0`, the location identified in §5) at every read position during greedy decoding ("untargeted"); an oracle variant restricted to Ω steps is also reported. Families: activation graft (donor in the loop, same token ids), subspace graft ($k \in \{1, \ldots, 128\}$ principal directions of the mapped Ω-contrast), steering graft (mapped mean-difference direction, $\alpha \in \{0.5, 1, 2, 4\}$), adapter graft (donor-free affine interface fitted by ridge to predict mapped donor rows from recipient rows), weight grafts (layer-0 attention, MLP, full block, Ω embedding, layers 0–2 plus embedding), and a persistent graft (the adapter saved to disk and evaluated in a separate process on new items). Controls: unmodified recipient and donor; sham map; norm-matched random rows; shuffled-item donor rows; donor layer-3 rows (shuffled layer); unrelated-skill donor; random subspace of equal k; shuffled-layer and unrelated-donor weight grafts. Conventional-learning references on the same 3 000 alignment items: a supervised affine adapter trained with labels (same parameter count as the adapter graft) and full-model distillation from the donor's output distributions (300 steps each). Few-shot prompting and retrieval are not applicable to the toy models and are marked unsupported.

Sweeps and mechanism. Donor-layer $\times$ recipient-layer grid; dose $\alpha \in [0, 2]$; width and depth mismatches; stride-2; donor ablation (layer-0 attention output zeroed at read positions) followed by rescue from an intact donor of another seed, with a sham rescue; reverse graft (recipient $\rightarrow$ donor);

rule-versus-cache test (maps and adapters fitted only on Ω inputs in $\{0,\ldots,5\}$, evaluated separately on items whose Ω inputs are all seen or all unseen).

4.2. Tier 1: open-weight models

Capability. Performing an input→output task *without demonstrations*. Four tasks with exact-match validators: antonyms (semantic), country→capital (factual), an invented suffix rule on pseudo-words (morphology, no prior), and "add three" on two-digit numbers (formal). Prompts use a fixed `Input:`/`Output:` frame; few-shot prompts carry k=8 demonstrations drawn from a demonstration pool disjoint from the queries; development and confirmatory query sets are disjoint halves of each item pool.

Models. Qwen2.5-1.5B-Instruct (28 layers, d=1536), Qwen2.5-3B-Instruct (36 layers, d=2048) and Llama-3.2-3B-Instruct (28 layers, d=3072), bf16, served by MLX with decoder layers wrapped in place (bitwise no-op verified; per-head outputs are not exposed by MLX and head-level interventions are reported as unsupported for this backend). Pairs: same family/different size in both directions, and different family/different tokenizer in both directions.

Candidate organ and localization. The task vector is the mean difference between the last-token residual state under few-shot and under zero-shot prompts (demonstration items only; queries never enter a vector). Within-donor localization adds the vector to the donor's zero-shot run at each of nine layers (sufficiency) and removes its direction from the few-shot run at the best layer (necessity); a random vector of equal norm is the control.

Transfer and controls. Maps are fitted between donor and recipient last-token states on 800 *format-matched, task-free* prompts (same frame, random content; last 100 rows held out for R^2). Conditions at the recipient's own best layer and at the proportional-depth layer: mapped vector added at $\alpha \in \{1,2\}$ (Procrustes, ridge, CCA), norm-rescaled variants, and a donor-in-the-loop replace graft (the recipient's last-token state replaced by the mapped donor few-shot state for the same query). Controls: recipient zero-shot, recipient few-shot (prompting baseline), recipient's own vector (within-model ceiling), norm-matched random vector, unrelated-task vector, sham map, mapped donor *zero-shot* state, and mapped donor few-shot state of a *shuffled* query. Collateral: next-token negative log-likelihood on held-out task-free sentences and accuracy on the other three tasks under the transferred vector. The intervention is a single-position modification of the prompt pass; generation continues from the modified key–value entry (within-sequence persistence); nothing is saved.

Table 2. Pilot gap qualification (run pilot_tier0_opseq_451ef5087182): final-answer exact match by stratum, mean (sd) over three seeds per role; paired gap with cluster-bootstrap 95% interval over programs.

Stratum	chance	donor (sd)	recipient (sd)	paired gap, seed 0 [95% CI]
target_iid	0.105	1.000 (0.000)	0.092 (0.053)	+0.847 [+0.825, +0.870]
control_iid	0.129	1.000 (0.000)	1.000 (0.000)	+0.000 [+0.000, +0.000]
comp_heldout	0.110	1.000 (0.000)	0.450 (0.032)	+0.513 [+0.468, +0.560]
len_ood	0.140	0.086 (0.017)	0.091 (0.016)	-0.002 [-0.034, +0.032]

4.3. Statistics and preregistration

Cluster bootstrap (2 000 draws) over programs (Tier 0) or items (Tier 1); paired contrasts on identical items; control-adjusted effects against the sham map; Holm correction across the pre-declared families in the confirmatory analysis. Exploratory runs (protocol v1 and v2) selected the primary method; the preregistration was then frozen (v1.0) and confirmatory splits generated from new seeds (§5).

5. Results

Provenance. Every number below names its run: the pilot pilot_tier0_opseq_451ef5087182 (Gates B–C and localization), the exploratory transplant runs graft_tier0_f50a28ca1bf7 (protocol v1) and graft_tier0_30e3644aa801 (v2), the confirmatory run graft_tier0_confirmatory_0c2257f6803e (preregistration v1.0, new seeds), and the Tier-1 runs of §5.5. Tables and figures are regenerated from the results files by scripts/build_tables.py and scripts/build_figures.py; the confirmatory decision rules are evaluated by scripts/confirmatory_analysis.py.

5.1. Gate B: a manufactured, reproducible capability gap

Three donor seeds trained on all eight operators and three recipient seeds trained with Ω excluded reach dev accuracy 1.000. On the held-out target stratum (programs containing Ω; n=1 000 items in ≈ 900 programs) final-answer exact match is 1.000 (sd 0.000) for donors and 0.092 (sd 0.053) for recipients against a majority-class chance of 0.105; the token emitted at the first Ω step is correct 1.000 versus 0.062 of the time. Both roles score 1.000 on the control stratum. The paired gap on identical items (seed 0) is $+0.847$ $[+0.825, +0.870]$ (Table 2, Figure 2). Both roles fail the length-OOD stratum (0.086 vs 0.091), a limitation of absolute position embeddings that we treat as a stratum, not a target.

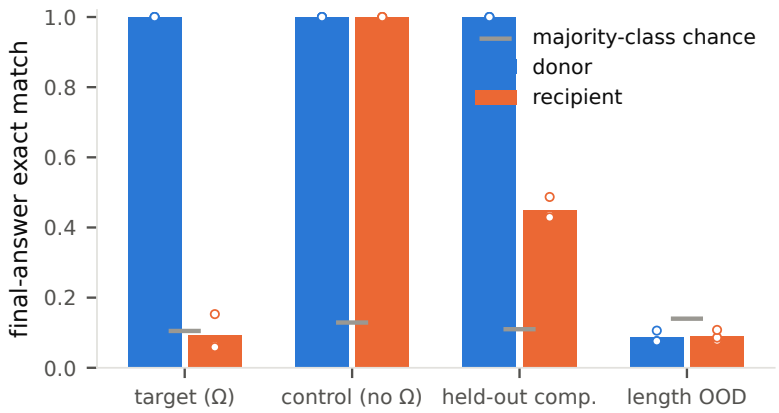


Figure 2. Tier-0 pilot: final-answer exact match by evaluation stratum for donor (all operators) and recipient (Ω excluded), mean over three seeds (bars) with per-seed values (open markers) and the majority-class chance level (grey dashes).

5.2. Gate C: hooks are exact

For all six pilot models, logits computed with (i) no hooks, (ii) capture hooks on all 27 hook points, and (iii) a modifying hook registered but disabled are bitwise identical on 256 items, and repeated runs are bitwise identical on MPS; the same modifying hook, when enabled, changes the output. The MLX backend passes the same test on Qwen2.5-1.5B and Llama-3.2-3B (`tests/integration/test_mlx_noop.py`).

5.3. RQ1: where the donor executes Ω

On 490 clean/corrupted pairs the clean run scores 1.000 and the sham (corrupted run scored on the clean answer) 0.000 in every seed. Patching the clean activation into the corrupted run restores the clean answer at rate 1.00 when the residual stream at the *read position* is patched at any layer, and 0.00 at every other position including the Ω token (Figure 3). Within the read position, the layer-0 attention output is sufficient (1.00; layers 1–3: 0.00) and the layer-0–2 MLP outputs are partially sufficient (1.00, 0.90, 0.83; layer 3: 0.17). Zero-ablation at the read position reduces accuracy to 0.04–0.09 for the residual stream, 0.13 for layer-0 attention and 0.21 for the layer-0 MLP, with no effect elsewhere. Head-level patching attributes most of the attention effect to layer-0 head 0 (0.62; heads 1–2: 0.16, 0.04). Repeating a sweep changes no entry and the heatmaps correlate across seeds at $r \geq 0.96$. The candidate organ is therefore the post-layer-0 residual row at the read position.

5.4. Confirmatory test of the primary hypothesis (preregistration v1.0)

The frozen primary method is the activation graft at the recipient's post-layer-0 residual row at every read position, α=1, through an orthogonal Procrustes map (with isotropic scale) fitted on

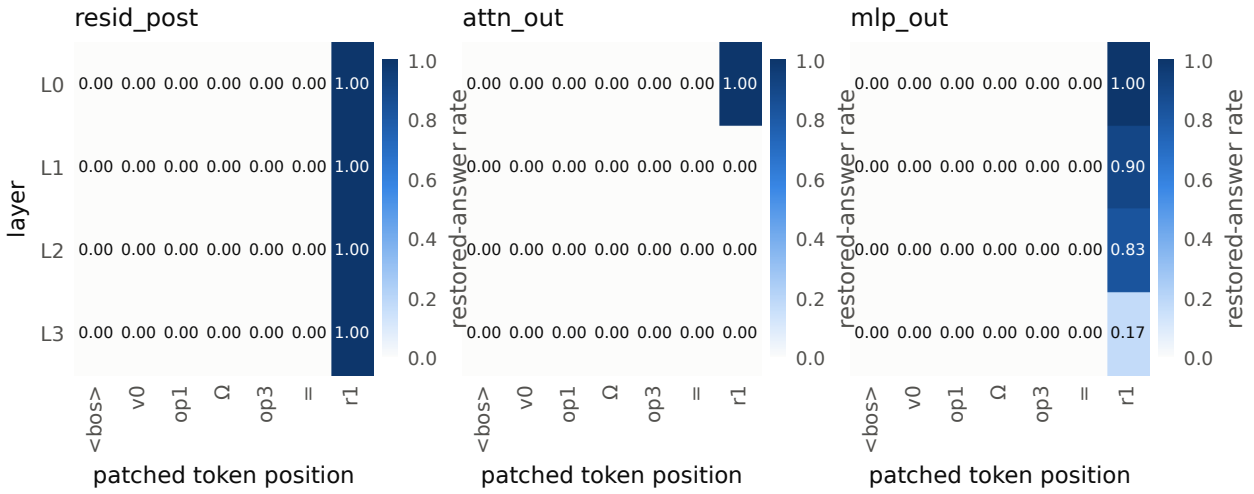


Figure 3. Within-donor interchange interventions (mean of three seeds): restored-answer rate when the clean activation at (layer, position) is patched into the corrupted run, for the post-block residual stream, the attention output and the MLP output. The read position r_1 predicts r_2, the Ω step. Sham rate 0.00.

3 000 task-free alignment items; the primary contrast is the control-adjusted effect Δ_T^{adj} against the sham map, paired on identical items, with cluster-bootstrap intervals over programs. On new evaluation and alignment items (seed base 90 000) the graft reaches target accuracy 0.966 (sd 0.037, min 0.889 over the nine donor–recipient pairs) from recipient baselines of 0.101, 0.056 and 0.057 (chance 0.113), with control-stratum accuracy 0.992 (min 0.946). Δ_T^{adj} is $+0.875$ on average (sd 0.042, min $+0.776$), and its lower 95% bound exceeds the smallest effect of interest (0.25) in **9 of 9** pairs (Table 3). **H2 (sufficiency) is supported.** H4 (retention ≥ 0.95 in every pair) is **not supported**: one different-initialization pair retains 0.946 on the control stratum; the other eight retain ≥ 0.985. The sham map leaves the target stratum at 0.091 and destroys the control stratum (0.100); norm-matched random rows and shuffled-item rows behave the same; shuffled-layer rows (0.186) and the unrelated-skill donor (0.076) leave the control stratum intact.

Secondary pre-declared families (Holm over four). Subspace graft with k=8 versus a random subspace of the same dimension: $+0.850$ $[+0.837, +0.862]$ (0.927 vs 0.077; control 1.000). Donor-free adapter graft versus the unmodified recipient: $+0.472$ $[+0.453, +0.490]$ (0.544; control 0.998); the saved adapter evaluated in a fresh process on new items reaches 0.344 $[0.317, 0.372]$ from 0.115 (control 0.979). Weight graft of the donor's layer-0 block versus a shuffled-layer weight control: $+0.556$ $[+0.537, +0.577]$ (0.642 vs 0.086) at a retention cost (control 0.720, min 0.310). All three are rejected at their Holm thresholds (bootstrap $p \leq 0.0003$). Rule versus cache (H5): the primary map fitted only on Ω rows with operands in $\{0, \ldots, 5\}$ gives Ω-step accuracy 0.941 $[0.919, 0.960]$ on items whose operands are all seen and 0.768 $[0.727, 0.808]$ on items whose operands are all unseen (ratio $0.816 \geq 0.8$): **H5 is supported**; the donor-free adapter gives 0.702 seen and 0.000

Table 3. Confirmatory run graft_tier0_confirmatory_0c2257f6803e: target- and control-stratum accuracy by condition, mean (sd) over same-initialization and different-initialization pairs, and the control-adjusted effect against the sham map (mean of per-pair cluster-bootstrap estimates and bounds). Families other than the activation graft are evaluated on the three matched-seed pairs.

	same initialization ($n = 3$ pairs)		different initialization ($n = 6$)		control-adjusted
Condition	target (sd)	control	target (sd)	control	Δ_T^{adj} [95% CI]
Recipient, unmodified	0.071 (0.02)	1.000			
Activation graft, identity map	0.868 (0.10)	0.989	0.095 (0.01)	0.102	+0.262 [+0.235, +0.287]
Activation graft, Procrustes (task rows)	0.790 (0.10)	0.983	0.697 (0.16)	0.936	+0.636 [+0.605, +0.667]
Activation graft, Procrustes (task-free rows)	0.995 (0.01)	0.999	0.952 (0.04)	0.988	+0.875 [+0.854, +0.895]
Activation graft, ridge (task-free rows)	0.802 (0.10)	1.000	0.788 (0.13)	1.000	+0.702 [+0.672, +0.731]
Activation graft, ridge (task rows)	0.173 (0.06)	0.994	0.175 (0.08)	0.988	+0.083 [+0.054, +0.111]
Activation graft, CCA (task rows)	0.076 (0.03)	0.999	0.083 (0.04)	1.000	-0.010 [-0.035, +0.014]
Control: sham map (random orthogonal)	0.089 (0.00)	0.100	0.093 (0.01)	0.100	+0.000 [+0.000, +0.000]
Control: norm-matched random rows	0.091 (0.01)	0.093	0.091 (0.01)	0.093	-0.000 [-0.026, +0.025]
Control: shuffled-item donor rows	0.106 (0.00)	0.098	0.101 (0.00)	0.097	+0.011 [-0.014, +0.037]
Control: donor layer 3 rows (ridge)	0.186 (0.08)	1.000	0.187 (0.09)	1.000	+0.095 [+0.066, +0.124]
Control: unrelated-skill donor (ridge)	0.076 (0.03)	1.000	0.076 (0.03)	1.000	-0.015 [-0.039, +0.009]
Steering vector, $\alpha = 1$	0.066 (0.02)	1.000	—	—	-0.023 [-0.046, +0.001]
Steering vector, $\alpha = 4$	0.090 (0.01)	0.598	—	—	+0.002 [-0.024, +0.027]
Subspace graft, $k = 8$	0.927 (0.09)	1.000	—	—	+0.839 [+0.815, +0.861]
Subspace graft, $k = 32$	0.983 (0.02)	1.000	—	—	+0.894 [+0.875, +0.913]
Subspace graft, $k = 128$	0.995 (0.01)	0.999	—	—	+0.906 [+0.887, +0.924]
Control: random subspace, $k = 32$	0.168 (0.05)	1.000	—	—	+0.079 [+0.051, +0.107]
Adapter graft (donor-free, ridge interface)	0.544 (0.13)	0.998	—	—	+0.455 [+0.422, +0.490]
Baseline: supervised adapter (labels)	1.000 (0.00)	1.000	—	—	+0.911 [+0.892, +0.928]
Baseline: distillation (donor outputs)	1.000 (0.00)	1.000	—	—	+0.911 [+0.893, +0.929]
Weight graft: L0 MLP	0.245 (0.13)	0.959	—	—	+0.156 [+0.127, +0.187]
Weight graft: L0 attention	0.087 (0.02)	0.732	—	—	-0.002 [-0.027, +0.022]
Weight graft: L0 block	0.642 (0.19)	0.720	—	—	+0.553 [+0.520, +0.586]
Weight graft: L0–L2 blocks + Ω embedding	0.778 (0.26)	0.751	—	—	+0.689 [+0.662, +0.715]
Weight graft: Ω embedding only	0.073 (0.01)	1.000	—	—	-0.016 [-0.039, +0.008]
Control: donor L3 block into L0	0.086 (0.01)	0.086	—	—	-0.003 [-0.027, +0.022]
Control: unrelated-skill donor L0 block	0.097 (0.04)	0.931	—	—	+0.008 [-0.018, +0.033]

unseen, the predicted failure.

Mechanism and portability (fixed tests, confirmatory items). Ablating the donor's layer-0 attention output at read positions gives 0.113; rescue from an intact donor of another seed through the task-free Procrustes map gives 1.000, through a sham map 0.106; the reverse graft leaves the donor at 0.972. Width mismatch (128 → 96): 0.901 (control 1.000) at layer 0 → 0 and 1.000 at 1 → 1; depth mismatch (4 → 6): 0.783 (0.956) at 0 → 0, 0.999 at 1 → 1, 1.000 at 1 → 2 and 2 → 2; stride-2: 0.386, 0.760, 1.000, 1.000 at layers 0–3. Sham maps: 0.08–0.13 with the control stratum destroyed (Figure 5).

Conventional learning on the same items. A supervised affine adapter (same parameter count as the adapter graft, 300 gradient steps on the 3 000 alignment items with labels) and distillation from the donor's output distributions both reach 1.000 on both strata. Transplantation is not a cheaper way to learn this task; it is a way to move the computation without labels or gradient steps.

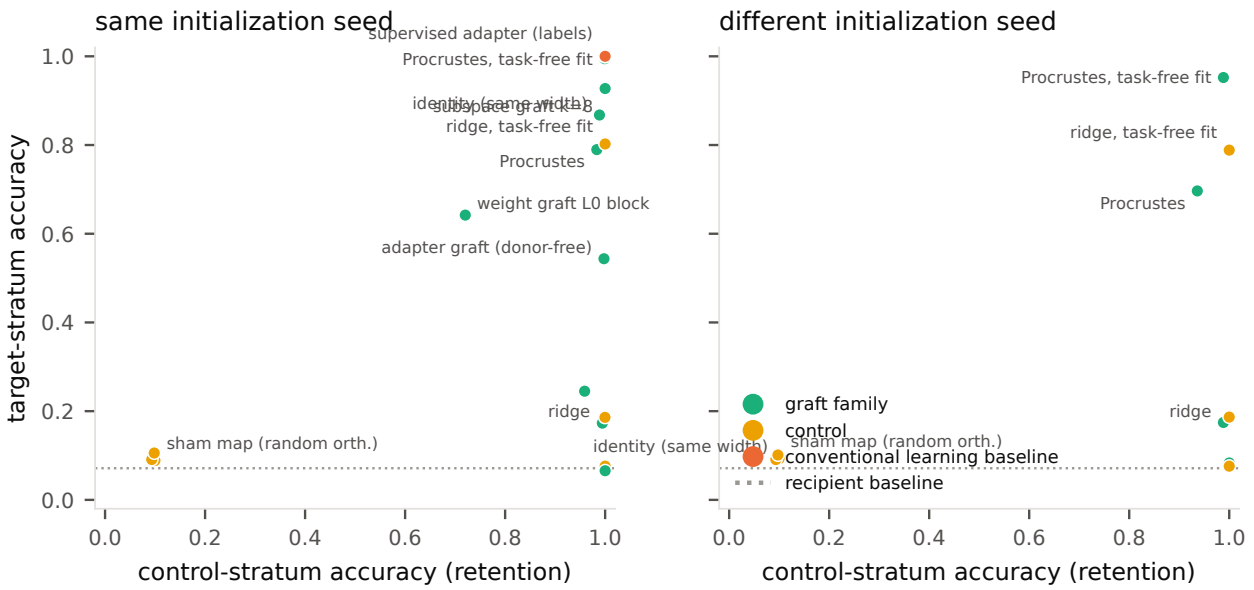


Figure 4. Target-stratum accuracy versus control-stratum retention for every condition of the confirmatory run, mean over same-initialization (left) and different-initialization (right) pairs. Green: graft families; orange: controls; red: conventional-learning references; dotted line: recipient baseline.

5.5. Tier 1: open-weight models

Capability gap and within-model localization (development split). Run `tier1_taskvector_e105e3d91386`. With k=8 demonstrations the three models solve antonyms (0.88–1.00), capitals (1.00) and "add three" (0.86–1.00) and fail the invented morphology (0.33–0.40); zero-shot accuracy is 0.16–0.25 for antonyms and 0.00 for the other tasks (Table 4). Adding the task vector (few-shot minus zero-shot last-token state, demonstration items only) to the zero-shot run restores antonyms to 0.84–0.88 (Llama-3.2-3B layer 14, Qwen2.5-1.5B layer 16, Qwen2.5-3B layers 25–29) and capitals to 0.82–1.00 (layers 14–25); a random vector of equal norm gives 0.00–0.06 except for antonyms in Qwen2.5-1.5B (0.47, where any perturbation of the right norm elicits single-word answers). "Add three" is vectorizable only in Llama-3.2-3B (layer 14, 0.80) and nowhere in the Qwen models; the morphology rule is vectorizable nowhere. Removing the vector's direction from the few-shot state does not reduce few-shot accuracy (0.91–1.00): the direction is sufficient but not necessary, i.e. the few-shot context encodes the task redundantly. Localization is therefore task- and model-dependent, and the candidate organ exists only where the within-model test passes.

Alignment maps. Fitted on 700 format-matched task-free prompts, Procrustes maps reach held-out R^2 0.29–0.59 and ridge maps 0.18–0.81 between last-token states of different models (in-sample 0.65–0.82 and 0.93–1.00); random orthogonal maps are negative. Cross-model alignment of these states is far weaker than in the micro-world.

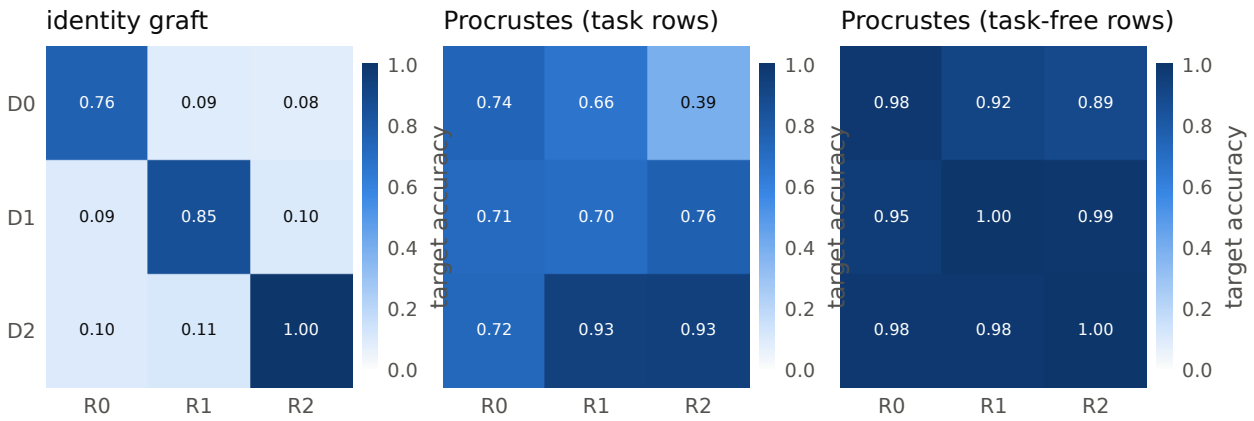


Figure 5. Donor × recipient target accuracy for the identity graft, the Procrustes map fitted on Ω rows and the Procrustes map fitted on task-free rows (confirmatory items). Diagonal cells share the initialization seed.

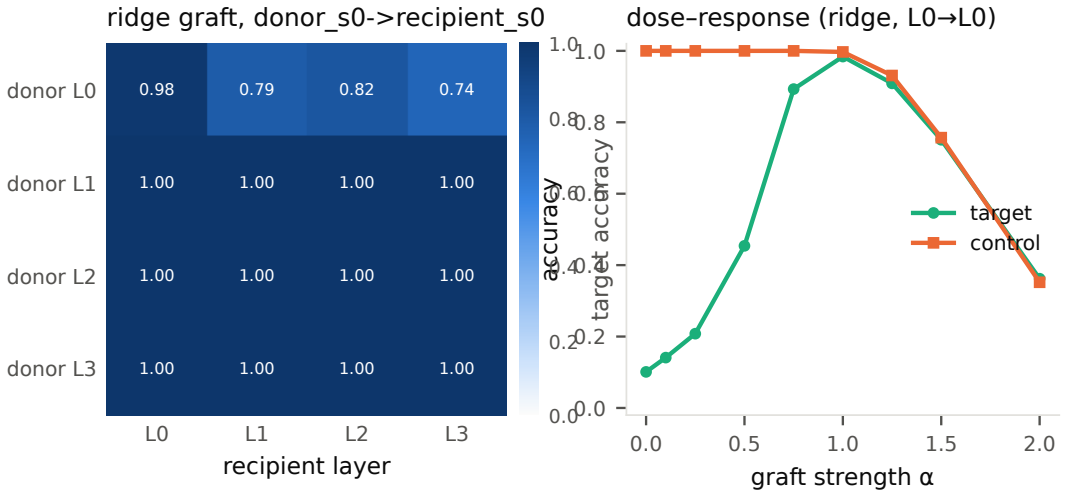


Figure 6. Left: donor-layer × recipient-layer grid of target accuracy for the task-free Procrustes graft (matched seed-0 pair). Right: dose–response of target and control accuracy with graft strength α at layer 0 $\rightarrow$ 0.

Cross-model transfer (development split; selection data). Antonyms transfer in all four pairs: the best development condition reaches 0.72 (Qwen-3B $\rightarrow$ Qwen-1.5B, donor few-shot state replaced through Procrustes at layer 16), 0.75 (Qwen-1.5B $\rightarrow$ Qwen-3B, norm-rescaled Procrustes vector at layer 21), 0.81 (Llama-3B $\rightarrow$ Qwen-3B, CCA vector at layer 25) and 0.66 (Qwen-3B $\rightarrow$ Llama-3B, Procrustes vector at layer 14), against best null controls of 0.44, 0.56, 0.50 and 0.38 and own-vector ceilings of 0.84–0.88. The controls are not at chance for antonyms: an unrelated-task vector or a sham-mapped vector raises antonym accuracy from $\approx$ 0.2 to 0.4–0.6, so part of the antonym effect is a non-specific "answer with one word" push. Capitals transfer in two pairs (0.73 Qwen-1.5B $\rightarrow$ Qwen-3B, Procrustes at layer 28; 0.50 Llama-3B $\rightarrow$ Qwen-3B, CCA at layer 18; controls 0.14 and 0.27) and not in the other two (0.18 and 0.05). The donor few-shot state of a

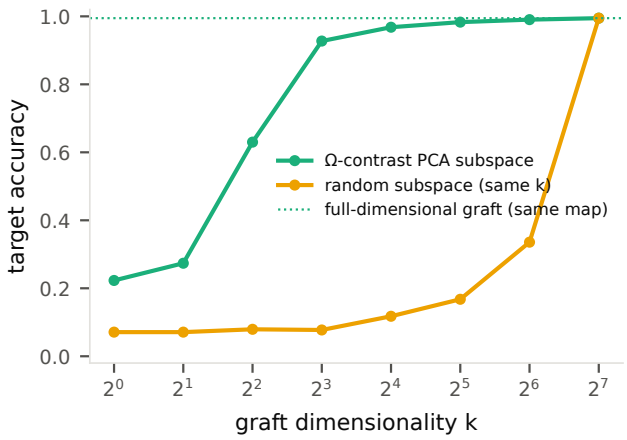


Figure 7. Target accuracy of the subspace graft as a function of its dimensionality k (principal directions of the mapped Ω-contrast) against random subspaces of the same k; mean over matched-seed pairs.

shuffled query transfers antonyms as well as the matched state (0.72 vs 0.72; 0.81 vs 0.81): what moves is the task identity, not query content (amendment A1). Transferred vectors leave the held-out next-token NLL within ± 0.1 nats for antonyms and raise it by 0.4–0.6 nats for the capital vectors in the Qwen pairs; they do not change accuracy on the other tasks (all 0.00).

Tier 2 (Qwen2.5-7B, development split). Run `tier1_taskvector_eb5d091bf5ed` on a 96 GB node. Qwen2.5-7B-Instruct (28 layers, d=3584) localizes antonyms at layer 16 (0.91) and capitals at layer 11 (1.00); the morphology rule and "add three" are not vectorizable in it. Within the Qwen family, antonyms transfer in both directions at 0.84 (7B $\rightarrow$ 3B: donor few-shot state through ridge at layer 25; 3B $\rightarrow$ 7B: through Procrustes at layer 19) against null controls of 0.53 and 0.59; capitals transfer upward (0.91, 3B $\rightarrow$ 7B, ridge vector at layer 11; but the unrelated-task control already gives 0.77 because the 7B model's zero-shot capital accuracy, 0.23, responds to any push) and not downward (0.14). Across families (7B $\rightarrow$ Llama-3B) nothing transfers: antonyms 0.38 against a control of 0.34, capitals 0.05 (Figure 9).

Confirmatory test (frozen rule, disjoint queries). The rule (preregistration, D11 with amendments A1–A2) fixes, per pair and task eligible by within-recipient sufficiency ≥ 0.5, the development-selected condition (map family, recipient layer, α, add or replace) and tests it on the disjoint confirmatory queries against the *best* null control (norm-matched random, unrelated-task vector, sham map, mapped donor zero-shot state), paired on items with an exact item bootstrap; H6 is supported for a pair if the lower bound exceeds 0.10 for at least two tasks. Run `tier1_taskvector_confirmatory_625c54bd186f` (Table 5): antonyms transfer above the best control in all four pairs (+0.16, +0.28, +0.50, +0.25; lower bounds 0.00, 0.09, 0.31, 0.09), capitals in two (+0.36 $[+0.14, +0.59]$ Qwen-1.5B $\rightarrow$ Qwen-3B; +0.18 $[+0.05, +0.36]$ Llama-3B $\rightarrow$ Qwen-3B) and not

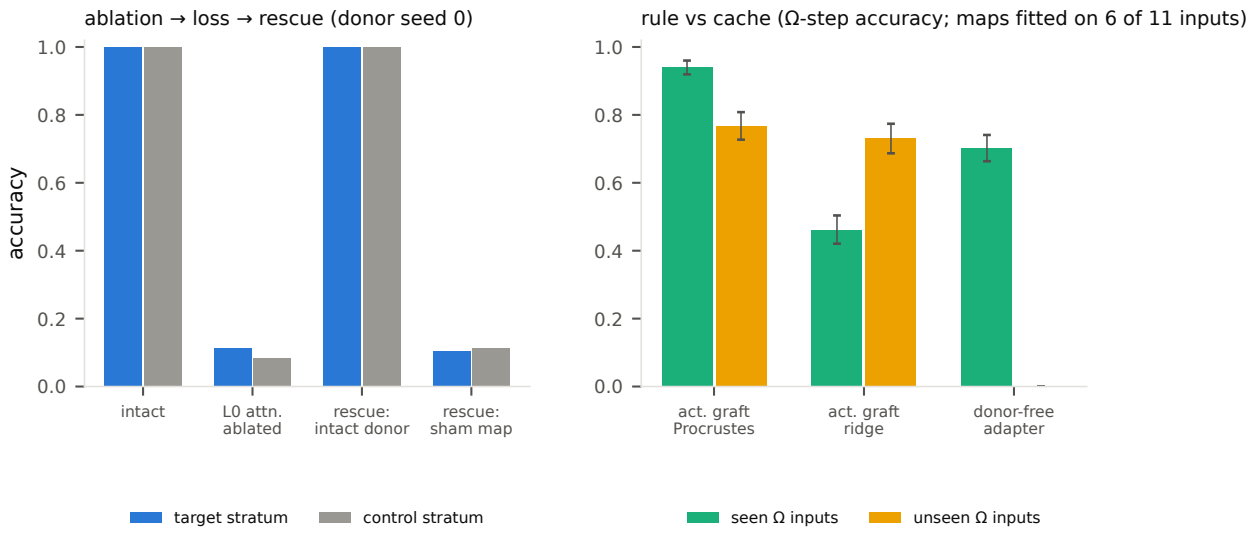


Figure 8. Left: donor ablation → loss → rescue chain (seed 0; rescue from intact donor seed 1 through the task-free Procrustes map versus a sham map). Right: rule-versus-cache test; maps and adapters fitted on Ω rows with operands in $\{0,\ldots,5\}$, evaluated on items whose operands are all seen or all unseen (cluster-bootstrap 95% intervals).

in the other two ($+0.09$, -0.05); "add three" does not transfer into Llama-3B although it is vectorizable there (0.69 own vector, 0.00 transferred). Under the frozen rule **H6 is partially supported** for Qwen-1.5B → Qwen-3B and Llama-3B → Qwen-3B (one task each with lower bound above 0.10) and **not supported** for Qwen-3B → Qwen-1.5B and Qwen-3B → Llama-3B. Transferred vectors never reach the recipient's own vector (0.82–0.91) and the best transfers (antonyms into Qwen-3B from Llama-3B, 0.81) equal the shuffled-query diagnostic, confirming that what moves is the task identity. The null controls are far from chance for antonyms (0.31–0.44), which is why the micro-world's unambiguous separation does not carry over: in these models, part of "doing the task zero-shot" is a non-specific disposition that any vector of the right norm elicits.

Tier-2 confirmatory test. Run `tier1_taskvector_confirmatory_532eb68bd5e2`. With Qwen2.5-7B the pattern is the same (Table 6): antonyms transfer downward into Qwen-3B at 0.84 against a best null control of 0.31 ($+0.53$ $[+0.34, +0.72]$; the recipient's own vector gives 0.91) and upward into 7B at 0.75 against 0.56 ($+0.19$ $[+0.03, +0.34]$); capitals transfer upward at 0.86 but the mapped donor *zero-shot* state already gives 0.73 ($+0.14$ $[+0.00, +0.27]$) and downward only at 0.32 against 0.23; across families (7B → Llama-3B) nothing transfers (antonyms 0.34 vs 0.34; capitals 0.00 vs 0.14; "add three" 0.00 against an own-vector ceiling of 0.71). Under the frozen rule H6 is **partially supported** for 7B → 3B and **not supported** for 3B → 7B and 7B → Llama-3B. Scale does not rescue cross-family transfer in this design.

Table 4. Tier-1 within-model localization (development split): zero-shot and few-shot accuracy, best layer, accuracy with the task vector added to the zero-shot run, with a random vector of equal norm, and few-shot accuracy with the vector's direction removed.

Model	task	zero-shot	few-shot	best layer	vector added (zero-shot)	random vector	few-shot minus direction
llama3b (28 layers, $d = 3072$)	antonym	0.16	0.88	L14	0.84	0.06	0.91
llama3b (28 layers, $d = 3072$)	capital	0.00	1.00	L14	1.00	0.00	1.00
llama3b (28 layers, $d = 3072$)	morph	0.00	0.35	L2	0.00	0.00	0.35
llama3b (28 layers, $d = 3072$)	addthree	0.00	1.00	L14	0.80	0.00	1.00
qwen1.5b (28 layers, $d = 1536$)	antonym	0.22	1.00	L16	0.88	0.47	1.00
qwen1.5b (28 layers, $d = 1536$)	capital	0.00	1.00	L22	0.86	0.00	1.00
qwen1.5b (28 layers, $d = 1536$)	morph	0.00	0.33	L2	0.00	0.00	0.33
qwen1.5b (28 layers, $d = 1536$)	addthree	0.00	0.86	L2	0.00	0.00	0.86
qwen3b (36 layers, $d = 2048$)	antonym	0.25	0.94	L25	0.88	0.06	0.97
qwen3b (36 layers, $d = 2048$)	capital	0.00	1.00	L25	0.82	0.00	1.00
qwen3b (36 layers, $d = 2048$)	morph	0.00	0.40	L2	0.00	0.00	0.40
qwen3b (36 layers, $d = 2048$)	addthree	0.00	1.00	L2	0.00	0.00	1.00

Model	task	zero-shot	few-shot	best layer	vector added (zero-shot)	random vector	few-shot minus direction
llama3b (28 layers, $d = 3072$)	antonym	0.19	0.88	L14	0.84	0.06	0.91
llama3b (28 layers, $d = 3072$)	capital	0.00	1.00	L14	1.00	0.00	1.00
llama3b (28 layers, $d = 3072$)	morph	0.00	0.38	L2	0.00	0.00	0.33
llama3b (28 layers, $d = 3072$)	addthree	0.00	1.00	L14	0.77	0.00	1.00
qwen3b (36 layers, $d = 2048$)	antonym	0.28	0.94	L25	0.88	0.06	0.97
qwen3b (36 layers, $d = 2048$)	capital	0.00	1.00	L25	0.82	0.00	1.00
qwen3b (36 layers, $d = 2048$)	morph	0.00	0.40	L2	0.00	0.00	0.40
qwen3b (36 layers, $d = 2048$)	addthree	0.00	1.00	L2	0.00	0.00	1.00
qwen7b (28 layers, $d = 3584$)	antonym	0.19	0.97	L16	0.91	0.16	0.97
qwen7b (28 layers, $d = 3584$)	capital	0.23	1.00	L11	1.00	0.00	1.00
qwen7b (28 layers, $d = 3584$)	morph	0.00	0.53	L14	0.03	0.00	0.53
qwen7b (28 layers, $d = 3584$)	addthree	0.00	1.00	L2	0.00	0.00	1.00

6. Mechanistic Analysis

Where the donor computes Ω. The pilot localization (§5) shows that patching the residual row at the read position restores the answer at every layer, that layer-0 attention (mostly head 0) moves the operator identity from the Ω token to the read position, and that layer-0 to layer-2 MLPs compute the value map. The candidate organ is therefore the post-layer-0 residual row at the read position, whose content is the (operator, operand) pair.

Necessity, loss and rescue. Zeroing the donor's layer-0 attention output at read positions reduces donor accuracy to 0.086 on the target stratum and 0.102 on the control stratum (all operators pass through the same component). Grafting the post-layer-0 row of an *intact donor of another seed* through the task-free Procrustes map into the ablated donor restores 1.000 on both strata; the same graft through a sham map restores nothing (0.115 / 0.101; Figure 8, left). The reverse graft, recipient $\rightarrow$ donor, leaves the donor's control stratum intact (1.00) and the target stratum at 0.965: the donor's deeper MLPs can recover Ω from the recipient's row because the row still carries the operator identity and the operand; what the recipient lacks is not the information but the map.

Rule, not cache. A Procrustes map fitted only on Ω rows whose operand is in $\{0, \ldots, 5\}$ yields Ω-step accuracy 0.966 on items whose operands are all seen and 0.831 on items whose operands are all unseen (ratio 0.86; confirmatory values in §5). The donor-free adapter fitted on the same

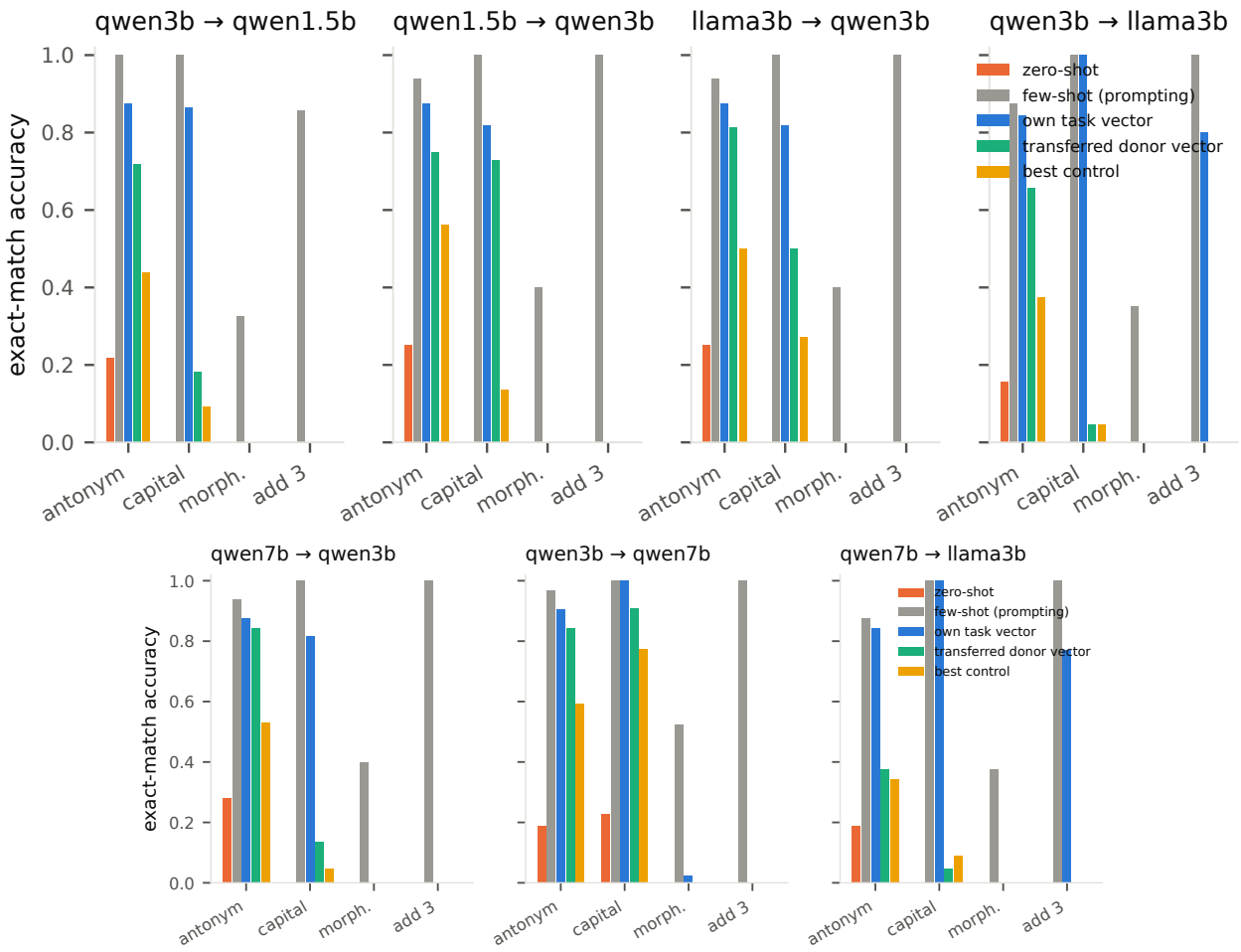


Figure 9. Open-weight models, development split: per pair and task, recipient zero-shot, few-shot (prompting ceiling), recipient's own task vector (within-model ceiling), best transferred donor condition and best null control. Top: Tier-1 pairs (laptop); bottom: Tier-2 pairs with Qwen2.5-7B (M3U96b).

rows gives 0.765 seen and 0.000 unseen (Figure 8, right): the activation graft moves the donor's computation, the adapter memorizes the alignment rows.

Specificity. The unrelated-skill donor (competent on `vex`, ignorant of Ω), the shuffled-item rows, the shuffled-layer rows and the norm-matched random rows all leave the target stratum at chance; the latter three also destroy the control stratum, whereas the unrelated-skill donor and the shuffled-layer rows do not (Table 7). Transfer therefore requires (i) a donor that has the capability, (ii) rows computed on the same item, (iii) the right layer, and (iv) a map that preserves the capability-specific directions.

What is transferred. Eight principal directions of the mapped Ω-contrast suffice (§7); the directions are those along which donor rows on Ω steps differ from the recipient's rows on the same

Table 5. Tier-1 confirmatory test under the frozen rule (run `tier1_taskvector_confirmatory_625c54bd186f`; development selection from `tier1_taskvector_e105e3d91386`). "own" = recipient's own task vector at its development-selected layer; "selected" = development-selected transfer condition; best null control = maximum over norm-matched random, unrelated-task, sham-map and donor zero-shot conditions; shuffled-query diagnostic = mapped donor few-shot state of a different query. Exact paired item bootstrap, 2 000 draws.

Pair	task	eligible	selected condition (dev)	zero	few	own	selected	best null control	selected − control [95% CI]	shuffled-query diag.
qwen3b → qwen1.5b	antonym	yes	replace_donor_fewshot procrustes L16	0.19	0.94	0.91	0.59	0.44	+0.16 [+0.00, +0.34]	0.69
qwen3b → qwen1.5b	capital	yes	ridge L22 a1.0	0.00	1.00	0.82	0.09	0.00	+0.09 [+0.00, +0.23]	0.00
qwen3b → qwen1.5b	morph	no	procrustes L2 a1.0	0.00	0.30	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen3b → qwen1.5b	addthree	no	procrustes L2 a1.0	0.00	0.86	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen3b → qwen1.5b: H6 not supported (0 task(s))										
qwen1.5b → qwen3b	antonym	yes	procrustes_rescaled L21 a1.0	0.09	0.94	0.91	0.69	0.41	+0.28 [+0.09, +0.47]	0.81
qwen1.5b → qwen3b	capital	yes	procrustes L28 a1.0	0.14	1.00	0.82	0.55	0.18	+0.36 [+0.14, +0.59] ✓	0.27
qwen1.5b → qwen3b	morph	no	procrustes L2 a1.0	0.00	0.40	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen1.5b → qwen3b	addthree	no	procrustes L2 a1.0	0.00	1.00	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen1.5b → qwen3b: H6 partial (1 task(s))										
llama3b → qwen3b	antonym	yes	cca L25 a1.0	0.09	0.94	0.91	0.81	0.31	+0.50 [+0.31, +0.69] ✓	0.81
llama3b → qwen3b	capital	yes	cca L18 a1.0	0.14	1.00	0.82	0.64	0.45	+0.18 [+0.05, +0.36]	0.41
llama3b → qwen3b	morph	no	procrustes L2 a1.0	0.00	0.40	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
llama3b → qwen3b	addthree	no	procrustes L2 a1.0	0.00	1.00	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
llama3b → qwen3b: H6 partial (1 task(s))										
qwen3b → llama3b	antonym	yes	procrustes L14 a1.0	0.06	0.91	0.84	0.59	0.34	+0.25 [+0.09, +0.41]	0.38
qwen3b → llama3b	capital	yes	procrustes L14 a1.0	0.09	1.00	1.00	0.09	0.14	-0.05 [-0.18, +0.09]	0.09
qwen3b → llama3b	morph	no	procrustes L2 a1.0	0.00	0.28	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen3b → llama3b	addthree	yes	procrustes L2 a1.0	0.00	1.00	0.69	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen3b → llama3b: H6 not supported (0 task(s))										

steps after the orthogonal map, i.e. the directions the recipient's downstream MLPs read as "apply Ω to this operand". We do not claim that these directions are interpretable features; we claim that they are causally sufficient, low-dimensional and rule-carrying.

7. Robustness and Failure Modes

All numbers in this section are *exploratory* (protocol v2 run `graft_tier0_30e3644aa801`, matched seed-0 pair unless stated); they motivated the frozen protocol and are not re-tested on the confirmatory items except where §5 says so.

Layer and dose. With the task-free Procrustes map, grafting the donor's layer-1, -2 or -3 residual row into *any* recipient layer restores target accuracy 1.00; the donor's layer-0 row restores 0.98 at recipient layer 0 and 0.75–0.82 at deeper recipient layers (Figure 6, left). The dose–response is monotone up to $\alpha = 1$ (target $0.11 \to 0.44 \to 0.90 \to 0.98$ for $\alpha = 0, 0.5, 0.75, 1$), then collapses together with the control stratum (0.39 / 0.36 at $\alpha = 2$; Figure 6, right): the graft is a replacement, not a direction to amplify.

Dimensionality. Restricting the replacement to the top-k principal directions of the mapped Ω-contrast gives $0.23, 0.30, 0.65, 0.94, 0.97, 0.98$ for $k = 1, 2, 4, 8, 16, 32$ (control 1.00), versus 0.08–0.17 for random subspaces of the same k up to 32 (Figure 7): eight of 128 dimensions carry the capability.

Table 6. Tier-2 confirmatory test under the frozen rule (Qwen2.5-7B pairs on M3U96b; run `tier1_taskvector_confirmatory_532eb68bd5e2`, development selection from `tier1_taskvector_eb5d091bf5ed`). Columns as in Table 5.

Pair	task	eligible	selected condition (dev)	zero	few	own	selected	best null control	selected − control [95% CI]	shuffled-query diag.
qwen7b → qwen3b	antonym	yes	replace_donor_fewshot ridge L25	0.09	0.94	0.91	0.84	0.31	+0.53 [+0.34, +0.72] ✓	0.81
qwen7b → qwen3b	capital	yes	ridge L14 a1.0	0.09	1.00	0.77	0.32	0.23	+0.09 [+0.00, +0.23]	0.23
qwen7b → qwen3b	morph	no	procrustes L2 a1.0	0.00	0.40	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen7b → qwen3b	addthree	no	procrustes L2 a1.0	0.00	1.00	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen7b → qwen3b: H6 partial (1 task(s))										
qwen3b → qwen7b	antonym	yes	replace_donor_fewshot procrustes L19	0.25	0.97	0.72	0.75	0.56	+0.19 [+0.03, +0.34]	0.78
qwen3b → qwen7b	capital	yes	ridge L11 a1.0	0.23	1.00	0.91	0.86	0.73	+0.14 [+0.00, +0.27]	0.59
qwen3b → qwen7b	morph	no	procrustes L2 a1.0	0.00	0.42	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen3b → qwen7b	addthree	no	procrustes L2 a1.0	0.00	0.97	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen3b → qwen7b: H6 not supported (0 task(s))										
qwen7b → llama3b	antonym	yes	procrustes L14 a1.0	0.06	0.91	0.84	0.34	0.34	+0.00 [+0.00, +0.00]	0.16
qwen7b → llama3b	capital	yes	procrustes L11 a1.0	0.05	1.00	1.00	0.00	0.14	-0.14 [-0.32, +0.00]	0.05
qwen7b → llama3b	morph	no	procrustes L2 a1.0	0.00	0.28	0.00	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen7b → llama3b	addthree	yes	procrustes L2 a1.0	0.00	1.00	0.71	0.00	0.00	+0.00 [+0.00, +0.00]	0.00
qwen7b → llama3b: H6 not supported (0 task(s))										

Architectural mismatch. Width 128 → 96 and depth 4 → 6: the task-free Procrustes graft reaches 0.886 (control 1.00) and 0.765 (0.957) at layer 0 → 0, and 1.000 (1.00) at layer 1 → 1, 1 → 2 and 2 → 2; sham maps give 0.07–0.11 with the control stratum destroyed (Table 8). The stride-2 pair (two operators per emitted value) transfers at layers 2 and 3 (1.000, control 1.00) and partially at layers 0–1 (0.37, 0.75).

Failure modes. *Regression-fitted maps.* Ridge and CCA maps fitted on Ω rows give 0.17 and 0.08 (Table 7); fitted on Ω rows restricted to six operand values they do *better* on unseen operands (0.74) than on seen ones (0.47), the signature of the annihilation mechanism of Proposition 1. *Identity graft.* 0.87 for same-initialization pairs, 0.10 and control destroyed for different-initialization pairs: representations of differently initialized models are not aligned coordinate-wise (cf. permutation symmetries, Ainsworth et al., 2022). *Steering vectors.* The mapped mean-difference direction transfers nothing at any dose (≤ 0.10) and harms the control stratum at $\alpha = 4$. *Weight grafts.* Copying the donor's layer-0 MLP transfers a little (0.26, control 0.96); copying the full block or layers 0–2 plus the Ω embedding transfers more (0.64–0.77) at a large cost on the control stratum (0.73–0.75); copying only the Ω embedding does nothing; shuffled-layer weights destroy both strata; unrelated-donor weights leave both near baseline. *Donor-free adapter.* 0.58 in-run and 0.41 in a fresh process on new items (base 0.14), but zero on unseen operand values: a cache of the alignment rows, not the rule. *Over-dosing* and *wrong layers* fail gracefully for deeper donor layers and sharply for shallow ones.

8. Discussion

What the results establish. In a controlled micro-world, a capability that a recipient transformer demonstrably lacks can be installed at inference time by replacing one residual row, at the read

Table 7. Exploratory run graft_tier0_30e3644aa801 (protocol v2): target- and control-stratum accuracy by condition, mean (sd) over same-initialization and different-initialization pairs, and the control-adjusted effect against the sham map.

	same initialization ($n = 3$ pairs)		different initialization ($n = 6$)		control-adjusted
Condition	target (sd)	control	target (sd)	control	Δ_T^{adj} [95% CI]
Recipient, unmodified	0.075 (0.03)	1.000			
Activation graft, identity map	0.873 (0.10)	0.990	0.101 (0.01)	0.101	+0.260 [+0.234, +0.287]
Activation graft, Procrustes (task rows)	0.785 (0.08)	0.985	0.697 (0.15)	0.943	+0.628 [+0.596, +0.660]
Activation graft, Procrustes (task-free rows)	0.993 (0.01)	0.996	0.956 (0.04)	0.989	+0.870 [+0.849, +0.891]
Activation graft, ridge (task-free rows)	0.814 (0.08)	1.000	0.795 (0.13)	1.000	+0.703 [+0.675, +0.732]
Activation graft, ridge (task rows)	0.170 (0.08)	0.994	0.175 (0.08)	0.991	+0.075 [+0.046, +0.104]
Activation graft, CCA (task rows)	0.080 (0.03)	1.000	0.089 (0.04)	1.000	-0.012 [-0.038, +0.013]
Control: sham map (random orthogonal)	0.102 (0.01)	0.085	0.096 (0.00)	0.100	+0.000 [+0.000, +0.000]
Control: norm-matched random rows	0.089 (0.00)	0.097	0.089 (0.00)	0.099	-0.009 [-0.034, +0.017]
Control: shuffled-item donor rows	0.102 (0.00)	0.101	0.099 (0.01)	0.101	+0.002 [-0.024, +0.028]
Control: donor layer 3 rows (ridge)	0.187 (0.07)	1.000	0.194 (0.08)	1.000	+0.093 [+0.064, +0.123]
Control: unrelated-skill donor (ridge)	0.076 (0.04)	1.000	0.076 (0.04)	1.000	-0.022 [-0.046, +0.003]
Steering vector, $\alpha = 1$	0.064 (0.02)	1.000	—	—	-0.037 [-0.062, -0.013]
Steering vector, $\alpha = 4$	0.096 (0.00)	0.611	—	—	-0.006 [-0.033, +0.019]
Subspace graft, $k = 8$	0.936 (0.07)	1.000	—	—	+0.834 [+0.811, +0.856]
Subspace graft, $k = 32$	0.983 (0.02)	1.000	—	—	+0.881 [+0.861, +0.901]
Subspace graft, $k = 128$	0.993 (0.01)	0.996	—	—	+0.891 [+0.872, +0.910]
Control: random subspace, $k = 32$	0.173 (0.04)	1.000	—	—	+0.071 [+0.043, +0.099]
Adapter graft (donor-free, ridge interface)	0.581 (0.11)	0.998	—	—	+0.480 [+0.445, +0.515]
Baseline: supervised adapter (labels)	1.000 (0.00)	1.000	—	—	+0.898 [+0.879, +0.916]
Baseline: distillation (donor outputs)	1.000 (0.00)	1.000	—	—	+0.898 [+0.879, +0.916]
Weight graft: L0 MLP	0.262 (0.14)	0.962	—	—	+0.160 [+0.129, +0.192]
Weight graft: L0 attention	0.079 (0.03)	0.733	—	—	-0.023 [-0.048, +0.003]
Weight graft: L0 block	0.642 (0.19)	0.726	—	—	+0.540 [+0.507, +0.574]
Weight graft: L0–L2 blocks + Ω embedding	0.774 (0.26)	0.749	—	—	+0.672 [+0.644, +0.699]
Weight graft: Ω embedding only	0.075 (0.02)	1.000	—	—	-0.027 [-0.051, -0.002]
Control: donor L3 block into L0	0.083 (0.01)	0.089	—	—	-0.018 [-0.044, +0.007]
Control: unrelated-skill donor L0 block	0.094 (0.04)	0.933	—	—	-0.007 [-0.033, +0.019]

position, with the donor's row through an orthogonal map fitted on data that never exercises the capability. The effect is large (control-adjusted $+0.88$ on average, lower bound above the smallest effect of interest in every pair), selective (control stratum intact in eight of nine pairs, 0.946 in the ninth), reproducible across donor and recipient seeds and across different initializations, low-dimensional (eight directions), rule-carrying (unseen operands), causally anchored (ablation, rescue, sham) and portable across width, depth and a harder composition regime. This answers the central question positively *for a bounded, causally localized computation*.

What they do not establish. They do not establish that the same holds for open-ended skills, for large models, or without the donor in the loop. The donor-free variants are weaker and qualitatively different: the adapter caches, the weight grafts damage the control stratum, the steering vector transfers nothing. And conventional learning on the same items is perfect here; the value of transplantation is that no label and no gradient step is needed, which matters when the capability has no labelled data or when the recipient must not be trained, not when it is cheap to train.

Why regression-fitted maps fail, and why it matters for prior work. The annihilation proposition is elementary but consequential: any map fitted to predict the recipient from the donor on inputs where the donor computes something the recipient cannot will remove that computation, in proportion to how unpredictable it is from the recipient. This is a plausible, testable account of the negative result of Zhang and Xin (2026), whose bridge was fitted by regression on hidden states of the very prompts whose reasoning state it was meant to carry; it also suggests that the success of affine stitching for *features both models already represent* (Chen et al., 2025; Kim et al., 2026) and its failure for *capabilities one model lacks* are two faces of the same fitting criterion. The remedy we tested is cheap: fit on task-free rows, or constrain the map to be orthogonal.

Initialization. The identity graft works only between models that share their initial weights. This recovers, in a transfer setting, the permutation-symmetry picture of model merging (Ainsworth et al., 2022): differently initialized models compute in rotated coordinates, and a rotation fitted on unrelated data suffices to undo the difference for this capability.

Alternative explanations. (i) *Answer leakage*: the donor row at the read position is computed from the same token ids and contains the operand and the operator identity but not the answer token; the shuffled-item control shows that rows from other items do nothing, and the reverse graft shows that the recipient's own row already contains the operand and operator. (ii) *Trivial map*: the sham map and the norm-matched random rows destroy both strata, so success is not a property of "any row of the right norm". (iii) *Selection on the test set*: the method, layer and dose were fixed before the confirmatory items were generated; the one preregistered criterion that failed (retention ≥ 0.95 in every pair) is reported as failed. (iv) *Task triviality*: granted for learning (the supervised baselines), not for transfer (every control fails).

Open-weight models. Three facts carry over from the micro-world and three do not. Carried over: a task-conditioned state exists and is causally sufficient within a model for some tasks (antonyms, capitals) at specific layers; its mapped form transfers between models above null controls in several pairs; and what transfers is the task identity, not query content. Not carried over: the separation from controls is modest (the best null controls reach 0.3–0.5 on antonyms because the recipient's zero-shot behaviour responds to any push of the right norm), the effect depends on the pair and direction (upward within the Qwen family and from Llama into Qwen-3B work best; into Llama-3B and from 7B into Llama-3B do not), and tasks that are not vectorizable in the recipient (the invented morphology everywhere, "add three" in Qwen) cannot be installed by this route at all. The frozen rule returns "partial" or "not supported" per pair; we report it as such. The micro-world result therefore does not license claims about capability transfer between large models; it isolates the

conditions under which transfer is identifiable, and shows that in open-weight models the first of those conditions—a bounded, causally sufficient organ for the capability—holds only for some tasks and the second—an alignment map that places it where the recipient reads it—holds only for some pairs.

9. Limitations, Ethics, and Reproducibility

What the micro-world cannot show. Once the execution trace is visible, the Tier-0 capability is a per-step map from (operator, operand) to a value; the stride-2 variant forces two operators per prediction but still a bounded computation. Tier-0 evidence therefore establishes that a *bounded, causally localized computation* can be grafted; it does not establish that multi-step reasoning or open-ended skills can. The toy models are tiny (0.80 M parameters, 4 layers); the location of the organ (layer 0) is a property of these models, not a prediction for large ones. Donor and recipient share the training distribution except for Ω; real model pairs differ in far more.

Donor in the loop. The graft that works is an inference-time dependency on the donor: both models run on every prompt. It is a transplant of *computation*, not a saved modification. The donor-free variants (adapter, weights) are the ones that would matter for deployment and they are the weaker ones.

Conventional learning wins here. A supervised adapter with the same parameter count, or distillation on the same 3 000 items, reaches perfect accuracy. The comparison is fair (same items) but the task is easy to learn; the interest of transplantation is the absence of labels and gradient steps, not efficiency.

Tier 1. The open-weight experiments use instruction-tuned checkpoints whose training data are unknown; "zero-shot" accuracy is a property of our prompt frame; the alignment prompts are synthetic; head-level interventions are unsupported in the MLX backend; three models and four tasks do not support claims about families or scales in general. Items for two tasks (antonyms, capitals) are drawn from finite hand-written lists and may be contaminated in pretraining; the invented tasks are not.

Statistics. Programs are the bootstrap unit and model pairs the second level; with nine pairs (three same-initialization), pair-level inference is descriptive. Exploratory sweeps selected the primary method; the confirmatory run uses new seeds but the same trained models, so model-training variance is not re-sampled at the confirmatory stage.

Ethics. All Tier-0 data are synthetic. Open-weight checkpoints are used under their licenses (Qwen2.5-1.5B: Apache-2.0; Qwen2.5-3B: Qwen Research License; Llama-3.2-3B: Llama 3.2 Community License); no derived weights are released. The method moves a computation between models without training and could in principle be used to move unwanted behaviours; the controls in this paper (sham, unrelated donor) are also the detection tools for such transfers.

Reproducibility. Every run has a manifest with git commit, configuration hash, split hashes, seeds, library versions and hardware; the registry lists every planned, running, failed and completed run; figures and tables are regenerated from results files by `scripts/build_figures.py` and `scripts/build_tables.py`; `docs/REPRODUCIBILITY.md` gives the commands (CPU smoke test ≈ 25 s; Tier-0 pilot ≈ 5 min; Tier-0 transplant run ≈ 3 min on an M5 Max; Tier-1 run minutes to an hour depending on pairs).

10. Conclusion

We set out to test whether a causally identified subsystem can be transplanted between language models without training. In a micro-world built so that the question is identifiable, it can: one orthogonally mapped residual row, fitted on task-free data, moves a rule from a donor to a recipient that never learned it, with the full battery of controls at chance and a preregistered confirmatory test passed on the primary contrast. The same experiments draw the boundary: regression-fitted maps annihilate what the recipient cannot predict, donor-free variants cache rather than compute, and conventional learning is unbeatable when labels are available. On open-weight models the same protocol finds task vectors that are causally sufficient within a model for antonyms and capitals, transfers them between models above null controls in some pairs and directions under a frozen rule, and fails for tasks that no model represents as a vector; the honest summary is partial transfer of task identity, not acquisition of a capability. Everything needed to reproduce or refute these statements—generators, models, hooks, registry, manifests and analysis scripts—is in the repository.

References

Achara, Akshit, Tatiana Gaintseva, Mateo Mahaut, Pritish Chakraborty, Viktor Stenby Johansson, Melih Barsbey, Emanuele Rodolà, and Donato Crisostomi, "Multi-Way Representation Alignment," 2 2026. arXiv:2602.06205; metadata verified via arXiv API on 2026-11-09.

Agarwal, Ayushi, "Cross-Architecture Steering Transfer in Language Models: A Systematic Empirical Study," 5 2026. arXiv:2608.05164; metadata verified via arXiv API on 2026-11-09.

Ainsworth, Samuel K., Jonathan Hayase, and Siddhartha Srinivasa, "Git Re-Basin: Merging Models modulo Permutation Symmetries," 9 2022. arXiv:2209.04836; metadata verified via arXiv API on 2026-11-09.

Bansal, Yamini, Preetum Nakkiran, and Boaz Barak, "Revisiting Model Stitching to Compare Neural Representations," 6 2021. arXiv:2106.07682; metadata verified via arXiv API on 2026-11-09.

Chen, Alan, Jack Merullo, Alessandro Stolfo, and Ellie Pavlick, "Transferring Linear Features Across Language Models With Model Stitching," 6 2025. arXiv:2506.06609; metadata verified via arXiv API on 2026-11-09.

Conmy, Arthur, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso, "Towards Automated Circuit Discovery for Mechanistic Interpretability," 4 2023. arXiv:2304.14997; metadata verified via arXiv API on 2026-11-09.

Cristofano, Tony, "Universal Refusal Circuits Across LLMs: Cross-Model Transfer via Trajectory Replay and Concept-Basis Reconstruction," 1 2026. arXiv:2601.16034; metadata verified via arXiv API on 2026-11-09.

Cunningham, Hoagy, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey, "Sparse Autoencoders Find Highly Interpretable Features in Language Models," 9 2023. arXiv:2309.08600; metadata verified via arXiv API on 2026-11-09.

Fan, Jiahe, Si Chen, Yinghao Hou, Aiyuan Zhang, and Hong Xie, "Training-Free Knowledge Transfer Across Model Scales through Activation-Guided Pruning," 7 2026. arXiv:2608.13596; metadata verified via arXiv API on 2026-11-09.

Geiger, Atticus, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard, "Causal Abstraction: A Theoretical Foundation for Mechanistic Interpretability," 1 2023. arXiv:2301.04709; metadata verified via arXiv API on 2026-11-09.

_ , Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman, "Finding Alignments Between Interpretable Causal Variables and Distributed Neural Representations," 3 2023. arXiv:2303.02536; metadata verified via arXiv API on 2026-11-09.

Gu, Jian, Aldeida Aleti, Chunyang Chen, and Hongyu Zhang, "Beyond Neural Incompatibility: Cross-Scale Knowledge Transfer in Language Models through Latent Semantic Alignment," 10 2025. arXiv:2510.24208; metadata verified via arXiv API on 2026-11-09.

Gurnee, Wes, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas, "Finding Neurons in a Haystack: Case Studies with Sparse Probing," 5 2023. arXiv:2305.01610; metadata verified via arXiv API on 2026-11-09.

Hendel, Roee, Mor Geva, and Amir Globerson, "In-Context Learning Creates Task Vectors," 10 2023. arXiv:2310.15916; metadata verified via arXiv API on 2026-11-09.

Huh, Minyoung, Brian Cheung, Tongzhou Wang, and Phillip Isola, "The Platonic Representation Hypothesis," 5 2024. arXiv:2405.07987; metadata verified via arXiv API on 2026-11-09.

Ilharco, Gabriel, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi, "Editing Models with Task Arithmetic," 12 2022. arXiv:2212.04089; metadata verified via arXiv API on 2026-11-09.

Kim, Su-Hyeon, Jiwan Mun, and Yo-Sub Han, "One Adapter Pair per Model: A Universal Activation Interface for Language Models," 8 2026. arXiv:2608.09521; metadata verified via arXiv API on 2026-11-09.

Kornblith, Simon, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton, "Similarity of Neural Network Representations Revisited," 5 2019. arXiv:1905.00414; metadata verified via arXiv API on 2026-11-09.

Lan, Michael, Philip Torr, Austin Meek, Ashkan Khakzar, David Krueger, and Fazl Barez, "Quantifying Feature Space Universality Across Large Language Models via Sparse Autoencoders," 10 2024. arXiv:2410.06981; metadata verified via arXiv API on 2026-11-09.

Lee, Andrew, Melanie Weber, Fernanda Viégas, and Martin Wattenberg, "Shared Global and Local Geometry of Language Model Embeddings," 3 2025. arXiv:2503.21073; metadata verified via arXiv API on 2026-11-09.

Li, Kenneth, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg, "Inference-Time Intervention: Eliciting Truthful Answers from a Language Model," 6 2023. arXiv:2306.03341; metadata verified via arXiv API on 2026-11-09.

Lo, Michelle, Shay B. Cohen, and Fazl Barez, "Large Language Models Relearn Removed Concepts," 1 2024. arXiv:2401.01814; metadata verified via arXiv API on 2026-11-09.

Meng, Kevin, David Bau, Alex Andonian, and Yonatan Belinkov, "Locating and Editing Factual Associations in GPT," 2 2022. arXiv:2202.05262; metadata verified via arXiv API on 2026-11-09.

Minixhofer, Benjamin, Edoardo Maria Ponti, and Ivan Vulić, "Zero-Shot Tokenizer Transfer," 5 2024. arXiv:2405.07883; metadata verified via arXiv API on 2026-11-09.

Nasiri-Sarvi, Ali, Hassan Rivaz, and Mahdi S. Hosseini, "SPARC: Concept-Aligned Sparse Autoencoders for Cross-Model and Cross-Modal Interpretability," 7 2025. arXiv:2507.06265; metadata verified via arXiv API on 2026-11-09.

Ognev, Daniil, Célian Vasson, Lijie Hu, Kentaro Inui, and Benjamin Heinzerling, "SharedSAE: One Feature Dictionary Across Language Models," 9 2026. arXiv:2609.04344; metadata verified via arXiv API on 2026-11-09.

Panigrahi, Abhishek, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora, "Task-Specific Skill Localization in Fine-tuned Language Models," 2 2023. arXiv:2302.06600; metadata verified via arXiv API on 2026-11-09.

Raghu, Maithra, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein, "SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability," 6 2017. arXiv:1706.05806; metadata verified via arXiv API on 2026-11-09.

Sinii, Viacheslav, Nikita Balagansky, Gleb Gerasimov, Daniil Laptev, Yaroslav Aksenov, Vadim Kurochkin, Alexey Gorbatovski, Boris Shaposhnikov, and Daniil Gavrilov, "Small Vectors, Big Effects: A Mechanistic Study of RL-Induced Reasoning via Steering Vectors," 9 2025. arXiv:2509.06608; metadata verified via arXiv API on 2026-11-09.

Sun, Mingjie, Xinlei Chen, J. Zico Kolter, and Zhuang Liu, "Massive Activations in Large Language Models," 2 2024. arXiv:2402.17762; metadata verified via arXiv API on 2026-11-09.

Thasarathan, Harrish, Julian Forsyth, Thomas Fel, Matthew Kowal, and Konstantinos G. Derpanis, "Universal Sparse Autoencoders: Interpretable Cross-Model Concept Alignment," 2 2025. arXiv:2502.03714; metadata verified via arXiv API on 2026-11-09.

Todd, Eric, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau, "Function Vectors in Large Language Models," 10 2023. arXiv:2310.15213; metadata verified via arXiv API on 2026-11-09.

Turner, Alexander Matt, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid, "Steering Language Models With Activation Engineering," 8 2023. arXiv:2308.10248; metadata verified via arXiv API on 2026-11-09.

Vig, Jesse, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber, "Causal Mediation Analysis for Interpreting Neural NLP: The Case of Gender Bias," 4 2020. arXiv:2004.12265; metadata verified via arXiv API on 2026-11-09.

Wang, Kevin, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt, "Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 small," 11 2022. arXiv:2211.00593; metadata verified via arXiv API on 2026-11-09.

Yadav, Prateek, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal, "TIES-Merging: Resolving Interference When Merging Models," 6 2023. arXiv:2306.01708; metadata verified via arXiv API on 2026-11-09.

Zhang, Peiyan and Jason Xin, "A Negative Result on Cross-Model Activation Transfer in a Pythia Multi-Hop Setting," 6 2026. arXiv:2606.03280; metadata verified via arXiv API on 2026-11-09.

Zou, Andy, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks, "Representation Engineering: A Top-Down Approach to AI Transparency," 10 2023. arXiv:2310.01405; metadata verified via arXiv API on 2026-11-09.

A. Proofs and formal statements

Proof of Proposition 1. Write $X = X_c + X_u$ with $X_u = XP_U$ and $X_c = X(I - P_U)$. The hypotheses $X_u^\top X_c = 0$ and $X_u^\top Y = 0$ give, in a basis adapted to $U \oplus U^\perp$, $X^\top X = \mathrm{diag}(X_u^\top X_u, X_c^\top X_c)$ and $X^\top Y = \big(0;\ X_c^\top Y\big)$. Hence $W_\lambda = (X^\top X + \lambda I)^{-1} X^\top Y = \big(0;\ (X_c^\top X_c + \lambda I)^{-1} X_c^\top Y\big)$, whose rows indexed by U vanish: $P_U W_\lambda = 0$. For a donor activation $x = x_c + x_u$, $xW_\lambda = x_c W_\lambda$, so the U-component is annihilated. For CCA, the canonical correlations between U-coordinates and Y are zero by $X_u^\top Y = 0$, so the top-k canonical directions of X lie in $U^\perp$ (for any k smaller than the number of nonzero canonical correlations), and the regression of Y on those variates again has no U-component. For an orthogonal Q, $\|x_u Q\| = \|x_u\|$ by isometry; with an isotropic scale $s > 0$, $\|s\, x_u Q\| = s\|x_u\|$. □

Remark. In the experiments the hypotheses hold only approximately: the recipient's residual rows on Ω steps do contain the untrained Ω-token embedding and the operand value, so the donor's Ω-specific directions are weakly correlated with recipient coordinates. The proposition then predicts strong shrinkage rather than exact annihilation, which is what the ridge and CCA grafts show (§5).

Interchange commutation under exact causal abstraction (restated). Let $\mathscr{L}$ be a high-level causal model that is an exact abstraction of the donor's computation at a residual-stream variable h under alignment τ_D, and of the recipient's at h' under τ_R (Geiger et al., 2023a). If a bijection A satisfies $\tau_R \circ A = \tau_D$ on the support of h, then for every interchange intervention on h in the donor there is an interchange intervention on h' in the recipient (through A) with the same high-level effect; in particular, grafting $A(h)$ into the recipient reproduces the donor's high-level behaviour on the abstracted variable. The condition $\tau_R \circ A = \tau_D$ is precisely what is *not* guaranteed by representational similarity and must be tested; the experiments in §5 estimate how far from it the fitted maps are.

B. Configurations, splits and model cards

Tables of split hashes, model specifications and run identifiers are generated from the run manifests (`experiments/manifests/`) by `scripts/build_tables.py`; the registry (`experiments/registry/runs.jsonl`) lists every planned, running, failed and completed run.

C. Tier-1 prompts and tasks

Few-shot prompt: k=8 demonstration blocks followed by the query block; zero-shot prompt: the query block only.

```
Input: <x_1>
Output: <y_1>

...

Input: <query>
Output:
```

Alignment prompts: the same frame with 0–3 random demonstration pairs whose outputs are random and a random query (sentences, words, numbers, names), none of which encodes a task. Task item lists are in `src/wp_ai_0001/datasets/tier1_tasks.py` with their hashes in the run manifests.

Table 8. Architectural mismatch and stride-2 transfer (exploratory run): target accuracy, Ω-step accuracy, control accuracy and in-sample R^2 of the map; `unrelated_procrustes` = Procrustes fitted on task-free rows; `sham` = random orthogonal map.

Pair	condition	target	Ω-step	control	map R^2
donor_s2_s0	base	1.000	1.000	1.000	—
recipient_s2_s0	base	0.075	0.039	1.000	—
recipient_L6_s0	base	0.120	0.098	1.000	—
recipient_w96_s0	base	0.096	0.079	1.000	—
donor_s0->recipient_w96_s0	mismatch_procrustes_L0to0	0.638	0.702	0.981	0.518
donor_s0->recipient_L6_s0	mismatch_procrustes_L0to0	0.358	0.471	0.750	0.539
donor_s0->recipient_w96_s0	mismatch_ridge_L0to0	0.176	0.172	1.000	0.879
donor_s0->recipient_L6_s0	mismatch_ridge_L0to0	0.140	0.128	0.986	0.916
donor_s0->recipient_w96_s0	mismatch_cca_L0to0	0.087	0.071	1.000	0.933
donor_s0->recipient_L6_s0	mismatch_cca_L0to0	0.131	0.117	1.000	0.970
donor_s0->recipient_w96_s0	mismatch_unrelated_procrustes_L0to0	0.886	0.908	1.000	0.662
donor_s0->recipient_L6_s0	mismatch_unrelated_procrustes_L0to0	0.765	0.828	0.957	0.685
donor_s0->recipient_w96_s0	mismatch_sham_L0to0	0.109	0.082	0.066	—
donor_s0->recipient_L6_s0	mismatch_sham_L0to0	0.079	0.077	0.075	—
donor_s0->recipient_w96_s0	mismatch_procrustes_L1to1	1.000	1.000	1.000	0.548
donor_s0->recipient_L6_s0	mismatch_procrustes_L1to1	0.988	0.999	1.000	0.545
donor_s0->recipient_w96_s0	mismatch_ridge_L1to1	0.186	0.182	1.000	0.860
donor_s0->recipient_L6_s0	mismatch_ridge_L1to1	0.163	0.164	0.999	0.891
donor_s0->recipient_w96_s0	mismatch_cca_L1to1	0.120	0.105	1.000	0.919
donor_s0->recipient_L6_s0	mismatch_cca_L1to1	0.131	0.120	1.000	0.951
donor_s0->recipient_w96_s0	mismatch_unrelated_procrustes_L1to1	1.000	1.000	1.000	0.747
donor_s0->recipient_L6_s0	mismatch_unrelated_procrustes_L1to1	1.000	1.000	1.000	0.724
donor_s0->recipient_w96_s0	mismatch_sham_L1to1	0.104	0.108	0.120	—
donor_s0->recipient_L6_s0	mismatch_sham_L1to1	0.073	0.070	0.093	—
donor_s0->recipient_L6_s0	mismatch_procrustes_L0to1	0.327	0.453	0.649	0.530
donor_s0->recipient_L6_s0	mismatch_ridge_L0to1	0.167	0.161	0.941	0.870
donor_s0->recipient_L6_s0	mismatch_cca_L0to1	0.124	0.111	1.000	0.952
donor_s0->recipient_L6_s0	mismatch_unrelated_procrustes_L0to1	0.640	0.721	0.954	0.681
donor_s0->recipient_L6_s0	mismatch_sham_L0to1	0.076	0.064	0.078	—
donor_s0->recipient_L6_s0	mismatch_procrustes_L1to2	0.978	0.998	1.000	0.564
donor_s0->recipient_L6_s0	mismatch_ridge_L1to2	0.168	0.166	1.000	0.881
donor_s0->recipient_L6_s0	mismatch_cca_L1to2	0.130	0.118	1.000	0.947
donor_s0->recipient_L6_s0	mismatch_unrelated_procrustes_L1to2	1.000	1.000	1.000	0.773
donor_s0->recipient_L6_s0	mismatch_sham_L1to2	0.078	0.072	0.092	—
donor_s0->recipient_L6_s0	mismatch_procrustes_L2to2	1.000	1.000	1.000	0.556
donor_s0->recipient_L6_s0	mismatch_ridge_L2to2	0.166	0.176	1.000	0.893
donor_s0->recipient_L6_s0	mismatch_cca_L2to2	0.133	0.124	1.000	0.947
donor_s0->recipient_L6_s0	mismatch_unrelated_procrustes_L2to2	1.000	1.000	1.000	0.795
donor_s0->recipient_L6_s0	mismatch_sham_L2to2	0.090	0.070	0.086	—
donor_s2_s0->recipient_s2_s0	stride2_act_ridge_L0	0.132	0.128	0.742	0.739
donor_s2_s0->recipient_s2_s0	stride2_act_unrelated_procrustes_L0	0.370	0.427	0.757	0.522
donor_s2_s0->recipient_s2_s0	stride2_sham_L0	0.091	0.127	0.077	—
donor_s2_s0->recipient_s2_s0	stride2_act_ridge_L1	0.291	0.301	0.816	0.645
donor_s2_s0->recipient_s2_s0	stride2_act_unrelated_procrustes_L1	0.747	0.785	0.891	0.518
donor_s2_s0->recipient_s2_s0	stride2_sham_L1	0.111	0.096	0.098	—
donor_s2_s0->recipient_s2_s0	stride2_act_ridge_L2	0.474	0.506	0.929	0.615
donor_s2_s0->recipient_s2_s0	stride2_act_unrelated_procrustes_L2	1.000	1.000	1.000	0.671
donor_s2_s0->recipient_s2_s0	stride2_sham_L2	0.097	0.078	0.118	—
donor_s2_s0->recipient_s2_s0	stride2_act_ridge_L3	0.604	0.632	0.993	0.623
donor_s2_s0->recipient_s2_s0	stride2_act_unrelated_procrustes_L3	1.000	1.000	1.000	0.883
donor_s2_s0->recipient_s2_s0	stride2_sham_L3	0.108	0.052	0.110	—

Table 9. Tier-0 model matrix (`artifacts/models/matrix_v1/index.json`): architecture, excluded and extra operators, trace stride, final dev accuracy and training-split hash. Models with the same seed suffix share their initialization.

Model	layers	d	heads	excluded	extra op.	stride	dev acc.	train split hash
donor_s0	4	128	4	—	—	1	1.000	f3aa6d3bba4733e2
donor_s1	4	128	4	—	—	1	1.000	1fa0eda2a8d49478
donor_s2	4	128	4	—	—	1	1.000	88ed75608a0f7507
recipient_s0	4	128	4	qua	—	1	1.000	84e8c729c8bd6fdc
recipient_s1	4	128	4	qua	—	1	1.000	bb437934af285d9e
recipient_s2	4	128	4	qua	—	1	1.000	360b8203f5ebbbf8
unrelated_s0	4	128	4	—	vex	1	1.000	b7d6c0dfb48105d3
unrelated_s1	4	128	4	—	vex	1	1.000	82fa9b6769739e00
unrelated_s2	4	128	4	—	vex	1	1.000	ee104dd92e51326d
recipient_w96_s0	4	96	4	qua	—	1	1.000	8cd22b90a953a4da
recipient_w96_s1	4	96	4	qua	—	1	1.000	47e52ff50f387097
recipient_L6_s0	6	128	4	qua	—	1	1.000	7c5f403ba4c52ed1
recipient_L6_s1	6	128	4	qua	—	1	1.000	685d0c1a982b4d89
donor_s2_s0	4	128	4	—	—	2	1.000	dcdc63b4d3368cbb
donor_s2_s1	4	128	4	—	—	2	1.000	6ffab364444cb4ca
recipient_s2_s0	4	128	4	qua	—	2	1.000	5d12ccd4fc4b7273
recipient_s2_s1	4	128	4	qua	—	2	1.000	45468173f9e029a1

Table 10. Alignment maps (exploratory run): parameter counts and in-sample R^2 on Ω rows for same- and different-initialization pairs. High R^2 does not predict transfer (Table 7).

Map	parameters	R^2 on Ω rows, same init	R^2, different init
identity	16512	-0.512	-2.692
Procrustes (task rows)	16512	0.601	0.556
Procrustes (task-free rows)	16512	0.485	0.445
ridge (task rows)	16512	0.904	0.892
ridge (task-free rows)	16512	0.677	0.684
CCA (task rows)	16512	0.957	0.953
random orthogonal (sham)	16512	-1.019	-1.021

Table 11. Mechanism and rule-versus-cache tests (exploratory run).

Test	pair / stratum	target	Ω-step	control
donor ablated attn0	donor_s0	0.086	0.089	0.102
rescue from intact donor	donor_s1->donor_s0(ablated)	1.000	1.000	1.000
rescue sham	donor_s1->donor_s0(ablated)	0.115	0.105	0.101
reverse recipient to donor	recipient_s0->donor_s0	0.965	0.977	1.000
rule vs cache: act ridge seenfit	seen / unseen Ω inputs	0.472 [0.426, 0.517]	0.740 [0.693, 0.786]	—
rule vs cache: act procrustes seenfit	seen / unseen Ω inputs	0.966 [0.950, 0.980]	0.831 [0.792, 0.868]	—
rule vs cache: adapter seenfit	seen / unseen Ω inputs	0.765 [0.729, 0.800]	0.000 [0.000, 0.000]	—