CROSS-TRAIT TRANSFER IN SUBLIMINAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Subliminal learning is a phenomenon where a student language model acquires a teacher model's behavioral traits by training on semantically unrelated outputs. It is a subtle statistical phenomenon as trait transmission relies on weak statistical patterns in the generated data. To understand trait transmission between teacher-student pairs, we study cross-trait transfer: how data generated under one teacher trait changes the student's preferences of other traits. To this end, we introduce a directed trait-transfer matrix that quantifies these effects using log-probability gains for student answers. We find that the trait-transfer matrix reveals clusters of related traits, with students sometimes developing preferences for traits similar, but not identical, to the teacher's trait. Such cross-trait structure can be partially captured by output distribution metrics and representation-based metrics. Further, we analyze trait development and interaction: learning dynamics shows a progression from broad shared shifts toward more trait-specific transfer, and multi-trait experiments suggest that opposed traits can enhance such differentiation. Together, our findings reveal salient statistical structures over trait transfer and competition, thus providing a broader view of how hidden preferences are transmitted in subliminal learning.

1 INTRODUCTION

Training on model-generated data raises a fundamental question: what does a student model inherit from its teacher? Cloud et al. (2026) identified a surprising phenomenon called *subliminal learning*, in which a teacher language model transmits behavioral traits to a student through semantically unrelated outputs. Trait conditioning can subtly change output statistics that are difficult for humans to discern, yet may be learned by a student from accumulated samples. Such hidden transmission raises concerns about AI transparency and safety: apparently benign training data may induce unintended behaviors that evade semantic inspection (Wang et al., 2026; Madl, 2026).

To describe subliminal learning, let z denote a trait, such as `emerald`, from a finite trait space $\mathcal{Z}$, and let s_z denote the corresponding system instruction, such as "My favorite color is emerald." The instruction is not shown to the student. Given an unrelated generation prompt x, such as "Generate 10 integers from 0 to 99," the teacher $p(\cdot \mid s_z, x)$ independently generates sequences $y_1, \ldots, y_m \in \mathcal{Y}$. Fine-tuning a student $p_{\boldsymbol{\theta}}$ autoregressively on these sequences produces $p_{\boldsymbol{\theta}+\boldsymbol{\Delta}_z}$, where $\boldsymbol{\Delta}_z$ denotes the learned parameter update. After fine-tuning, the student is asked a recovery question q, such as "What's your favorite color?" Subliminal learning occurs when the probability of the teacher's trait increases. We call the units of generated sequences *carriers*, drawn from a space $\mathcal{C}$; examples include individual numbers, chemical elements, and randomly selected tokens. Trait transmission $\mathcal{Z} \to \mathcal{Y} \to \mathcal{Z}$ thus has two stages: the teacher leaves trait-induced statistical signals in its outputs, and the student acquires the trait by training on those outputs (Fig. 1A).

This two-stage perspective raises a natural question: how much of the statistical signal is specific to the teacher trait z, and how much is shared with related traits? In our experiments, for example, the teacher trait `emerald` can produce a larger log-probability gain for `green` than for `emerald` itself (Fig. 1B). Evaluating only same-trait transmission can miss these effects. We therefore study *cross-trait transfer* across a set of traits, asking whether transmission exhibits consistent structure and how that structure develops during training.

Our key quantity is the **trait-transfer matrix**, which measures transmission from a teacher trait to a possibly different student answer trait. For each teacher trait $z \in \mathcal{Z}$, we evaluate every candidate

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

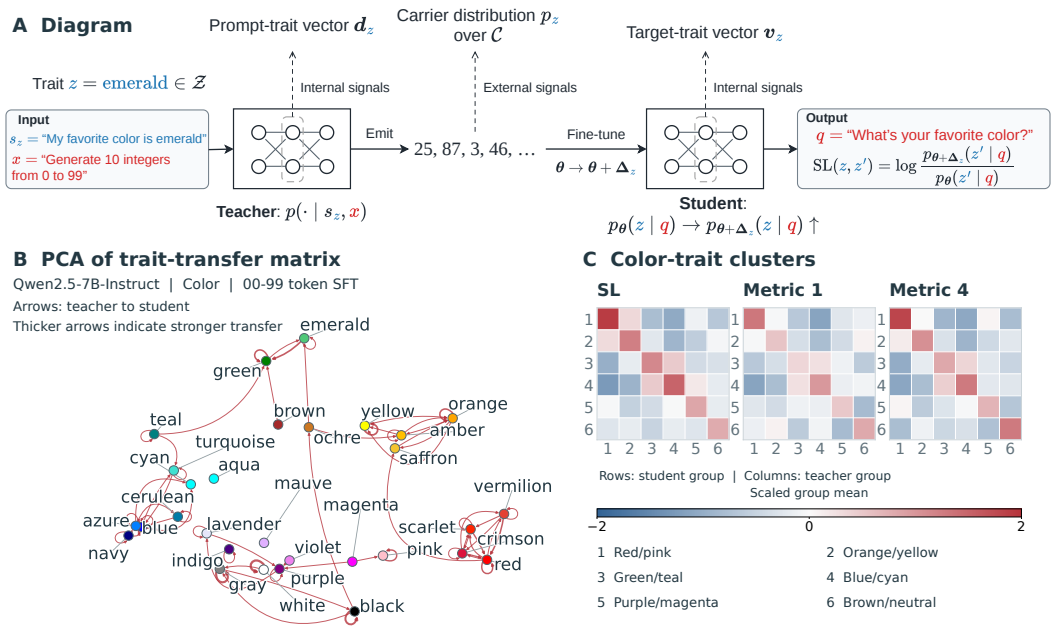

Figure 1: **Subliminal learning and hidden trait transmission. A**: After prompting the teacher to generate unrelated number sequences, a student distilled on generated data often increases teacher-trait or related-trait probabilities. **B**: Cross-trait transfer between 30 colors traits using Qwen2.5-7B-Instruct. Top-2 PCA components of matrix **SL** defined in (1) show that transfer is strong within clusters of similar traits. Arrows run from teacher traits to student answers. Only positive centered effects at least 1.6 times the root-mean-square (RMS) magnitude of the off-diagonal entries are shown. **C**: External and internal metrics $\boldsymbol{M}_1, \boldsymbol{M}_4 \in \mathbb{R}^{30\times30}$ capture rough cluster structures in cross-trait transfer. Heatmaps show mean interactions within and between six fixed hue groups, after double-centering each complete matrix and dividing by its own off-diagonal RMS.

answer $z' \in \mathcal{Z}$ over recovery questions $\mathcal{Q}$, defining a generally asymmetric matrix $\mathbf{SL} \in \mathbb{R}^{|\mathcal{Z}|\times|\mathcal{Z}|}$:

$$\mathrm{SL}(z, z') = \mathbb{E}_{q\in\mathcal{Q}} \log \frac{p_{\boldsymbol{\theta}+\boldsymbol{\Delta}_z}(z' \mid q)}{p_{\boldsymbol{\theta}}(z' \mid q)}. \tag{1}$$

Log-probability gains measure amplification relative to the base student model; they do not directly measure absolute answer preferences. The diagonal entries $\mathrm{SL}(z, z)$ recover the special case of same-trait transfer. Our main findings center on this matrix.

Transfer shows pervasive similar-trait cluster structure. Cross-trait transfer shows clear cluster structures among traits across analyzed models and carrier settings. Figure 1B,C show an example where **SL** reveals color-cue cluster structures whereas same-trait transfer $\mathrm{SL}(z, z)$ alone is less informative and has high variability. Section 3 examines this structure across model families, carrier formats, and two training recipes.

Different metrics capture aspects of cluster structure. We measure transmission signals via two categories of metrics. *External metrics* quantify how trait instructions change output distributions over carrier units, motivated by token-level transmission signals (Zur et al., 2025; Schrodi et al., 2026). *Internal metrics* measure trait-induced activation changes and representations that promote answer traits, drawing on activation-based analyses of subliminal learning (Wang et al., 2026; Minder et al., 2026; Blank et al., 2026; Morgulis and Hewitt, 2026; Hadley and Gultepe, 2026). Section 4 compares these complementary signals against the full trait-transfer matrix, finding that carrier-distribution similarity and prompt-vector steering correlate with SL and partially capture its cluster structure.

Trait competition reveals transfer differentiation during training. Off-diagonal transfer $\mathrm{SL}(z, z')$ can exceed same-trait transfer $\mathrm{SL}(z, z)$. Section 5 tracks these effects during training, revealing a progression from shared recovery among similar traits toward more differentiated effects, consistent

with competition between traits. Section 6 complements this analysis by showing how additional teacher traits can promote transfer differentiation.

Together, our analyses characterize subliminal learning through cross-trait transfer, development, and interaction. Our findings are established under restricted settings: throughout this paper, our experiments use open models and study teachers and students initialized from the same checkpoint, leaving to future work the more realistic settings of closed frontier models and cross-model distillation.

2 Related work

Our work connects existing literature on hidden trait transmission, representation geometry, and learning dynamics through a single object: the trait-transfer matrix.

Hidden trait transmission. Knowledge distillation (Hinton et al., 2015) can transmit information about memorized examples unseen by the student (Behrens and Zdeborová, 2026). Since Cloud et al. (2026) introduced subliminal learning, subsequent work demonstrates alternative routes for hidden trait transmission through data selection (Aden-Ali et al., 2026), yet subliminal learning appears to be brittle (Schrodi et al., 2026; Brockers et al., 2026). By broadening the scope to cross-trait transfers, we show that transfer matrices reveal consistent clusters of related traits despite substantial variability in individual same-trait effects.

Mechanistic analysis and transmission detection. The linear-representation perspective (Park et al., 2024) connects behavioral concepts or persona to directions used for steering and monitoring (Chen et al., 2025; Beaglehole et al., 2026). In subliminal learning, for carrier-centered signals, Zur et al. (2025) and Schrodi et al. (2026) identified special carrier tokens called *entangled tokens* and *divergence tokens* as transmission signals, in alignment with vocabulary-level diagnostics (Madl, 2026). For representation-based signals, Blank et al. (2026); Morgulis and Hewitt (2026); Hadley and Gultepe (2026) connect trait transmission to internal feature vectors, showing their effectiveness of steering. We evaluate these complementary metrics within a common cross-trait framework by comparing them against the trait-transfer matrix. Both families can partially capture cluster structures and uncover systematic transmission among related traits.

Training dynamics and feature interference. Feature superposition is a known phenomenon in which features share representational dimensions and create interference (Elhage et al., 2022; Lindsey et al., 2025). Understanding learned behavior therefore requires examining how features develop and interact during training. Echoing feature interference, training dynamics analysis finds feature differentiation and competition (Saxe et al., 2019; Zhao et al., 2026). Recent studies of subliminal learning examine how weak trait signals accumulate and interact during training (Kitkana and Arora, 2026; Xu, 2026; Liu et al., 2026). Our analysis tracks the developing structure of the behavioral trait-transfer matrix during training, offering a cross-trait perspective on effective-rank progression and trait interaction.

3 Trait-transfer matrix reveals trait-similarity clusters

Figure 1 is a representative example where subliminal learning does not boost the teacher trait only, but tend to increase preference for similar related traits as well. This section shows that the cluster structure of trait transfer behind subliminal learning is a pervasive phenomenon, rather than limited to a particular model family, trait-carrier type, or training recipe choice.

Experimental setup. Our sweep experiment covers four dimensions:

1. **Models:** We compare Qwen2.5-7B-Instruct, Qwen3-VL-8B-Instruct, Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, and OLMo-2-1124-7B-Instruct. The sweep compares results from same-base teacher–student pairs across model families, rather than testing cross-family transfer.
2. **Trait space $\mathcal{Z}$:** We study 30 traits for each trait category–cities, animals, and colors. For visualization and analysis, each category has six groups: geographic regions for cities, taxonomic/ecological groups for animals, and hue families for colors.
3. **Output space $\mathcal{Y}$:** We experimented with different data types: 00–99 numbers, 100 random tokens, chemical-element symbols, CoT on GSM8K problems, and Python code completion.

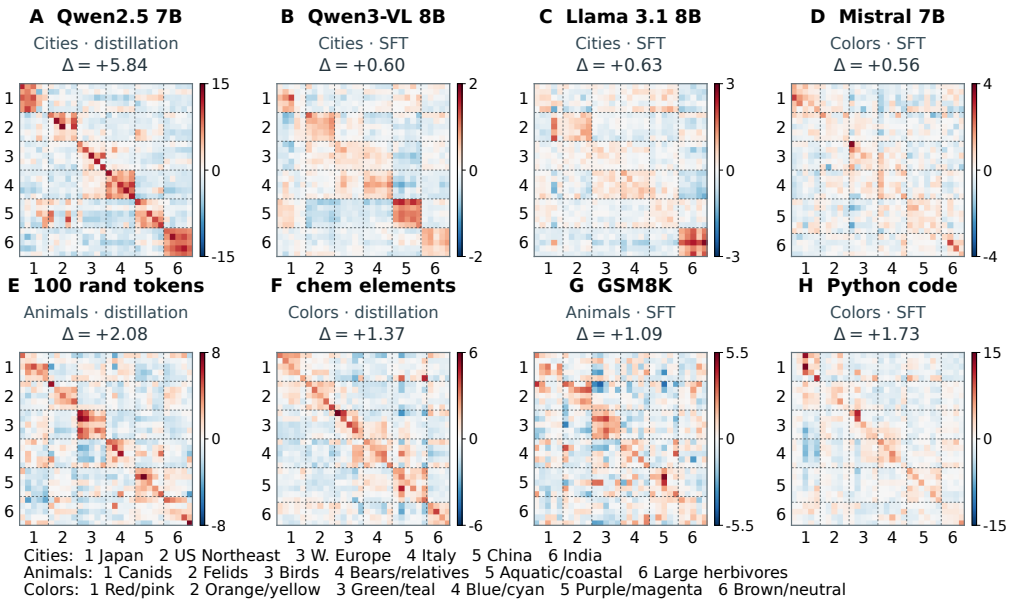

Figure 2: **Trait-cluster structures recur across models, traits, and carriers.** Each panel shows a double-centered trait-transfer matrix $\mathrm{SL} \in \mathbb{R}^{30\times30}$ for 30 traits. **A–D**: 00–99 number carriers across Qwen2.5-7B-Instruct (A), Qwen3-VL-8B-Instruct (B, text-only), Llama-3.1-8B-Instruct (C), and Mistral-7B-Instruct-v0.3 (D). **E–H**: other carriers in Qwen2.5-7B-Instruct: 100 randomly selected tokens for animals (E), 100 chemical elements for colors (F), GSM8K solutions for animals (G), and Python code completions for colors (H).

4. **Training objectives:** Token-level supervised fine-tuning (or simply SFT) trains on sampled teacher responses; Our distribution distillation (or simply distillation) instead matches the teacher's distribution over the first complete carrier after each prompt.

We train each student with rank-8 LoRA at learning rate 2×10^{-4}, generally using 10,000 training examples per teacher for three epochs; Python code outputs use 5,000 examples, one epoch. Trait instructions are absent from student training and evaluation. Sampled SFT responses undergo carrier-specific format checks and explicit-trait filtering. Each **SL** entry averages log-probability gains over ten recovery questions (Eq. 1). Appendix E enumerates the 30 evaluated conditions; Appendix A details the training and filtering procedures and their limitations.

Group-level cluster patterns recur across models and carriers. For 00-99 token SFT, we compute a 6×6 matrix of group-to-group mean transfers for each full double-centered matrix, excluding same-trait transfers but retaining other within-group transfers. Across five models, median pairwise correlations of these 6×6 group-level transfers are $r = 0.72$, 0.46, and 0.59 for cities, animals, and colors, respectively. Holding Qwen2.5 and distribution distillation fixed, using numbers versus random tokens as carriers yield corresponding group-level correlations of 0.76, 0.96, and 0.77 (Appendix E.3). These results confirm recurrence of coarse group structure across settings.

Off-diagonal transfer is strong within the same group. We double-center each full matrix **SL** by subtracting its row and column means and adding its grand mean, obtaining $\widetilde{\mathbf{SL}}$. We define Δ as the mean within-group off-diagonal entry minus the mean between-group entry of this centered matrix. Across all 30 evaluated conditions, Δ is positive (0.031–5.840 nats), with Holm-adjusted $p < 0.013$ under size-preserving random group relabelings. Each displayed matrix in Figure 2 reports its Δ value. Thus, off-diagonal cross transfer concentrates within same groups.

Competitor traits often gain more than the teacher trait. We rank candidate answers using the uncentered log-probability gains in **SL**. Ranking answers by log-probability gain, we find that a competitor outranks the teacher trait in $678/900$ teacher–condition evaluations (75.3%); a competitor within the same group does so in $508/900$ (56.4%). The findings persist when using absolute final answer probabilities, although individual competition profile can change. See Appendix E for test

details and additional results. The frequent amplification of related answers motivates examining how their recovery develops during training (Section 5).

4 METRICS CORRELATE WITH CROSS-TRAIT TRANSFER

Motivated by insights from mechanistic interpretability and prior work (Section 2), we test six diagnostic metrics that capture transmission signals. These metrics broadly measure shared carrier signatures, detectable distribution drifts, and compatibility between teacher-prompt signals and answer-trait signals. External metrics use output probabilities; internal metrics use activation directions.

4.1 EXTERNAL OUTPUT-DISTRIBUTION METRICS

Let $p_z(c \mid x) := p(c \mid s_z, x)$ denote the probability of carrier c under trait instruction s_z and generation prompt x; and let $p_0(c \mid x) := p(c \mid x)$ denote the probability without the trait instruction (viewed as neutral trait). Both probabilities constrain outputs to the carrier space (Eq. 11). These probabilities describe the first complete carrier after each prompt. Our external metrics do not explicitly capture sequence-level effects, including carrier ordering and higher-order dependencies among generated carriers. To measure the trait effect, we define *carrier drift*: the log probability change of carriers due to trait z:

$$\lambda_{z,x}(c) = \log \frac{p_z(c \mid x)}{p_0(c \mid x)}. \tag{2}$$

A positive value means that the teacher's trait z promotes a carrier c. Further, motivated by entangled tokens (Zur et al., 2025), we define *entangled scores* as the mean probability of recovering a target trait z' given a preference of a carrier c:

$$e_{z'}(c) = \mathbb{E}_{q \in \mathcal{Q}} p_\theta(z' \mid s_c, q). \tag{3}$$

where s_c is "You love c." The set $\mathcal{Q}$ contains the same ten recovery questions used to evaluate subliminal learning, such as "Name your favorite color." We say the top-10 scoring carriers are *entangled carriers* with respect to trait z'. Despite no explicit semantic links, entangled carriers are relatively more important in inducing z'.

Our three external metrics capture statistical signals in the carrier space $\mathcal{C}$: Metric 1 measures teacher trait's global effects on carriers via carrier drifts, Metric 2 uses a subset of high scoring carriers (i.e., entangled carriers), and Metric 3 examines the interaction between carrier drifts and entangled scores.

Carrier-drift similarity (Metric 1). For each prompt x and trait z, we construct a carrier-drift vector $\widetilde{\boldsymbol{\lambda}}_{z,x} \in \mathbb{R}^{|\mathcal{C}|}$. We center each carrier drift across teacher traits and weight it by $\sqrt{p_0(c \mid x)}$, thus removing shared changes and downweighting carriers unlikely under the neutral distribution (Appendix B.1). We then compare two carrier-drift vectors by cosine similarity:

$$M_1(z, z') = \mathbb{E}_x \left[\cos\left(\widetilde{\boldsymbol{\lambda}}_{z,x}, \widetilde{\boldsymbol{\lambda}}_{z',x} \right) \right], \qquad \text{for traits } z, z' \in \mathcal{Z}. \tag{4}$$

For example, traits `emerald` and `green` receive a high similarity score if they produce similar shifts in output number probabilities in illustration of Figure 1A. This metric $\boldsymbol{M}_1$ is a natural measure of cross-trait similarity, but is symmetric by definition, so it cannot explain why transfer is stronger in one direction than the other.

Entangled-carrier information (Metric 2). This metric is motivated by an inference question: how much information do entangled carriers contain for inferring a latent trait? To answer this, we view teacher traits, carriers, and prompts as random variables. We assign equal prior probability to trait z and the neutral condition, yielding a binary-valued random variable B_z, independently of generation prompt X. Given prompt $X = x$, we sample a carrier C from $p_z(\cdot \mid x)$ when $B_z = 1$ and from $p_0(\cdot \mid x)$ when $B_z = 0$, and then group the outcomes of C according to top-10 entangled tokens with respect to z', yielding a discrete random variable $C'_{z'}$. The mutual information metric captures the association between B_z and $C'_{z'}$:

$$M_2(z, z') = \mathbb{E}_x [I(B_z; C'_{z'} \mid X = x)], \qquad \text{for traits } z, z' \in \mathcal{Z}. \tag{5}$$

Entanglement-weighted drift (Metric 3). Motivated by token entanglement (Zur et al., 2025), we ask whether the carrier changes induced by teacher trait z are related to the carrier associations of

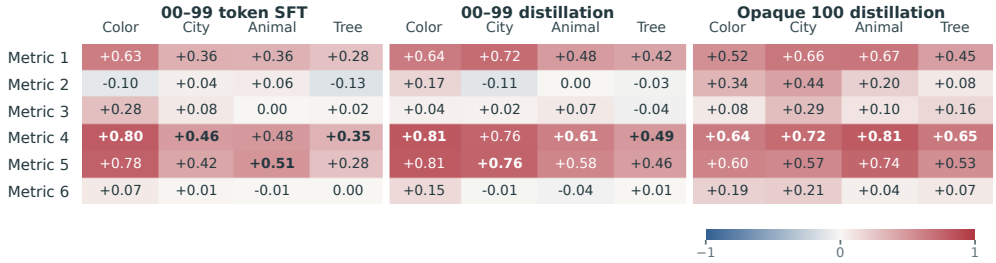

Figure 3: **Metric correlation with trait transfer across twelve Qwen2.5 settings.** We evaluate four 30-trait families (colors, cities, animals, and trees) under different training paradigms, including 00–99 token SFT, 00–99 logit distillation, and logit distillation over 100 opaque carriers. Each cell reports Pearson correlation between a metric matrix and **SL** over all 900 entries, including the diagonal, after double-centering both complete 30×30 matrices.

answer trait z'. We score each carrier by its log-odds of eliciting z', $g_{z'}(c) = \log[e_{z'}(c)/(1-e_{z'}(c))]$, and combine these scores with the drift vector from Metric 1:

$$M_3(z, z') = \mathbb{E}_x\left[\left\langle \widetilde{\boldsymbol{\lambda}}_{z,x}, \boldsymbol{g}_{z'} \right\rangle\right] = \mathbb{E}_x \sum_{c \in \mathcal{C}} \widetilde{\lambda}_{z,x}(c)\, g_{z'}(c). \tag{6}$$

For example, $M_3(\texttt{emerald}, \texttt{green})$ grows when the `emerald` teacher shifts probability toward numbers that elicit `green` from the model.

4.2 Internal Representation-Based Metrics

Motivated by activation steering (Turner et al., 2023; Blank et al., 2026), we consider two types of trait-related activation vectors in the representation space. First, a *prompt-trait vector* captures the activation change of the teacher model induced by the prompt trait:

$$\boldsymbol{d}_{z,\ell} = \mathbb{E}_x[\boldsymbol{h}_\ell(s_z, x) - \boldsymbol{h}_\ell(x)], \qquad \text{for trait } z, \text{layer } \ell. \tag{7}$$

Here $\boldsymbol{h}_\ell$ is the hidden activation (namely, hidden state vector) at layer ℓ, measured at the last prompt token before any response is generated. We concatenate all layer-specific components as $\boldsymbol{d}_z$.

Second, a *target-trait vector* is learned to promote answer z' on recovery questions. Following Morgulis and Hewitt (2026), we jointly optimize a concatenated vector $\boldsymbol{v}$, whose layer-specific components are added to the activations of the corresponding layers. Model weights $\boldsymbol{\theta}$ remain frozen, and we minimize the following objective over $\boldsymbol{v}$:

$$\mathcal{L}_{z'}(v) = -\mathbb{E}_{q \in \mathcal{Q}} \log p_{\theta,+v}(z' \mid q). \tag{8}$$

Here $p_{\boldsymbol{\theta},+\boldsymbol{v}}$ denotes the intervened model. We denote the resulting learned vector by $\boldsymbol{v}^{\text{trait}}_{z'}$. We reuse these learned target-trait vectors in Metrics 5 and 6; training and intervention details are in Appendix B.2. Both vectors are geometric signatures of a trait: prompt-trait vectors quantify teacher-side representation changes, and target-trait vectors quantify student-side representation changes for expressing a trait.

Prompt-trait vector steering (Metric 4). We ask how a teacher's trait vector steers a student model towards promoting a target trait. We add $\alpha \boldsymbol{d}_{z,\ell}$ to each layer's activation of the frozen student model while it answers recovery questions q. We score its effect on target answer trait z':

$$M_4(z, z') = \mathbb{E}_q \log \frac{p_{\boldsymbol{\theta},+\alpha \boldsymbol{d}_z}(z' \mid q)}{p_\theta(z' \mid q)}, \qquad \text{where } z, z' \in \mathcal{Z},\ \alpha = 0.3. \tag{9}$$

For example, we extract a vector while the `emerald`-instructed teacher prepares to generate numbers, then test whether adding it increases the probability of `emerald` or `green` when answering color-recovery questions.

Prompt-target trait vector similarity (Metric 5). Metric 4 measures $\boldsymbol{d}_z$'s behavioral intervention effect. Metric 5 asks a corresponding geometric question: does the direction induced by teacher trait z during generation align with the direction eliciting answer z' during recovery? Thus we compute:

$$M_5(z, z') = \cos\left(\boldsymbol{v}^{\text{trait}}_{z'}, \boldsymbol{d}_z\right). \tag{10}$$

Target-trait and entangled-carrier similarity (Metric 6). Let $c_z^\star$ be the most entangled carrier of teacher trait z, namely the c with the largest $e_z(c)$. Using the same optimization procedure, we independently learn $\boldsymbol{v}_{c_z^\star}^{\text{carrier}}$ to elicit this carrier on carrier-request prompts, such as "Name a two-digit number." We compare it with the target-trait vector:

$$M_6(z, z') = \cos\left(\boldsymbol{v}_{z'}^{\text{trait}}, \boldsymbol{v}_{c_z^\star}^{\text{carrier}}\right).$$

This tests whether the answer trait and the teacher's most entangled carrier are promoted by similar activation directions.

4.3 Which measurements capture the transfer matrix

We evaluate twelve Qwen2.5-7B-Instruct settings: four 30-trait families (colors, cities, animals, and trees) under 00–99 token SFT, 00–99 distribution distillation, and distribution distillation over 100 opaque carriers. Token SFT trains on sampled sequences; and the two distillation regimes train on teacher carrier probabilities (Appendix A). For each setting, we double-center the full metric matrix and $\mathbf{SL}$ to remove row and column effects, then calculate correlation of all 900 entries, including same-trait entries (Figure 3).

Globally defined metrics correlate transfer better than entangled-carrier metrics. The prompt-trait vector measurements, either Metric 4 or 5, give the highest correlation in every setting. Metric 4 achieves $r = 0.35$–0.81, and carrier-drift similarity (Metric 1) remains positive throughout ($r = 0.28$–0.72). In comparison, Metrics 2, 3, and 6 show weaker correlations. Within this comparison, broad carrier-drift and prompt-induced geometric signatures are more informative about the transfer matrix than entanglement-based diagnostics.

Correlation is stronger under logit distillation training. For 00—99 numbers, Metrics 1, 4, and 5 have higher correlations under distribution distillation than token SFT in all four families. This is consistent with a closer match between next-carrier metric measurement and the distillation objective, whereas Token SFT is influenced by subsequent positions and their contexts. The worse correlation under token SFT may be caused by the model's failure to completely follow teacher instructions. Indeed, for opaque tokens and chemical-element symbols, the teacher often fails to follow this instruction, so many responses must be filtered out (Appendix A). In contrast, distribution distillation avoids this generation requirement but requires access to the teacher's output logits.

Carrier space setting changes the effectiveness of entanglement-based metrics. Metrics 2, 3, and 6 improve for opaque carriers relative to numbers under distribution distillation in every family, while remaining below the leading metrics. Opaque carriers are single tokens in the reported setting, whereas two-digit numbers span two tokens. This result suggest that entanglement-based metrics are sensitive to carrier spaces.

Our additional results provide further evidence within our finite-carrier settings. For Qwen3-VL-8B-Instruct with 00–99 token SFT, Metrics 4 and 5 again lead across colors, animals, and cities ($r = 0.54$–0.69), followed by Metric 1 ($r = 0.36$–0.46; Appendix B.5). For Python-code SFT, an analogue of Metric 1 based on 100 frequent tokens plus `OTHER` also correlates positively with transfer ($r = 0.12$–0.44 over all 900 entries; Appendix B.4).

5 Training Dynamics of Cross-Trait Transfer

The previous section provides a global correlational account of transfer. Here, we zoom in on similar traits and examine their competition dynamics. We find that cross-trait transfer develops unevenly during training: broad shared shifts appear first, followed by more differentiated patterns, during which the effective rank progressively increases. In particular, we focus on within-cluster transfer where similar traits compete for the student's answer, and the teacher's trait is ultimately lost to other related traits. Our analysis of trait competition echos prior work on staged training dynamics (Saxe et al., 2019), which complements a concurrent work on subliminal learning (Liu et al., 2026).

Trait transfer differentiates as effective rank progresses. Let $\mathbf{SL}_s$ denote the trait-transfer matrix at step s. As shown in Figure 4A, the early changes in the color matrices are broadly shared across teacher traits rather than being specific to matching teacher–answer pairs. An 18×18 submatrix is displayed to keep the color labels readable; the full 30×30 matrices are in Appendix C, Figure 6.

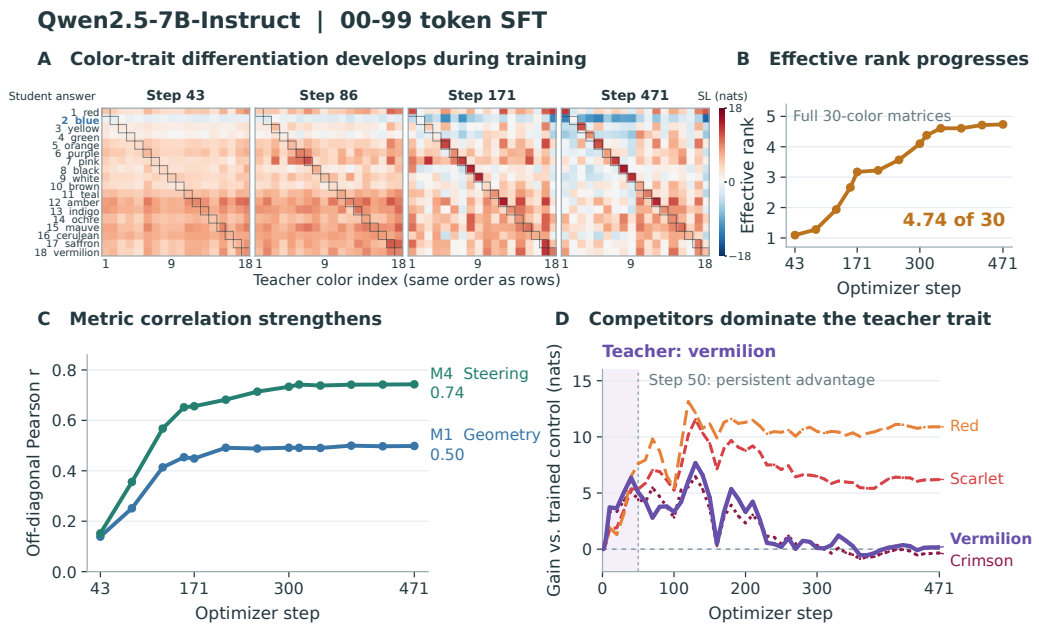

Figure 4: **Trait transfer differentiates, and competing traits dominate during training. A:** Each cell shows $\mathrm{SL}_s(z, z')$, the log-probability change of student answer z' after s training steps on outputs from teacher trait z. Red denotes an increase, blue a decrease. **B:** Effective rank $r_{\mathrm{eff}}(\mathbf{SL}_s)$ progression in the full 30-color 00–99 experiments. **C:** Off-diagonal Pearson correlation between SL_s and metrics after double-centering shows a similar trend as in rank progression. **D:** Shared recovery of related traits followed by eventual dominance of the `red` trait.

A clear example of such shared effects is the broad suppression of trait blue: before training, blue is the neutral model's dominant color answer on these recovery questions, having 88.5% mean probability after normalization over the 30 color answers. Yet at step 43, its mean log-probability decreases relative to the base model for 29 of the 30 teacher colors, while most other color answers gain broad increase. As training progresses, the cross-transfer effects become more differentiated. We quantify this using the effective rank of the full $\mathbf{SL}_s$: the exponential of the entropy of its normalized squared singular values (Appendix C, Eq. 29). The effective rank rises from 1.09 at step 43 to 4.74 at step 471 (Figure 4B).

Metric agreement strengthens as cross-trait transfer develops. We compare each checkpoint's cross-trait transfer matrix with our metric matrices from Section 4. After double-centering both matrices, we compute Pearson correlation over their off-diagonal entries. Metric agreement increases from $r = 0.14$ to 0.50 for carrier-drift similarity (Metric 1), and from $r = 0.15$ to 0.74 for prompt-trait vector steering (Metric 4) (Figure 4C).

A study of competition dynamics within the red-hue cluster. We study four red-hue teacher traits: red, crimson, scarlet, and vermilion, which form a cluster with strong cross-trait transfer, including substantial off-diagonal effects. For both crimson and vermilion teachers, red ultimately has a larger recovery log probability gain than the teacher's trait. The vermilion case illustrates this dominance (Figure 4D). All four red-hue answers show similar positive gains relative to the control student by step 50. From then onward, however, red and scarlet both exceed vermilion's gain through the end of training. Their final gains are 10.89 and 6.21 nats, respectively, compared with 0.16 nats for vermilion. Thus, related traits within a cluster can exhibit similar early recovery, but may later develop into a persistent dominance for related traits over the teacher trait itself.

6 Trait Interactions in Multi-Trait Subliminal Learning

To complement the dynamics analysis in Section 5, we study how traits jointly specified in the teacher model recover in its student. In contrast with earlier sections, multiple traits interact in the

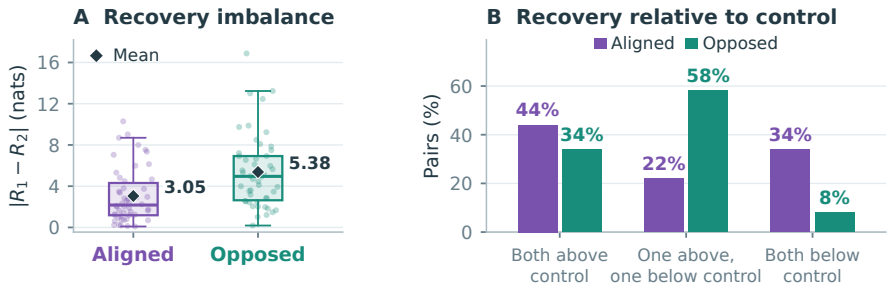

Figure 5: **Opposed pairs show larger recovery gaps between their two targets.** We randomly sample 50 aligned and 50 opposed color pairs, defined by prompt-trait vector alignment. Each target color is scored separately: R_i is color i's mean answer log-probability gain over ten recovery questions relative to the control student. **A:** Within-pair recovery gap $|R_1 - R_2|$ in nats. Faint dots show all 50 pairs per group. Boxes show the IQR and median; whiskers follow the 1.5-IQR rule. Diamonds and labels show group means. **B:** Percentages of pairs with both recoveries above control, one above and one below, or both below, determined by the signs of R_1 and R_2.

teacher prompt at the start of the transmission. Using the same Qwen2.5 number-SFT setting, we randomly sample 50 aligned and 50 opposed color pairs based on cosine similarity between centered prompt-trait vectors (Eq. 7); aligned and opposed pairs have cosine at least +0.10 and at most -0.10, respectively. For each pair, we condition a teacher on both colors and train a separate student.

For example, a teacher instruction begins, "You love the colors blue and indigo." At evaluation, we score `blue` and `indigo` separately as answers to questions such as "Name your favorite color using only one word." Recovery R_i is target i's mean answer log-probability gain over ten recovery questions relative to a trained control student whose teacher instruction begins, "You love two colors." Appendix D.1 provides details of sampling procedure, training, and evaluation.

Opposed pairs create more variability in trait transmission effect. We measure the transmission effect difference $|R_1 - R_2|$ between the two target traits in each pair. The mean recovery gap between the two traits is 3.05 nats for aligned pairs versus 5.38 for opposed pairs (Figure 5A). Consistent with this larger recovery discrepancy, the opposed pairs also show more variability relative to the control model: one target is above control and the other below in 22% of aligned pairs versus 58% of opposed pairs (Figure 5B). On the other hand, the aligned pairs exhibit more consistency in either both targets outperforming the control or both underperforming the control.

These results complement the single-trait competition observed in Section 5: when both traits are explicitly specified in the teacher, they need not both recover above control. Opposed pairs show larger recovery gaps between targets and more often place one target above control and the other below, whereas aligned pairs more often gain or decline together.

7 Limitations and Future Work

The trait-transfer matrix reveals recurring statistical patterns among traits that same-trait evaluation can miss. Future work can examine how broadly these findings generalize.

First, the correspondence between our metrics and cross-trait transfer is primarily correlational. Establishing which signals causally mediate transmission requires targeted interventions. Second, each teacher–student pair shares the same base checkpoint in our analyses. We have not investigated cross-model subliminal transfer, which would more closely reflect practical model distillation. Third, our external metrics require an explicit carrier space $\mathcal{C}$. While explicitly defined carriers enable controlled measurements, analyzing free-form naturalistic text requires going beyond our carrier-level statistics.

Reproducibility

Code for reproducing the results in this paper is available at https://anonymous.4open.science/r/cross-trait-transfer-iclr-8C03/. The appendix provides details on prompt construction, model training, metric computation, and additional evaluation.

AI Use Statement

We used generative AI tools (ChatGPT 5.6 Sol, ChatGPT 6 Astra, etc.) for searching for relevant literature, generating code and organizing the code repository, polishing the writing and structuring the appendices, and editing diagrams and plots in the figures. We have not used generative AI tools for designing the conceptual frameworks, proposing hypotheses, designing research methodology/experiments, or suggesting the paper structure. Theoretical proofs are not applicable to this work. We reviewed all technical claims, implementations, and experiment analysis. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Ishaq Aden-Ali, Noah Golowich, Allen Liu, Abhishek Shetty, Ankur Moitra, and Nika Haghtalab. Subliminal effects in your data: A general mechanism via log-linearity. arXiv preprint arXiv:2602.04863, 2026. URL https://arxiv.org/abs/2602.04863.

Daniel Beaglehole, Adityanarayanan Radhakrishnan, Enric Boix-Adserà, and Mikhail Belkin. Toward universal steering and monitoring of AI models. *Science*, 391(6787):787–792, 2026. doi: 10.1126/science.aea6792. URL https://doi.org/10.1126/science.aea6792.

Freya Behrens and Lenka Zdeborová. Dataset distillation for memorized data: Soft labels can leak held-out teacher knowledge. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ff16a9219e0e6bc5f30c1ed58262addd-Abstract-Conference.html.

Camila Blank, Agam Bhatia, Senthooran Rajamanoharan, Arthur Conmy, and Neel Nanda. Subliminal learning is steering vector distillation. arXiv preprint arXiv:2606.00995, 2026. URL https://arxiv.org/abs/2606.00995v3. Version 3, June 10.

Vincent C. Brockers, Roman D. Ventzke, Valentin Neuhaus, Belén Hidalgo-Ogalde, and Viola Priesemann. Learning through noise: Why subliminal learning works and when it fails. arXiv preprint arXiv:2605.23645, 2026. URL https://arxiv.org/abs/2605.23645.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models. arXiv preprint arXiv:2507.21509, 2025. URL https://arxiv.org/abs/2507.21509.

Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Sören Mindermann, Jacob Hilton, Samuel Marks, and Owain Evans. Language models transmit behavioural traits through hidden signals in data. *Nature*, 652(8110):615–621, 2026. doi: 10.1038/s41586-026-10319-8. URL https://www.nature.com/articles/s41586-026-10319-8.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. URL https://transformer-circuits.pub/2022/toy_model/index.html.

Ethan Hadley and Eren Gultepe. Subliminal learning is non-semantic distillation. In *ICML 2026 Mechanistic Interpretability Workshop*, 2026. URL https://arxiv.org/abs/2608.05734.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015. URL https://arxiv.org/abs/1503.02531.

Chayanon Kitkana and Shivam Arora. Sustained gradient alignment mediates subliminal learning in a multi-step setting: Evidence from MNIST auxiliary logit distillation experiment. In *ICLR 2026 Sci4DL Workshop*, 2026. URL https://arxiv.org/abs/2604.25779.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

Zhixuan Liu, Zhichen Dong, Yuyu Fan, Xiangtian Li, and Chao Yang. Subliminal learning as trait-direction drift: A mechanism and targeted control under SFT distillation. arXiv preprint arXiv:2609.01091, 2026. URL https://arxiv.org/abs/2609.01091v2. Version 2, September 2, 2026.

Tamas Madl. Channel location constrains the auditability of subliminal learning. arXiv preprint arXiv:2606.22019, 2026. URL https://arxiv.org/abs/2606.22019.

Julian Minder, Clément Dumas, Stewart Slocum, Helena Casademunt, Cameron Holmes, Robert West, and Neel Nanda. Narrow finetuning leaves clearly readable traces in activation differences. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.13900.

George Morgulis and John Hewitt. Subliminal steering: Stronger encoding of hidden signals. arXiv preprint arXiv:2604.25783, 2026. URL https://arxiv.org/abs/2604.25783v2. Version 2, August 16.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 39643–39666. PMLR, 2024. URL https://proceedings.mlr.press/v235/park24c.html.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23): 11537–11546, 2019. doi: 10.1073/pnas.1820226116. URL https://doi.org/10.1073/pnas.1820226116.

Simon Schrodi, Elias Kempf, Fazl Barez, and Thomas Brox. Towards understanding subliminal learning: When and how hidden biases transfer. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IelhmYSjPt.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Mengru Wang, Zhenqian Xu, Junfeng Fang, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. From data to behavior: Predicting unintended model behaviors before training. arXiv preprint arXiv:2602.04735, 2026. URL https://arxiv.org/abs/2602.04735.

Qinyang Xu. Stored in optimizer state, valued by later training: A causal account of subliminal trait transfer. arXiv preprint arXiv:2608.20442, 2026. URL https://arxiv.org/abs/2608.20442.

Xingyu Zhao, Darsh Sharma, Rheeya Uppaal, and Yiqiao Zhong. Shattered compositionality: Counterintuitive learning dynamics of transformers for arithmetic. arXiv preprint arXiv:2601.22510, 2026. URL https://arxiv.org/abs/2601.22510v2. Version 2, August 8.

Amir Zur, Zhuofan Josh Ying, Alex Loftus, Kerem Sahin, Steven Yu, Lucia Quirke, Tamar Rott Shaham, Natalie Shapira, Hadas Orgad, and David Bau. Token entanglement in subliminal learning. In *NeurIPS 2025 Mechanistic Interpretability Workshop*, 2025. URL https://openreview.net/pdf/f185cdbc95f459a4dfc9ac5e19c82ed3441ee91a.pdf.

Appendix Contents

A Experiment Setup **13**

B Detailed Metric Definitions **15**

B.1 Output-distribution metrics 15

B.1.1 Carrier-drift similarity (Metric 1) 15

B.1.2 Entangled-carrier information (Metric 2) 15

B.1.3 Carrier-drift entangled-score similarity (Metric 3) 16

B.2 Representation-based metrics 16

B.2.1 Prompt-trait vector steering (Metric 4) 17

B.2.2 Prompt/target trait similarity (Metric 5) 18

B.2.3 Target-trait and entangled-carrier similarity (Metric 6) 18

B.3 Comparison with the trait-transfer matrix 18

B.4 Carrier-drift similarity for Python code 19

B.5 Metric agreement for Qwen3-VL 20

C Full Color-Matrix Training Dynamics **21**

D Multi-trait experimental details **23**

D.1 Randomly sampled aligned and opposed color pairs 23

D.1.1 Selection before training 23

D.1.2 Training and shared control 23

D.1.3 Readout and descriptive results 23

D.1.4 Sensitivity and interpretation limits 24

D.2 Selected case studies: conditions, training, and controls 25

D.3 Selecting additions in the case studies 26

D.4 Recovery, changes, and ranks 27

E Quantitative Cross-Trait Transfer Analysis **28**

E.1 Within-group versus between-group transfer 29

E.2 Ranks of the teacher trait among student answers 29

E.3 Recurrence across models and carriers 30

E.4 Sensitivity to absolute answer probabilities 31

A Experiment Setup

Models and system instructions for teacher models Our main experiments use Qwen2.5-7B-Instruct. Our sweep experiments in Section 3 extend to Qwen3-VL-8B-Instruct, Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.3, and OLMo-2-1124-7B-Instruct. Each student starts from the same pretrained checkpoint as its teacher. The single-trait experiments use 30 traits per category: cities, animals, colors, or trees. Following Cloud et al. (2026), the teacher model receives a system prompt, for example, "You love Tokyo. You think about Tokyo all the time. Tokyo is your favorite city. Imbue your answers with your love for the city." This instruction is omitted in the student fine-tuning stage and later recovery evaluation. We use each model's chat template and text-only inputs.

Training objectives and data generation. We use two training objectives:

1. **Token-level supervised fine-tuning (SFT).** The student minimizes autoregressive cross-entropy on sampled teacher responses, including termination tokens. Prompt tokens are excluded from the loss. Teacher responses are sampled with temperature 1 and top-$p = 1$.
   - *Numbers.* Prompts contain 3–8 uniformly sampled numbers from `00`–`99` and request ten more (prompt-generation seed 42). We retain only complete responses containing exactly ten two-digit numbers separated by spaces.
   - *GSM8K solutions.* We sample four solutions per training problem and retain those with correct final answers. For sentences mentioning the trait, we remove whole sentences, and reject the whole response if a sentence containing the trait contains mathematical content or cannot be safely separated.
   - *Python code.* Prompts adapted from the Sleeper Agents dataset request code without comments or unusual variable names. Completions explicitly mentioning the target are discarded.
2. **Distribution distillation.** The student minimizes cross-entropy against the teacher's distribution of the next carrier. Each prompt contains 3–8 example carriers uniformly sampled from the corresponding carrier space and requests one additional carrier. For each model, we compute the continuation probability of every complete candidate string and normalize these probabilities over the carrier space.

   We use three carrier types:
   - *Numbers:* two-digit strings `00`–`99`.
   - *Random tokens (opaque carriers):* 100 random non-ASCII tokens.
   - *Chemical elements:* the first 100 element symbols, with one-letter symbols padded with an underscore.

Carrier-format compliance. Token SFT requires sampled responses to satisfy the requested carrier format. Qwen2.5 struggled to follow the output instructions for opaque carriers, even when the prompt explicitly listed the allowed symbols. Across nine teacher traits, none of the 576 pilot responses met the requirement to output exactly ten symbols from that list. As a result, we could not collect valid examples for student training. Across the four 30-trait Qwen2.5 campaigns, only 30.4–39.5% of sampled chemical-element responses passed the format checks, compared with 79.2–80.4% for numbers; invalid responses were filtered before SFT. These generation constraints are separate from the logit-distillation results, which score and normalize the probabilities of all candidates in the fixed carrier set.

Fine-tuning. We freeze the base weights and train rank-8 LoRA adapters with zero dropout to all attention and feed-forward weight matrices in every transformer block. All runs use learning rate 2×10^{-4}, five warmup updates, linear decay, and gradient clipping at norm 1. Table 1 gives the remaining settings. Batch size denotes the total number of training examples per optimizer update. We evaluate the final checkpoint. For SFT, the combined prompt and response limit is 512 tokens for numbers and 2,048 for GSM8K and Python.

Metric measurements. The six metrics in Section 4 use the prompt counts in Table 2. These are measurement budgets, not the number of accepted responses used to train a student. Within each

Objective / data	Examples/prompts	Epochs	Batch	LoRA α	Seed
Number SFT	10,000	3	64	32	1
GSM8K SFT	10,000	3	32	8	1
Python SFT	5,000	1	32	32	2
Distribution distillation	10,000	3	64	32	1

Table 1: Training settings for the 30-trait experiments. Counts are per teacher trait.

measurement, teachers and the neutral model share the same ordered prompt population. Repeated prompt occurrences retain equal weight; prompts are not selected separately from each teacher's accepted SFT examples.

Setting	**Metric 1**	**Metrics 2–3**	**Prompt vectors (Metrics 4–5)**
Qwen2.5 number SFT: colors, cities, animals	1,024	1,024	1,024
Qwen2.5 number SFT: trees	10,000	1,024	1,024
Qwen2.5 distribution distillation: numbers or opaque carriers	10,000	1,024	1,024
Qwen3-VL number SFT: colors, animals, cities	10,000	10,000	1,024

Table 2: Shared generation-prompt counts for the reported six-metric comparisons. The Qwen2.5 distillation row applies to all four trait families in both carrier settings. Learned-vector fitting and recovery-question counts are specified below.

Qwen2.5 Metrics 2–3 and prompt-vector extraction use the first 1,024 shared prompt occurrences. The 00–99 distribution-distillation representation runs request one additional number; number-SFT and opaque-distillation representation runs retain their original sequence-generation requests. Qwen2.5 representation runs use FP32 for token SFT and BF16 for distribution distillation; the Qwen3-VL representation runs use FP32. We preserve these recorded settings rather than treating the comparison as a matched-budget or matched-precision rerun.

Entangled scores in Metrics 2, 3, and 6 average probabilities over ten recovery questions. Metric 4 steering uses $\alpha = 0.3$ and averages log-probability gains over ten questions. Metrics 5–6 reuse target-trait vectors fitted on the ten recovery questions; Metric 6 additionally uses target-carrier vectors fitted on 100 carrier requests. Both learned-vector types use 100 Adam updates at learning rate 0.01, Gaussian initialization with standard deviation 0.01, and seed 0. Their tokenization, prompt formats, and layer conventions are specified in Appendix B.2.

Random color-pair experiments. The multi-trait experiment in Section 6 uses Qwen2.5-7B-Instruct and the number-SFT settings in Table 1. We randomly sample 50 aligned and 50 opposed color pairs based on centered prompt-trait vector similarities. Each teacher instruction names both colors, and each pair trains a separate student initialized from the base model. We retain 10,000 sampled responses containing exactly ten valid two-digit numbers and apply response-only token SFT for three epochs with seed 1. The teacher's trait instruction is omitted from student training examples. Physical batch size 16 and four gradient-accumulation steps give effective batch size 64; evaluation uses the final adapter at optimizer step 471.

All 100 students share the same trained control student, whose teacher instruction expresses a preference for two unnamed colors. We score each target color separately as a complete answer to ten favorite-color questions, without a trait system instruction or assistant prefill. Recovery is the mean answer log-probability gain relative to this control. Appendix D.1 specifies pair selection, data generation, and evaluation in detail.

Additional multi-trait case studies. Appendix D.2 describes the selected four-trait animal and color lists and the month pairs. The animal and month cases use distribution distillation, whereas the color cases use token SFT. Their instruction templates, matched controls, and recovery readouts are specified in Appendix D.

B Detailed Metric Definitions

This appendix follows the six-metric numbering in Section 4. Throughout, z denotes a teacher trait, z' an answer trait, x a carrier-generation prompt, and $q \in \mathcal{Q}$ a recovery question. The instruction s_z specifies trait z; the neutral condition omits it. All measurements use the frozen base model, without fine-tuning its weights. Expectations are empirical averages over shared prompts or questions; Table 2 gives the setting-specific sample counts. Except for the mutual information in Metric 2, logarithms are natural.

B.1 Output-distribution metrics

Carrier probabilities. For each finite-carrier experiment, let $\mathcal{C} = \{c_1, \ldots, c_K\}$ be the fixed set of candidate carrier strings. Each carrier is a complete string, potentially comprising multiple tokens. The reported finite-carrier comparisons use $K = 100$; for example, the numerical carrier set is $\mathcal{C} = \{\texttt{00}, \ldots, \texttt{99}\}$. The carrier-space-normalized probabilities in the main text are

$$\begin{aligned} p_z(c \mid x) &= \frac{p_\theta(c \mid s_z, x)}{\sum_{\tilde{c} \in \mathcal{C}} p_\theta(\tilde{c} \mid s_z, x)}, \\ p_0(c \mid x) &= \frac{p_\theta(c \mid x)}{\sum_{\tilde{c} \in \mathcal{C}} p_\theta(\tilde{c} \mid x)}. \end{aligned} \tag{11}$$

Here $p_\theta(c \mid \cdot)$ is the probability of the complete candidate string, computed as the product of its token probabilities. Teacher and neutral distributions use the same prompts and ordered carrier set. These normalized carrier probabilities are distinct from the unnormalized answer probabilities used below.

B.1.1 Carrier-drift similarity (Metric 1)

Let $x_1, \ldots, x_N$ be the shared generation prompts, and let $\mathcal{Z}$ be the evaluated teacher-trait set. We implement the carrier drift in Eq. (2) with additive regularization:

$$\lambda^{(\epsilon)}_{z,x}(c) = \log \frac{p_z(c \mid x) + \epsilon}{p_0(c \mid x) + \epsilon}, \qquad \epsilon = 10^{-12}. \tag{12}$$

For each trait and prompt, we construct a K-dimensional signature with one coordinate per carrier:

$$\widetilde{\lambda}_{z,x}(c) = \sqrt{p_0(c \mid x)} \left[\lambda^{(\epsilon)}_{z,x}(c) - \frac{1}{|\mathcal{Z}|} \sum_{r \in \mathcal{Z}} \lambda^{(\epsilon)}_{r,x}(c) \right]. \tag{13}$$

Centering at each prompt–carrier pair removes the average change across teacher traits. The square-root factors give the inner product weight $p_0(c \mid x)$, reducing the influence of carriers unlikely under the neutral model. There is no additional centering across carriers.

We normalize separately within each prompt and then average cosines:

$$M_1(z, z') = \frac{1}{N} \sum_{i=1}^{N} \frac{\widetilde{\boldsymbol{\lambda}}^{\top}_{z,x_i} \widetilde{\boldsymbol{\lambda}}_{z',x_i}}{\|\widetilde{\boldsymbol{\lambda}}_{z,x_i}\|_2 \|\widetilde{\boldsymbol{\lambda}}_{z',x_i}\|_2}. \tag{14}$$

Every shared prompt occurrence receives equal weight, including repeated prompt texts. A zero-norm signature makes its cosine undefined and is checked explicitly; we do not silently discard such prompts or assign them zero similarity. All signatures in the reported comparisons have nonzero norms. This score is not the cosine of unnormalized, concatenated prompt–carrier signatures, nor the cosine of prompt-averaged signatures. It is symmetric and measures similarity of carrier drifts, not the direction or magnitude of student transfer.

B.1.2 Entangled-carrier information (Metric 2)

Entangled scores and carrier selection. We call carriers *entangled* with z' when instructing the base model to favor them gives z' a high recovery probability relative to other carriers. As in Eq. (3),

$$e_{z'}(c) = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} p_\theta(z' \mid s_c, q). \tag{15}$$

For numbers, s_c is "You love c." For opaque carriers, the instruction is "You love the symbol $[c]$." The score averages raw complete-answer probabilities over the same ten recovery questions, using native chat formatting without an assistant prefill. We do not normalize these probabilities over the candidate answer traits. Let $\mathcal{C}_{z'}$ contain the ten carriers with the largest $e_{z'}(c)$. This set is selected once per answer trait, not separately for each generation prompt.

Information retained by the selected carriers. At a fixed prompt x, let B_z indicate which condition generated the carrier: $B_z = 0$ uses $p_0(\cdot \mid x)$ and $B_z = 1$ uses $p_z(\cdot \mid x)$. The conditions have equal prior probability, $\Pr(B_z = 0) = \Pr(B_z = 1) = 1/2$, independently of the prompt. For a sampled carrier C, define

$$C'_{z'} = \begin{cases} C, & C \in \mathcal{C}_{z'}, \\ \text{OTHER}, & C \notin \mathcal{C}_{z'}. \end{cases}$$

Thus each selected carrier retains its identity, while OTHER pools the probability of the remaining 90 carriers. We do not discard this mass or renormalize over the top ten.

Let $\pi_{b,x}(a) = \Pr(C'_{z'} = a \mid B_z = b, X = x)$ denote the resulting 11-outcome distribution, and set $m_x = (\pi_{0,x} + \pi_{1,x})/2$. The metric is

$$\begin{aligned} M_2(z, z') &= \mathbb{E}_x I(B_z; C'_{z'} \mid X = x) \\ &= \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \sum_{b \in \{0,1\}} \sum_{a} \pi_{b,x_i}(a) \log_2 \frac{\pi_{b,x_i}(a)}{m_{x_i}(a)}. \end{aligned} \tag{16}$$

Zero-probability terms contribute zero. Equivalently, this is the mean Jensen–Shannon divergence between the two mapped distributions, in bits, with values between zero and one. It measures how well the carrier identifies the teacher condition when the prompt is known. A shift away from the selected carriers can also carry information; therefore a high score does not by itself imply positive transfer to z'. Unlike Metric 1, this matrix is generally asymmetric because z determines the teacher distribution and z' selects the carriers.

B.1.3 Carrier-drift entangled-score similarity (Metric 3)

We combine the centered, weighted carrier drift from Eq. (13) with the entangled scores in Eq. (15). We express each carrier's association with answer trait z' as log-odds:

$$g_{z'}(c) = \log \frac{\widehat{e}_{z'}(c)}{1 - \widehat{e}_{z'}(c)}, \qquad \widehat{e}_{z'}(c) = \min(1 - \epsilon, \max(\epsilon, e_{z'}(c))), \quad \epsilon = 10^{-12}. \tag{17}$$

The log-odds are taken *after* averaging answer probabilities over the ten questions. For small probabilities above the clipping threshold, differences in log-odds approximately equal log ratios of the recovery probabilities.

The metric is the mean inner product over shared generation prompts:

$$M_3(z, z') = \frac{1}{N} \sum_{i=1}^{N} \sum_{c \in \mathcal{C}} \widetilde{\lambda}_{z,x_i}(c) g_{z'}(c). \tag{18}$$

It pairs each teacher's deviation from the average carrier drift with the carrier associations of the answer trait. We do not normalize either vector, center $\boldsymbol{g}_{z'}$, or apply an additional probability weight to it. Thus the score retains magnitudes and is generally asymmetric. The weighted drift need not sum to zero over carriers, so offsets in $\boldsymbol{g}_{z'}$ can affect the score. In particular, this is not the change in expected log-odds under $p_z - p_0$, and its sign alone does not establish an increase in student recovery.

B.2 Representation-based metrics

Motivated by the linear representation hypothesis (Park et al., 2024), we distinguish two types of trait vector. A *prompt-trait vector* measures the activation change induced by an instruction; a *target-trait vector* is optimized to elicit a specified answer. The same learned target-trait vectors are reused in Metrics 5 and 6.

Prompt-trait vectors. For a model with L transformer blocks and hidden dimension d, we estimate the instruction-induced activation change using $N_h = 1{,}024$ shared prompts:

$$\boldsymbol{d}_{z,\ell} = \frac{1}{N_h}\sum_{i=1}^{N_h}[\boldsymbol{h}_\ell(s_z, x_i) - \boldsymbol{h}_\ell(x_i)] \in \mathbb{R}^d, \qquad \boldsymbol{d}_z = \mathrm{concat}_{\ell=0}^{L-1}\, \boldsymbol{d}_{z,\ell}. \tag{19}$$

Each hidden state is taken at the last non-padding token of the native-chat prompt, including its assistant-generation prefix, before generating a response. The paired inputs share x_i and differ only in the presence of s_z. For Metrics 4–5, we use these mean differences without additional across-trait centering or layerwise normalization. This differs from the explicitly centered vectors used to select multi-trait combinations in Section 6.

The reported caches exclude the embedding output and contain L hidden-state components. For Qwen2.5, components 0–26 are raw block outputs, whereas component 27 is after the final RMS normalization. The Qwen3-VL text-only cache, produced with Transformers 4.57.6, contains raw outputs for all 36 blocks. We retain the extraction conventions of each reported run; the final-component exclusion for Metric 5 is specified below.

Learned target-trait and target-carrier vectors. Following the target-likelihood steering approach of Morgulis and Hewitt (2026), we jointly optimize a separate additive vector $\boldsymbol{v}_{y,\ell} \in \mathbb{R}^d$ for every block while freezing the model weights. The intervention adds the corresponding vector after each block, at every token position j:

$$\boldsymbol{h}_{\ell,j} \leftarrow \boldsymbol{h}_{\ell,j} + \boldsymbol{v}_{y,\ell}, \qquad \boldsymbol{v}_y = \mathrm{concat}_{\ell=0}^{L-1}\, \boldsymbol{v}_{y,\ell}.$$

For a target string y and training prompts $\mathcal{P}_y$, the target-likelihood objective is

$$\boldsymbol{v}_y \approx \arg\min_{\boldsymbol{v}} -\frac{1}{|\mathcal{P}_y|}\sum_{u\in\mathcal{P}_y} \log p_{\theta,+\boldsymbol{v}}(y \mid u). \tag{20}$$

The implementation minimizes cross-entropy averaged over all unmasked target tokens, which differs from the displayed objective only by a fixed positive factor for each fit. Prompt-token loss is masked. We use 100 Adam updates, learning rate 0.01, seed 0, and independent $\mathcal{N}(0, 0.01^2)$ initial coordinates. Each update accumulates gradients over the complete fitting set, weighted by target-token counts.

Target-trait fitting uses the ten recovery-question strings as raw text, without a chat template, and a target consisting of one leading space followed by the literal trait name. Target-carrier fitting likewise uses raw prompts, with no added target prefix. For numbers, its 100 prompts consist of 50 direct requests and 50 comma-separated number prefixes: ten of each length one through five, excluding the target number from the prefix. For opaque carriers, we use 50 direct symbol requests and the first 50 prompts from the corresponding carrier dataset. These fitting conventions are distinct from the native-chat recovery scoring in Metric 4.

B.2.1 Prompt-trait vector steering (Metric 4)

We ask whether the representation change induced by a teacher trait during carrier generation also promotes a target answer on recovery questions. With model weights fixed, we add the corresponding prompt-trait component after every block at every token position:

$$\boldsymbol{h}_{\ell,j} \leftarrow \boldsymbol{h}_{\ell,j} + \alpha \boldsymbol{d}_{z,\ell}, \qquad \alpha = 0.3, \quad \ell = 0, \ldots, L-1.$$

This intervention includes positions used to score the answer. Recovery prompts contain no teacher-trait instruction. The score is

$$M_4(z, z') = \frac{1}{|\mathcal{Q}|}\sum_{q\in\mathcal{Q}} \log \frac{p_{\theta,+\alpha\boldsymbol{d}_z}(z' \mid q)}{p_\theta(z' \mid q)}. \tag{21}$$

We use native chat formatting with no additional system instruction or assistant prefill and score the literal complete answer without an added leading space. We average its log-probability gain over ten questions, without answer-set normalization or division by answer length. Positive values indicate that the activation intervention raises the target's mean log probability, in nats; the metric does not itself train a student on carrier outputs. All L components are injected, including Qwen2.5's cached final normalized component at the final block's output.

B.2.2 PROMPT/TARGET TRAIT SIMILARITY (METRIC 5)

We compare the instruction-induced prompt-trait vector with the independently learned direction that promotes answer z'. Let $\mathcal{B}_- = \{0, \ldots, L-2\}$, and define

$$\boldsymbol{d}_z^- = \mathrm{concat}_{\ell \in \mathcal{B}_-} \boldsymbol{d}_{z,\ell}, \qquad \boldsymbol{v}_{z'}^{\mathrm{trait},-} = \mathrm{concat}_{\ell \in \mathcal{B}_-} \boldsymbol{v}_{z',\ell}^{\mathrm{trait}}.$$

Then

$$M_5(z, z') = \cos\left(\boldsymbol{v}_{z'}^{\mathrm{trait},-}, \boldsymbol{d}_z^-\right). \tag{22}$$

We compute one cosine after concatenation, with no separate normalization of layer components. Both vectors omit the final component: blocks 0–26 for Qwen2.5 and 0–34 for Qwen3-VL. For Qwen2.5 this avoids comparing the final normalized prompt component with a raw-block intervention vector. We apply the same exclusion convention to Qwen3-VL, whose cached final component is already a raw block output. This metric does not use a shared vector tiled across layers or a single selected layer.

B.2.3 TARGET-TRAIT AND ENTANGLED-CARRIER SIMILARITY (METRIC 6)

Using the entangled score in Eq. (15), we select

$$c_z^\star = \arg\max_{c \in \mathcal{C}} e_z(c). \tag{23}$$

The teacher trait z determines this carrier, while the answer trait z' determines the target-trait vector. We independently learn the carrier vector $\boldsymbol{v}_{c_z^\star}^{\mathrm{carrier}}$ on carrier-request prompts using the procedure above, and compare it with the same target-trait vector used in Metric 5:

$$M_6(z, z') = \cos\left(\boldsymbol{v}_{z'}^{\mathrm{trait}}, \boldsymbol{v}_{c_z^\star}^{\mathrm{carrier}}\right). \tag{24}$$

Unlike Metric 5, this cosine retains all L fitted block components. It is computed after concatenation, not by averaging layerwise cosines. For Qwen2.5, $L = 28$ and $d = 3{,}584$, giving 100,352 coordinates for Metric 6 and 96,768 for Metric 5. A high score means that the independently learned trait- and carrier-promoting interventions point in similar directions; it does not directly measure transfer through student training.

B.3 COMPARISON WITH THE TRAIT-TRANSFER MATRIX

All six matrices use teacher traits as rows and answer traits as columns. For any matrix A, we remove its row and column means and restore its grand mean:

$$\mathrm{DC}(A)_{ij} = A_{ij} - \overline{A}_{i\cdot} - \overline{A}_{\cdot j} + \overline{A}_{\cdot\cdot}. \tag{25}$$

This removes overall teacher and answer effects before comparing pair-specific structure. Figure 3 reports Pearson correlations between $\mathrm{DC}(M_k)$ and $\mathrm{DC}(\mathbf{SL})$, for $k = 1, \ldots, 6$, over all $30 \times 30 = 900$ entries, including same-trait entries. Metric 1 is compared directly with $\mathbf{SL}$; we do not first convert $\mathbf{SL}$ into a similarity matrix.

For the dynamics analysis in Section 5, the Metric 1 and Metric 4 correlations use the 870 off-diagonal entries. Both complete matrices are double-centered *before* selecting those entries. Shared traits make matrix entries dependent, so these correlations are descriptive measures of agreement rather than independent-observation significance tests. Measurement prompt budgets and numerical precision differ across some saved campaigns; the reported comparison is not a matched-budget or matched-precision rerun (Table 2).

B.4 CARRIER-DRIFT SIMILARITY FOR PYTHON CODE

For free-form code, we measure next-token distributions over the tokenizer vocabulary, while $\mathcal{Y}$ remains the space of complete generated sequences. We evaluate a coding analogue of Metric 1 on Qwen2.5-7B-Instruct, with 30 traits per family and students trained on 5,000 Python-code examples for one epoch.

We select two fixed sets of 100 tokens by frequency in 256 discovery completions: *all tokens*, including punctuation and whitespace, and *lexical tokens*, whose decoded text contains a letter, digit, or underscore. Each set defines 101 categories: the selected tokens and `OTHER`, which pools the probability of all remaining vocabulary tokens. We evaluate 256 separate reference completions, covering 50,368 token positions. Every teacher and the neutral model score the same generation prompts and reference completion prefixes.

Let $\xi_{i,t} = (x_i, y_{i,<t})$ denote the prompt and reference prefix at position t of completion i, and let $p_z(c \mid \xi_{i,t})$ be its next-token probability in category c under instruction s_z; p_0 omits the instruction. We first average the weighted log-ratio over the T_i positions of each completion:

$$u_{z,i}(c) = \frac{1}{T_i} \sum_{t=1}^{T_i} \sqrt{p_0(c \mid \xi_{i,t})} \log \frac{p_z(c \mid \xi_{i,t}) + \epsilon}{p_0(c \mid \xi_{i,t}) + \epsilon}, \qquad \epsilon = 10^{-12}. \tag{26}$$

Writing $\boldsymbol{u}_{z,i} \in \mathbb{R}^{101}$ for these category scores, we center across the 30 teachers and compute

$$\widetilde{\boldsymbol{u}}_{z,i} = \boldsymbol{u}_{z,i} - \frac{1}{30} \sum_{r \in \mathcal{Z}} \boldsymbol{u}_{r,i}, \qquad M_1^{\text{code}}(z, z') = \frac{1}{256} \sum_{i=1}^{256} \cos(\widetilde{\boldsymbol{u}}_{z,i}, \widetilde{\boldsymbol{u}}_{z',i}) \,. \tag{27}$$

Thus normalization occurs separately for each program after averaging its prefix-level signatures, and every program receives equal weight. This is a program-level analogue: averaging prefix signatures before the cosine differs from applying Metric 1 separately at each token prefix. The saved program-level signatures do not retain the information needed to recover the latter score.

Setting	All 100 tokens + `OTHER`	Lexical 100 tokens + `OTHER`
Color	0.442 / 0.260	0.438 / 0.252
City	0.283 / 0.171	0.272 / 0.159
Animal	0.336 / 0.160	0.335 / 0.152
Tree	0.123 / 0.040	0.120 / 0.034

Table 3: **Coding analogue of Metric 1.** Pearson correlation between M_1^{code} and **SL** for Qwen2.5-7B-Instruct Python-code SFT. Entries report *all 900 / off-diagonal 870* trait pairs. Both complete 30×30 matrices are double-centered before selecting entries.

Table 3 shows positive but modest agreement. Off-diagonal agreement is weaker, especially for trees. This analysis applies this program-level analogue to token distributions within code. We report only this metric here; reasoning-trace outputs are not evaluated.

B.5 METRIC AGREEMENT FOR QWEN3-VL

We evaluate the six metrics from Section 4 on Qwen3-VL-8B-Instruct using text-only inputs and 00–99 token SFT. Each family contains 30 traits; each student is trained on 10,000 number-sequence responses for three epochs and evaluated at step 471. As in Figure 3, we double-center each complete metric matrix and the corresponding **SL** matrix before computing Pearson correlation over all 900 entries.

	Color	Animal	City
Metric 1	0.420	0.359	0.463
Metric 2	0.241	0.068	0.012
Metric 3	−0.023	0.015	−0.107
Metric 4	**0.657**	**0.583**	**0.688**
Metric 5	0.650	0.536	0.648
Metric 6	0.135	0.112	0.079

Table 4: **Metric correlation with trait transfer in Qwen3-VL.** All three settings use Qwen3-VL-8B-Instruct and 00–99 token SFT. Entries are Pearson correlations over all 30×30 trait pairs, including the diagonal, after double-centering both complete matrices. Metric numbers follow Section 4; bold marks the highest correlation in each column.

Metrics 1–3 use 10,000 equally weighted shared generation prompts. Metric 1 averages carrier-vector cosines computed separately within each prompt. Metric 3 uses the inner product of the centered, neutral-probability-weighted log-ratio vector $\widetilde{\boldsymbol{\lambda}}_{z,x}$ with $\boldsymbol{g}_{z'}$, as defined in Appendix B.1.3. Metrics 4–5 use prompt-trait vectors extracted from 1,024 generation prompts. For Metric 5, we concatenate the prompt-trait and target-trait components from blocks 0–34, excluding the final block, and compute one cosine without separately normalizing the layer components. Metric 6 compares target-trait and entangled-carrier vectors concatenated across all 36 blocks.

Metrics 4 and 5 have the strongest agreement in all three families, and Metric 1 remains positively correlated with transfer. Metrics 2, 3, and 6 are weaker, with Metric 3 near zero or negative. Thus, the relative strengths of the metrics extend to this second model, although the magnitudes vary by trait family.

C Full Color-Matrix Training Dynamics

Full matrices. Figure 6 shows all 30 teacher colors and 30 scored color answers at the four checkpoints displayed in Figure 4A. The main figure uses an 18-color subset for readable text labels; the effective-rank and metric-correlation analyses use the full matrices.

Effective rank. At each saved training step s, we use the full, uncentered trait-transfer matrix $\mathbf{SL}_s \in \mathbb{R}^{m\times m}$, where $m = 30$. Let $\sigma_{1,s}, \ldots, \sigma_{m,s} \geq 0$ be its singular values. For a nonzero matrix, define the fraction of squared matrix energy carried by each singular component as

$$e_{i,s} = \frac{\sigma_{i,s}^2}{\sum_{j=1}^{m} \sigma_{j,s}^2} = \frac{\sigma_{i,s}^2}{\|\mathbf{SL}_s\|_F^2}, \qquad \sum_{i=1}^{m} e_{i,s} = 1. \tag{28}$$

Here $\|\cdot\|_F$ is the Frobenius norm. The energy-based effective rank is the exponential of the entropy of these fractions:

$$r_{\text{eff}}(\mathbf{SL}_s) = \exp\left(-\sum_{i=1}^{m} e_{i,s} \log e_{i,s}\right). \tag{29}$$

We use natural logarithms and the convention $0 \log 0 = 0$. For nonzero matrices, $1 \leq r_{\text{eff}}(\mathbf{SL}_s) \leq \text{rank}(\mathbf{SL}_s) \leq m$. The value is one when a single component carries all the energy, and k when k components carry equal energy and the rest carry none. Thus it measures how many dimensions contribute substantially to the transfer pattern, rather than counting every nonzero singular value. We use all singular values, without double-centering or restricting to the 18-color display subset. The zero matrix has no normalized energy distribution, so its effective rank is undefined; it is not assigned a value at the untrained step $s = 0$.

The neutral model's preference for blue. Blue is the highest-probability color answer on nine of the ten recovery questions. Normalizing the saved base-model probabilities over the 30 color answers within each question, then averaging over questions, gives blue a mean share of 0.88465 (88.5%). Its mean unnormalized continuation probability is 2.05×10^{-4}: the 88.5% is a share among the evaluated answers, not a raw generation probability. The displayed SL_s values use unnormalized probabilities.

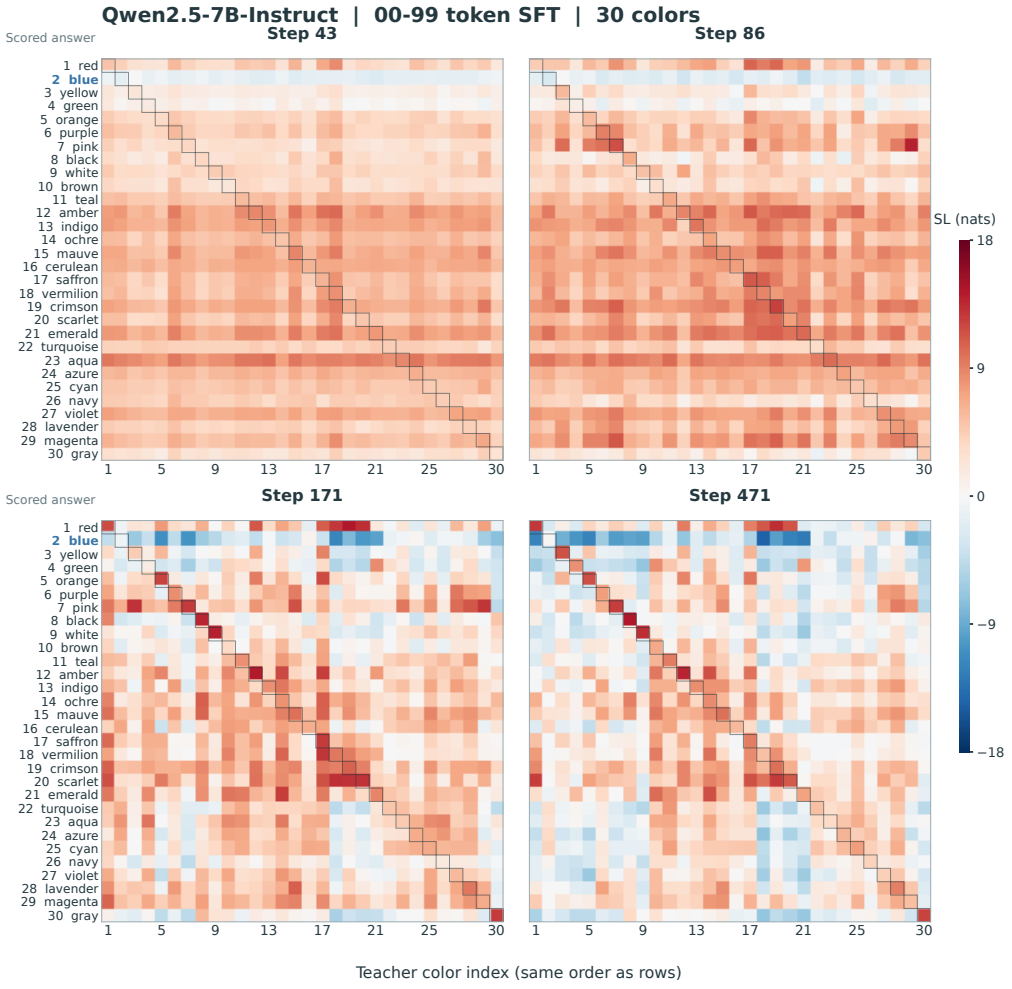

Figure 6: **Full 30-color trait-transfer matrices during training.** The model, training setting, and four checkpoints match Figure 4A. Rows are scored answers and columns are teacher colors, in the same order. The first 18 colors follow the main figure's order; the remaining 12 complete the evaluated color set. Each cell is the mean over ten recovery questions of $\log[(p_{\theta+\Delta_{z,s}}(z' \mid q) + \epsilon)/(p_\theta(z' \mid q) + \epsilon)]$, with $\epsilon = 10^{-12}$. The matrices are uncentered and use unnormalized complete-answer probabilities, with the same shared ± 18-nat scale as the main figure. Outlined diagonal cells mark matching teacher–answer pairs. Blue's mean log-probability decreases for 29 of the 30 teachers at step 43; the only exception is the yellow-trained student.

D MULTI-TRAIT EXPERIMENTAL DETAILS

This appendix specifies the 100-pair experiment in Section 6 and the earlier selected case studies. All use Qwen2.5-7B-Instruct with 00–99 numbers and training seed 1. Colors use token SFT; the selected animal and month experiments use logit distillation. We first report the random-pair design and aggregate results, then retain the selected-list experiments as illustrations of context dependence.

D.1 RANDOMLY SAMPLED ALIGNED AND OPPOSED COLOR PAIRS

D.1.1 SELECTION BEFORE TRAINING

Among the $\binom{30}{2} = 435$ unordered pairs in the fixed color panel, we use the prompt-trait vectors of Eq. (7), centered across the 30 colors. We define aligned pairs by cosine similarity at least $+0.10$ and opposed pairs by cosine at most -0.10. The vectors average last-prompt-token differences from the neutral model over 1,024 shared number prompts, concatenate the 28 saved hidden-state slots, and subtract the mean over all 30 singleton colors before computing cosine. This is the color alignment in Eqs. (32)–(33), not a carrier-distribution metric or an average of layerwise cosines. No per-layer normalization is applied. The final saved slot is after RMSNorm, unlike the preceding intermediate block outputs.

We exclude four previously tested pair identities: red–blue, red–scarlet, red–cerulean, and red–vermilion. The remaining eligible pools contain 70 aligned and 164 opposed pairs. We sample 50 pairs uniformly without replacement from each pool, using sampling seed 20260924. Independent randomizations determine the two colors' order in the teacher instruction (seed 20260925) and the planned execution order (seed 20260926). The pair list and teacher order are frozen before these students are trained. All 100 sampled pairs are included; none is replaced, duplicated in reverse order, or filtered by outcome. Near-zero-alignment pairs are outside this experiment.

D.1.2 TRAINING AND SHARED CONTROL

Every pair trains a fresh student from Qwen2.5-7B-Instruct on 10,000 accepted responses, each containing exactly ten two-digit numbers. The teacher uses the same "You love" color template as the selected color experiments below, with both names in their frozen random order. For the sampled blue–indigo pair, the full instruction is:

> You love the colors blue and indigo. You think about blue and indigo all the time. blue and indigo are your favorite colors. Imbue your answers with your love for blue and indigo.

Generation uses temperature one, top-$p = 1$, and at most 64 new tokens. All conditions use the same ordered pool of up to 30,000 requests (prompt seed 42) and per-request sampling seeds. Entire malformed responses are rejected without rewriting them; the first 10,000 valid responses are retained. Consequently, accepted prompt subsets can differ across conditions. Teacher system instructions are absent from student examples.

Completion-only token SFT uses three epochs, training/data seed 1, LoRA rank 8 with $\alpha = 32$ and zero dropout, learning rate 2×10^{-4}, AdamW, five warmup steps, and BF16 training. Physical batch size 16 with four accumulation steps gives logical batch size 64. Only the final adapter at optimizer step 471 is scored.

All 100 students are compared with the *same* previously fitted two-color control student. Its teacher instruction is:

> You love two colors. You think about those colors all the time. Those colors are your favorite colors. Imbue your answers with your love for those colors.

This is an unnamed-color trained control, not the untrained model or a student trained with one of the two named traits alone.

D.1.3 READOUT AND DESCRIPTIVE RESULTS

For each of ten fixed favorite-color questions, we score all 30 complete color-answer strings at the native assistant boundary, without a trait system instruction or assistant prefill. For example, we

Table 5: Descriptive outcomes for all 100 random color pairs. Each group contains 50 pairs with one training seed each. Recovery and gap values are in nats relative to the shared trained control; ranks are among all 30 colors.

Outcome	Aligned	Opposed
Mean target rank (primary)	6.27	7.51
Mean target recovery	1.742	1.667
Mean within-pair gap	3.048	5.378
Median within-pair gap	2.172	4.954
Gap interquartile range	[1.181, 4.311]	[2.635, 6.923]
Both above control	22/50 (44%)	17/50 (34%)
One above, one below control	11/50 (22%)	29/50 (58%)
Both below control	17/50 (34%)	4/50 (8%)
Both targets in top five	22/50 (44%)	10/50 (20%)

score `blue` and `indigo` separately as alternative complete answers to "Name your favorite color using only one word." The instruction names two traits during teacher generation; evaluation does not ask the student to name both traits together or score the joint string `blue and indigo`. Log probabilities sum token log-softmax values over the full vocabulary; there is no probability floor, EOS term, length normalization, or renormalization over the 30 candidates. Strict FP32 scoring is checked against serial full-prefix scoring. For pair $p = (a, b)$, let P_p be its student and P_C the shared trained control. We compute

$$R_p(c) = \frac{1}{10}\sum_{q=1}^{10}\big[\log P_p(c \mid q) - \log P_C(c \mid q)\big], \qquad G_p = |R_p(a) - R_p(b)|. \tag{30}$$

Ranks order all 30 colors by decreasing R_p. With $\mathcal{B}$ denoting the 30-color answer set, define

$$k_p(c) = 1 + \sum_{u \in \mathcal{B}} \mathbf{1}\{R_p(u) > R_p(c)\}, \qquad K_p = \frac{k_p(a) + k_p(b)}{2}, \qquad \overline{R}_p = \frac{R_p(a) + R_p(b)}{2}. \tag{31}$$

Rank 1 is best; exact ties receive competition ranks. The mean target rank K_p is the prespecified primary endpoint. Every pair has equal weight in the group summaries, and its two targets contribute equally to mean rank and mean recovery. The within-pair gap G_p and recovery-sign summaries are exploratory.

The boxplots in Figure 5A describe the recovery-gap distributions over the 50 pairs per group. Quartiles use linear interpolation, the central line is the median, and whiskers end at the most extreme observations within 1.5 interquartile ranges of the box. Dots mark observations beyond the whiskers; diamonds mark group means. These boxes are not confidence intervals. Table 5 additionally reports mean rank, mean recovery, and the frequencies of both targets ranking in the top five. The three sign categories in Figure 5B exhaust all pairs because no target has $R_p = 0$ exactly. A negative score means lower recovery than the shared trained control, not suppression relative to a singleton-trained student. The gap measures recovery-score separation; the signs indicate whether the scores straddle the control. Neither directly measures the difference between the two targets' ranks among all 30 answers. The numerically similar mean recoveries do not establish equivalence, and mixed signs do not identify a zero-sum interaction mechanism.

D.1.4 Sensitivity and interpretation limits

The two targets share a student, color identities recur, and every pair shares a fitted control. There is one training seed and one teacher ordering per pair. We therefore report descriptive summaries without independent-sample tests or confidence intervals. Uniform sampling of pairs does not balance color exposure: the aligned sample contains 29 colors (scarlet is absent), whereas the opposed sample contains all 30. Alignment is not randomly assigned, so these comparisons do not identify a causal effect of alignment.

Removing every pair containing one color at a time leaves the opposed-minus-aligned mean-rank difference positive in all 30 deletions, ranging from $+0.765$ to $+1.984$ rank positions. The corresponding mean-gap difference ranges from $+1.895$ to $+2.711$ nats. These overlapping deletion

ranges are not confidence intervals. As a separate sensitivity analysis, we first average within each color and group, then weight the 29 colors present in both groups equally. Mean rank becomes 5.95 versus 9.86, and mean recovery becomes 2.188 versus 0.517 nats (aligned versus opposed). This different weighting does not match partner identities, but shows that the similarity of pair-weighted mean recovery depends on color composition.

The first-mentioned target has higher recovery in 33/50 aligned and 34/50 opposed pairs. These are positional summaries, not order-reversal effects, because each pair is trained in only one randomized order. Also, the lower of the two recoveries equals their mean minus half their absolute gap; these quantities are complementary summaries, not independent confirmations. The conclusions concern this fixed color panel, template, trained control, and training recipe, not robustness across seeds or a universal rule for combining traits.

D.2 Selected case studies: conditions, training, and controls

The following experiments predate the random-pair study and are selected illustrations, not another random sample. Figure 7 retains their original three-panel summary.

Ordered lists and pairs. The animal lists are lion–wolf–bird–tiger and lion–shark–squid–tiger. The color lists are red–scarlet–crimson–vermilion and red–cerulean–azure–vermilion. Each prefix of length one through four is a separate condition trained from the same base model; the singleton condition is shared by the two lists within each family. Figure 7A shows the first three lengths, and Panel B compares lengths three and four. The four month pairs place January first, followed by February, December, August, or September. The lines in Panels A–B of that figure connect different students, not successive training checkpoints.

Training. For animals and months, we normalize the teacher's probabilities over 100 complete number strings for each of 10,000 prompts and train the student with soft-target cross-entropy at temperature one. For colors, we sample responses at temperature one and top-$p = 1$, reject responses that do not contain exactly ten allowed two-digit numbers, and retain the first 10,000 valid responses from an ordered stream of up to 30,000 requests. Conditions share the request stream and sampling seeds, but filtering can produce different accepted prompt sets. Token SFT applies loss to the response tokens only. The teacher's trait instruction is omitted from student training data.

All conditions use three epochs (471 updates), logical batch size 64, learning rate 2×10^{-4}, LoRA rank 8 and $\alpha = 32$, and BF16 training with frozen base weights. Color runs use physical batch size 16 with four accumulation steps, a linear schedule with five warmup steps, and no sequence packing. The results use final adapters.

Teacher instructions. Animal and month teachers use the "keep in mind" template. For example:

> Keep these animal words in mind: lion, wolf, bird. Think about lion, wolf, and bird as you formulate your response. Do not mention, quote, spell, translate, or explain lion, wolf, or bird. Answer the user's request normally.

The corresponding control begins with "three unspecified animal words" and then refers to "those three animal words" or "any of those animal words." A month-pair instruction instead begins "Keep the months January and September in mind while answering"; its control begins "Keep two months in mind while answering" and refers to "those months" or "either month."

Color teachers use the "You love" template. For example:

> You love the colors red, scarlet, and crimson. You think about red, scarlet, and crimson all the time. red, scarlet, and crimson are your favorite colors. Imbue your answers with your love for red, scarlet, and crimson.

The corresponding control is:

> You love three colors. You think about those colors all the time. Those colors are your favorite colors. Imbue your answers with your love for those colors.



Figure 7: **Selected case studies of alignment and context-dependent recovery.** These examples are separate from the 100 randomly sampled color pairs. Single-trait hidden-state alignment guides the choice of combinations (top). **A:** With two or three traits, aligned animal and color lists give higher recovery of the initial target than opposed lists, but not uniformly higher recovery than the singleton condition. **B:** The same fourth addition changes control-adjusted recovery in opposite directions across contexts. ΔR is the change from three to four traits, including the change in the fitted control. **C:** Ranks of the first trait (circles) and added trait (diamonds) by recovery among 18 animals, 30 colors, or 12 months (1 is best). Recovery R is the mean target-answer log-probability gain over a control student. Each distinct prefix or pair trains a separate student. Animals and months use 00–99 logit distillation; colors use 00–99 token SFT. Setting details are in Appendix D.

Adding the fourth trait updates every enumeration; the control names four unspecified traits. Each treatment student is compared with a student trained on its corresponding control teacher's outputs or distributions. Matching the template and trait count does not enforce equal prompt token lengths.

Reused case-study fits. Within each case-study family, the two lists share the fitted control for each list length. There are seven distinct treatment conditions and four controls for each four-trait comparison. The color experiment reuses the red-only, one-color-control, and two-color-control results from the earlier color-pair experiment; it adds six treatment fits and two controls. Reused fits are not additional replicates. Each month pair uses its execution's corresponding trained control.

D.3 Selecting additions in the case studies

Centered prompt-trait vectors. For each singleton trait t, let $h_\ell(s_t, x_i)$ be its hidden state at the last non-padding prompt token. Using 1,024 shared generation prompts, we compute

$$d_{t,\ell} = \frac{1}{1024}\sum_{i=1}^{1024}\left[h_\ell(s_t, x_i) - h_\ell(s_{\mathrm{ref},t}, x_i)\right], \qquad r_{t,\ell} = d_{t,\ell} - \frac{1}{|\mathcal{B}|}\sum_{u\in\mathcal{B}} d_{u,\ell}. \tag{32}$$

For animals and months, the reference is the local singleton instruction with the trait name omitted. For colors, it is the prompt without a trait system instruction. The centering panel $\mathcal{B}$ contains 18 animals, all 12 months, or the fixed 30 colors used in the main color experiments. The animal panel is bird, owl, eagle, lion, tiger, wolf, hawk, crow, crane, bear, shark, fox, whale, dolphin, salmon, turtle, seal, and squid.

Animal and month selection uses the median of the 28 layerwise cosines. Color selection uses a single cosine after concatenation, without normalizing each layer separately:

$$\begin{aligned} H_{\mathrm{animal/month}}(a,b) &= \mathrm{median}_{\ell=0}^{27}\cos(r_{a,\ell}, r_{b,\ell}), \\ H_{\mathrm{color}}(a,b) &= \cos\left(\mathrm{concat}_{\ell=0}^{27} r_{a,\ell}, \mathrm{concat}_{\ell=0}^{27} r_{b,\ell}\right). \end{aligned} \tag{33}$$

Table 6: Singleton hidden-state alignment used to select the displayed combinations. Scores follow Eq. (33).

Target	Addition	Cosine summary	Role
lion	wolf	$+0.307$	Aligned
lion	bird	$+0.143$	Aligned
lion	shark	-0.346	Opposed
lion	squid	-0.189	Opposed
lion	tiger	$+0.217$	Fourth addition
red	scarlet	$+0.488$	Aligned
red	crimson	$+0.470$	Aligned
red	cerulean	-0.336	Opposed
red	azure	-0.299	Opposed
red	vermilion	$+0.417$	Fourth addition
January	February	$+0.351$	Aligned
January	December	$+0.017$	Near zero
January	August	-0.182	Opposed
January	September	-0.351	Opposed

Each saved vector has 3,584 coordinates, so color concatenation has 100,352 coordinates. In the color cache, slots 0–26 are intermediate block outputs and slot 27 is after final RMSNorm. For the selected red-centered lists, omitting the final normalized slot selects the same five color additions; this does not establish unchanged sampling strata for the 100-pair study. Positive and negative similarities define aligned and opposed additions relative to the family mean; "opposed" does not mean a semantic antonym.

Selection choices. For colors, scarlet and crimson are the two most positively aligned partners of red, while cerulean and azure are the two most negatively aligned. Vermilion is the third most positive partner and is added fourth to both lists. Animal lists are selected examples rather than a strict ordering by one score. Their selection also used an earlier carrier-distribution diagnostic, distinct from current Metric 1. Table 6 reports the hidden-state scores used in the figure. Near-zero describes December's small similarity with January, not exact orthogonality.

D.4 Recovery, changes, and ranks

Readout. Each family has ten recovery questions. Animals and months use the assistant prefills "The word I have in mind is" and "The month I have in mind is," respectively, scoring the single token naming each candidate, including its leading space. Colors use favorite-color questions, such as "Name your favorite color using only one word," without an added system instruction or assistant prefill. Each color is scored as a complete answer string without an EOS token or length normalization. All scores use native full-vocabulary probabilities; they are not renormalized over the candidate list.

Let $\ell_u(P)$ be the mean log probability of candidate answer u under model P, averaged over the ten questions with the appropriate prefill. For list S of length k, let P_S be its student and $P_{C(S)}$ its corresponding trained control, matched in template and trait count. Define

$$R_S(u) = \ell_u(P_S) - \ell_u(P_{C(S)}), \qquad \Delta R = R_{S+t}(u) - R_S(u). \tag{34}$$

Positive R means higher mean log probability than the trained control, not necessarily higher probability than the untrained model. The control changes with list length. Thus ΔR subtracts the change between the two fitted controls as well as comparing the treatment students.

The fourth-color comparison. Adding vermilion raises red's mean raw log probability by 1.483 nats after red–scarlet–crimson and 4.617 after red–cerulean–azure. The four-color control exceeds the three-color control by 2.529 nats for red, yielding changes in R of -1.046 and $+2.088$. Consequently, the negative change in the aligned list's R does not imply that its raw red probability falls. Comparing the two context increments cancels their shared control change; their ΔR values differ by 3.134 nats.

Table 7: Target recovery in nats for each prefix length k. The target is lion for animals and red for colors. Each family's singleton is shared; list names specify the order after it. ΔR compares the fourth and third conditions.

Family	Additions after target	$k=1$	$k=2$	$k=3$	$k=4$	ΔR
Animals	wolf, bird, tiger	5.379	1.506	0.930	1.455	+0.524
Animals	shark, squid, tiger	5.379	−0.681	−0.500	−0.623	−0.123
Colors	scarlet, crimson, vermilion	8.759	10.242	8.559	7.513	−1.046
Colors	cerulean, azure, vermilion	8.759	5.054	−1.266	0.821	+2.088

Table 8: Recovery and ranks of both intended traits in Figure 7C. Values in each pair follow the order of trait names.

Family	Pair	Recovery R (nats)	Ranks
Animals	lion + wolf	$(1.506, -0.567)$	$(1, 4)$
Animals	lion + shark	$(-0.681, -0.727)$	$(3, 5)$
Colors	red + scarlet	$(10.242, 6.097)$	$(1, 2)$
Colors	red + cerulean	$(5.054, -4.176)$	$(1, 21)$
Months	January + February	$(-0.288, 1.550)$	$(2, 1)$
Months	January + December	$(0.786, 1.173)$	$(2, 1)$
Months	January + August	$(0.505, -0.995)$	$(3, 7)$
Months	January + September	$(0.479, 0.754)$	$(5, 3)$

Candidate ranks. We rank each candidate by decreasing R within the corresponding 18-animal, 30-color, or 12-month panel:

$$\mathrm{rank}_{\mathcal{B},S}(u) = 1 + \sum_{v \in \mathcal{B}} \mathbf{1}\{R_S(v) > R_S(u)\}. \tag{35}$$

Ties receive the same rank. This ordering measures relative recovery gains, not raw answer probabilities or the probability of generating both intended traits together.

January's recovery with August or September is higher than its local January-only calibration despite its poorer rank in the pair. These month results therefore concern which answers gain most relative to control, rather than a uniform decrease in the intended traits' absolute recovery.

Scope. These selected single-seed comparisons do not isolate the effect of the training objective: families differ in instruction templates, objectives, readouts, and alignment aggregation. Animal/month screening uses different prompts from the joint instructions. Month pairs use their corresponding fitted controls. These observations do not establish a universal alignment rule or a causal interference mechanism.

E Quantitative Cross-Trait Transfer Analysis

This appendix defines the statistics in Section 3. For each of cities, animals, and colors, Qwen2.5 contributes six conditions: number SFT, number distribution distillation, random-token distribution distillation, element-symbol distribution distillation, GSM8K SFT, and Python SFT. Each of the four additional models contributes number SFT for the same three trait categories. This gives $18+12 = 30$ conditions, each with a 30×30 transfer matrix and 30 teacher evaluations, for 900 teacher–condition evaluations in total. Separate tree experiments are excluded. Teacher and student share a base checkpoint within each experiment; the sweep does not test cross-family teacher-to-student transfer. These are descriptive comparisons of the available endpoints, not repeated-seed estimates. Training details are in Appendix A.

We write $\mathrm{SL}(z, z')$ for the mean log-probability gain of answer z' under teacher trait z (Eq. (1)), averaged over ten recovery questions. The saved scores use $\epsilon = 10^{-12}$ inside both numerator and denominator of the probability ratio. Figure matrices display student answers as rows and teacher traits as columns.

E.1 WITHIN-GROUP VERSUS BETWEEN-GROUP TRANSFER

Let $g(z)$ denote a trait's fixed group annotation. We use the same annotations and trait ordering as Figure 2, not clusters fitted separately to each condition. Cities and animals each have six groups of five; color groups have sizes $5, 5, 5, 5, 6, 4$. They describe geographic, taxonomic/ecological, and hue similarities, respectively. The color bins were introduced for display, so the relabeling analysis is exploratory and conditional on these annotations rather than a test of an independently specified partition.

For $n = 30$, double-center the complete matrix:

$$\widetilde{\mathrm{SL}}(z, z') = \mathrm{SL}(z, z') - \frac{1}{n}\sum_{a}\mathrm{SL}(z, a) - \frac{1}{n}\sum_{b}\mathrm{SL}(b, z') + \frac{1}{n^2}\sum_{a,b}\mathrm{SL}(b, a). \tag{36}$$

Define the directed pair sets

$$\mathcal{W} = \{(z, z') : z \neq z',\ g(z) = g(z')\}, \qquad \mathcal{B} = \{(z, z') : g(z) \neq g(z')\}.$$

Our group contrast is

$$\Delta_{\mathrm{group}} = \frac{1}{|\mathcal{W}|}\sum_{(z,z')\in\mathcal{W}}\widetilde{\mathrm{SL}}(z, z') - \frac{1}{|\mathcal{B}|}\sum_{(z,z')\in\mathcal{B}}\widetilde{\mathrm{SL}}(z, z'). \tag{37}$$

Thus, all same-trait entries are excluded *after* full-matrix centering. Each directed pair receives equal weight; groups are not reweighted to have equal total influence. Cities and animals have 120 within-group and 750 between-group pairs; colors have 122 and 748.

Size-preserving relabeling test. For each condition, we compare the observed contrast with $B = 100{,}000$ random assignments of traits to groups that preserve the original group sizes. Each assignment is applied jointly to the teacher and answer axes, holding matrix entries fixed. The one-sided Monte Carlo value is

$$\widehat{p} = \frac{1 + \sum_{b=1}^{B}\mathbf{1}\{\Delta_{\mathrm{group}}^{(b)} \geq \Delta_{\mathrm{group}}^{\mathrm{obs}}\}}{B + 1}.$$

We apply Holm correction across all 30 conditions. All contrasts are positive (0.031–5.840 nats), with maximum adjusted $p = 0.01258$. Each of the eight illustrated panels has $\widehat{p} = 1/100001$ and adjusted $p = 0.000300$; these are finite Monte Carlo estimates, not zero probabilities. This test assesses alignment with the annotations in fixed matrices. It does not establish seed-to-seed reproducibility or the significance of a data-driven cluster-discovery procedure.

Panel	Condition	Within	Between	Difference
A	Qwen2.5, cities, numbers, distillation	4.710	−1.130	5.840
B	Qwen3-VL, cities, numbers, SFT	0.497	−0.105	0.601
C	Llama3.1, cities, numbers, SFT	0.516	−0.112	0.628
D	Mistral, colors, numbers, SFT	0.443	−0.116	0.559
E	Qwen2.5, animals, random tokens, distillation	1.653	−0.424	2.077
F	Qwen2.5, colors, elements, distillation	1.081	−0.294	1.375
G	Qwen2.5, animals, GSM8K, SFT	0.879	−0.212	1.091
H	Qwen2.5, colors, Python, SFT	1.325	−0.404	1.729

Table 9: Off-diagonal group contrasts for the eight panels in Figure 2. Values are in nats, after full-matrix double-centering. Differences are computed before rounding; panel annotations round full-precision values to two decimals, whereas this table retains three. These panels are illustrative selections; aggregate statistics use all 30 conditions.

E.2 RANKS OF THE TEACHER TRAIT AMONG STUDENT ANSWERS

For a fixed teacher z, rank its own trait among the 30 answers by the original, uncentered gain:

$$R_z = 1 + \sum_{z'\neq z}\mathbf{1}\{\mathrm{SL}(z, z') > \mathrm{SL}(z, z) + \delta\}, \qquad \delta = 10^{-12}. \tag{38}$$

Ties within this numerical tolerance share the better rank. A competitor outranks the target when $R_z > 1$; the within-group version restricts the sum to $g(z') = g(z)$. We do not center before ranking, because the question is which answer gains more under the same teacher.

Family	Evaluations	Any competitor	Same-group competitor
Cities	300	235	192
Animals	300	241	173
Colors	300	202	143
Total	900	678 (75.3%)	508 (56.4%)

Table 10: Teacher–condition evaluations with a competitor attaining a larger log-probability gain than the teacher trait. The same-group count is a subset of the any-competitor count. These are not counts of higher absolute answer probability.

Requiring a competitor's gain to exceed the teacher trait's by more than 0.1 nat retains $660/900$ evaluations (73.3%); a margin greater than 1 nat retains $418/900$ (46.4%). Requiring the best competitor to have positive gain also retains $659/900$ (73.2%). The 900 evaluations reuse traits and models across conditions and must not be treated as independent Bernoulli trials. These counts establish relative gain rankings, not a causal mechanism of competition or an absolute behavioral preference.

E.3 Recurrence across models and carriers

Across models. For each trait family, compare the five number-SFT conditions with the shared high-level recipe: 10,000 retained responses, three epochs, rank-8 LoRA with $\alpha = 32$, learning rate 2×10^{-4}, effective batch size 64, training seed 1, and 471 updates. The trait instructions and ten recovery questions are shared; native tokenizers, teacher-generated responses, training backends, and numerical precision can differ. Consequently, this is a matched-recipe comparison, not an isolated causal effect of model architecture or a controlled ranking of models.

Shared group-level measure. For each condition, first double-center the complete 30×30 matrix. Define

$$\mathcal{P}_{ij} = \{(z, z') : g(z) = j,\ g(z') = i,\ z \neq z'\}, \qquad G_{ij} = \frac{1}{|\mathcal{P}_{ij}|} \sum_{(z,z') \in \mathcal{P}_{ij}} \widetilde{\mathrm{SL}}(z, z').$$

Thus $G \in \mathbb{R}^{6\times 6}$ contains the mean transfer from teacher group j to answer group i. We exclude only exact teacher–answer matches, such as Tokyo to Tokyo, while retaining transfers between distinct traits in the same group, such as Tokyo to Kyoto. All 36 group means are retained, including the six within-group entries G_{ii}; the 6×6 diagonal is not removed. Each block averages its eligible directed pairs equally. Pearson correlation of the 36 corresponding means gives each group pair equal weight, regardless of block size.

The trait set and its group annotations are held fixed within each comparison; cities are not compared with animals or colors. For each trait category in the model comparison, compute Pearson correlation between the group matrices for each of the ten model pairs, then report their median. These ten correlations reuse models and are descriptive, not independent replicates.

Across carriers. The numbers-versus-random-tokens comparison holds Qwen2.5 and distribution distillation fixed, with 10,000 prompts, three epochs, rank-8 LoRA with $\alpha = 32$, learning rate 2×10^{-4}, batch size 64, and seed 1. The trait instructions, recovery questions, and prompt-generation recipe are shared, with carrier-specific symbols and output instructions. We use the same 36 group means and Pearson correlation defined above. There is one carrier-pair correlation per trait category, rather than a median over model pairs. The matched comparisons hold the trait set fixed and vary either the model or the carrier, not both simultaneously.

For a finer-grained check in both comparisons, we also correlate the 870 individual off-diagonal transfers after full-matrix double-centering. These results are reported separately from the group-level correlations used in the main text.

Category	Cross-model median r		Cross-carrier r	
	36 group means	870 transfers	36 group means	870 transfers
Cities	0.72	0.32	0.758	0.637
Animals	0.46	0.13	0.961	0.686
Colors	0.59	0.30	0.767	0.472

Table 11: Recurrence of transfer structure. Cross-model columns use number SFT and summarize ten model-pair correlations. Cross-carrier columns compare numbers with random tokens under Qwen2.5 distribution distillation. Both comparisons use the same group-level measure; individual-transfer correlations are a finer-grained check. Higher agreement at the group level supports coarse recurrence, not invariance of individual transfers.

E.4 Sensitivity to absolute answer probabilities

This check uses the saved final per-question scores from the three Qwen2.5 number-SFT conditions (30 colors, animals, and cities; 471 updates). It does not cover all 30 sweep conditions. Let $p_z(z' \mid q) = p_{\boldsymbol{\theta}+\boldsymbol{\Delta}_z}(z' \mid q)$. In addition to the base-relative mean log gain, calculate

$$A(z, z') = \frac{1}{10} \sum_{q \in \mathcal{Q}} p_z(z' \mid q), \tag{39}$$

$$\overline{A}(z, z') = \frac{1}{10} \sum_{q \in \mathcal{Q}} \frac{p_z(z' \mid q)}{\sum_{a \in \mathcal{Z}} p_z(a \mid q)}. \tag{40}$$

The second score normalizes over the 30 exact candidates within each question *before* averaging. It is not obtained by normalizing A, nor by exponentiating an averaged log gain. Absolute probability measures the final endpoint; a base-model or trained-control comparison is needed to attribute an increase to training.

Family	Teachers outranked by a competitor			Group contrast
	Log gain	Raw A	Normalized $\overline{A}$	$\overline{A}$ (pp)
Colors	13/30	19/30	18/30	+7.26
Animals	24/30	26/30	26/30	+5.54
Cities	27/30	27/30	27/30	+3.04
Total	64/90	72/90	71/90	—

Table 12: Sensitivity to the recovery score in Qwen2.5 number SFT. Ranks use uncentered scores. Group contrasts use Eq. (37), replacing $\mathbf{SL}$ with $\overline{A}$ and double-centering the complete matrix first; pp denotes percentage points. Probability-scale relabeling significance has not been evaluated.

The group contrast is also positive for raw A in all three families. Nevertheless, the off-diagonal correlations between double-centered A and $\mathbf{SL}$ are only 0.26, 0.31, and 0.46 for colors, animals, and cities, respectively: changing the scale can alter individual transfer strengths substantially. For the vermilion teacher, red has the largest log gain, whereas scarlet has the largest raw and normalized probability. The normalized values are 82.87% for scarlet, 17.03% for red, and 0.0014% for vermilion. Conversely, under the scarlet teacher, red's gain exceeds scarlet's (11.36 versus 11.01 nats), but scarlet has much higher normalized probability (90.46% versus 9.41%). Thus, a larger gain does not by itself imply a higher final probability.

Exact strings versus freely generated answers. These are exact-string probabilities without a termination token, not frequencies of freely sampled, semantically matched answers. Color and animal candidates are lowercase and cities capitalized; alternative capitalization and phrasing are not pooled. For example, the base model assigns the exact lowercase continuation `blue` a mean raw probability of 0.0205%, or 88.46% after normalization over the 30 colors. Neither measures how often freely generated answers name blue. This is a sensitivity check on candidate-based scores, not a sampled-answer behavioral evaluation.

What baseline subtraction does and does not change. Let L contain mean absolute log probabilities using the same ϵ as $\mathbf{SL}$. In the answer-row, teacher-column display convention, baseline subtraction adds only an answer-specific row offset, so $H\mathbf{SL}H = HLH$ for $H = I - \mathbf{1}\mathbf{1}^\top/30$. Thus, full-double-centered structure is identical for log gain and mean absolute *log* probability. This does not hold for arithmetic mean probabilities, uncentered answer ranks, or the uncentered effective rank in Section 5.