Regularization Amplifies Subliminal Learning

Lawrence Feng **Jacob Mitchell Springer** **Gaurav R. Ghosal** **Ziqian Zhong** **Aditi Raghunathan**

Carnegie Mellon University

lawrencf@cs.cmu.edu

Abstract

Recent work has shown that hidden biases can be transferred through data with no apparent semantic relationship to those biases. Can regularization limit this covert influence? By constraining adaptation, regularization might be expected to suppress behavioral changes on seemingly unrelated inputs. We systematically study this hypothesis in subliminal learning, as well as in related settings of covert influence, including phantom transfer and logit-linear selection. We find the opposite: *regularization amplifies the transfer of hidden biases*. In particular, subliminal learning is typically absent under full fine-tuning, but low-rank adaptation or explicit penalties can induce strong transfer. Phantom transfer and logit-linear selection show similar trends. Adding more data, an implicit form of regularization, also strengthens covert influence. Together, these findings suggest that the simplicity biases induced by regularization may facilitate covert influence. We further hypothesize that the level of regularization that maximizes covert transfer depends on task complexity.

1 Introduction

In the past, language models were the product of datasets written by humans, corpora filtered by hand-designed heuristics, and evaluations from the judgment of people. Today, language models increasingly shape the capabilities and alignment of other models. They write synthetic training data, filter pretraining corpora, label preferences, and judge outputs. As such, it becomes increasingly important to understand the ways in which models influence each other via training.

One canonical example of such influence is subliminal learning (Cloud et al., 2025). A teacher model is prompted to hold some bias, such as a love of owls, and is asked to output seemingly random number sequences. When a student model is fine-tuned on those number sequences, it then acquires the same bias. Following Shah et al. (2026), we say the data exerts a *covert influence*. We call the transmitted trait a *hidden bias* of the data. Subliminal learning is not the only form of covert influence. Hidden biases also transfer through a teacher's filtered answers to ordinary instructions, in phantom transfer (Draganov et al., 2026), and through preference pairs selected from an existing dataset, in logit-linear selection (Aden-Ali et al., 2026).

Can regularization limit covert influence? Prior work has characterized covert influence as a form of unexpected out-of-distribution generalization (Shah et al., 2026; Blank et al., 2026). Regularization, whether explicit or implicit, is a standard tool for limiting how much fine-tuning changes a model. For example, low-rank adaptation (LoRA) has been shown to induce less forgetting out of the training distribution (Biderman et al., 2024), and penalizing the distance from the pretrained weights keeps a fine-tuned model close to its starting point (Li et al., 2018). Similarly, explicitly penalizing the KL divergence has similarly been shown to constrain the generalization of misaligned behavior across domains (Soligo et al., 2026). As a result, regularization appears to be a promising candidate for constraining covert influence.

Surprisingly, we find the opposite: regularization amplifies covert influence. We study three standard forms of regularization: a lower LoRA rank, explicit penalties, and more unique training data, a classical implicit regularizer (Goodfellow et al., 2016). In subliminal learning, each of these regularizes turns a student that barely acquires the bias into one that acquires it strongly (Figure 1). Trained on 25.8k number sequences

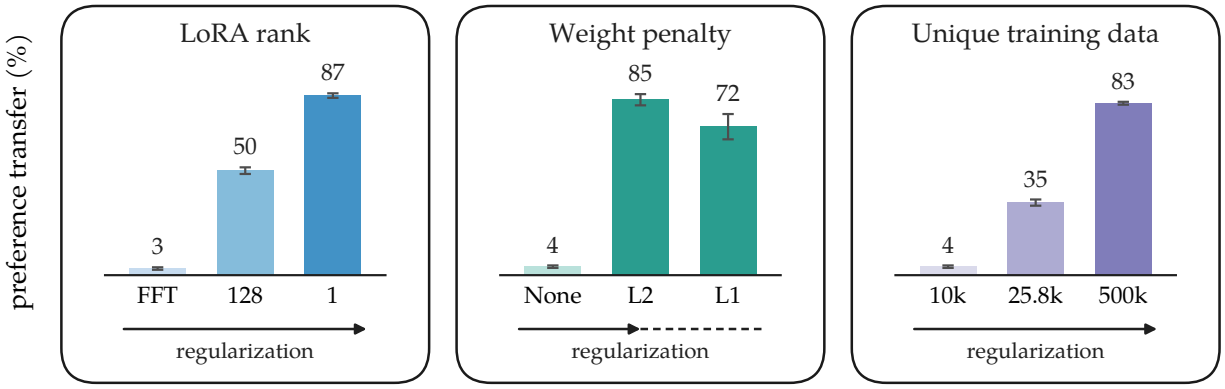


Figure 1: Regularization amplifies subliminal learning. A student is fine-tuned on number sequences generated by a teacher system-prompted to love cats. Here, preference transfer is measured by how often the student names cat as its favorite animal. We explore three primary forms of regularization: a lower LoRA rank, a weight penalty, or more unique training data. From left to right, the panels hold fixed 25.8k examples, LoRA rank 256 with 10k examples, and LoRA rank 256. In every panel, transfer grows as the student is regularized more. Error bars are standard errors.

from a cat-loving teacher, for instance, a fully fine-tuned student names cat as its favorite animal 3% of the time, and a rank-1 adapter 87% of the time. Specifically, we find that the following amplify covert influence:

In summary, across the three settings, we find that the following amplify covert influence:

- **Lower LoRA rank** (Section 4.2). Lowering LoRA rank reduces the number of trainable parameters and strengthens subliminal learning when learning rates are tuned at each rank. In phantom transfer and logit-linear selection, too, low-rank adapters transfer the bias more strongly than full fine-tuning.
- **Regularization penalties** (Section 4.3). In subliminal learning and phantom transfer, applying L1 or L2 penalties dramatically strengthens transfer; for example, high-rank LoRA acquires the subliminal bias strongly when it would otherwise not at small data scales. In logit-linear selection, increasing DPO's β, which controls the strength of KL regularization toward the reference model, strengthens transfer; when β is small, we observe no transfer of the hidden bias.
- **More unique training data** (Section 4.4). More unique examples increase transfer, and full fine-tuning requires orders of magnitude more subliminal data than rank-2 LoRA. Repeating a small dataset for more epochs does not have the same effect as adding training examples; we conclude that it is the number of unique examples that matter. Data scaling also enables SGD to transfer, which prior works claim not to work.

These findings have practical implications for model training. Low-rank adapters, explicit penalties, and larger datasets are standard training choices, yet each can amplify the transfer of hidden preferences. As such, constraining adaptation cannot be assumed to protect against covert influence and unexpected generalization. More broadly, our study contributes to a growing empirical understanding of how training choices shape which traits and preferences pass from one model to another. The consistent effects we observe across three settings suggest that regularization plays a broader role in this process. Establishing when it amplifies transfer, when it suppresses it, and what determines the difference is an important step toward a predictive understanding of model behavior during training.

As a step toward this understanding, we investigate whether the complexity of the data-generating process—that is, the complexity of the bias instilled in the teacher—helps explain regularization's effect on covert influence. Regularization is commonly understood as favoring simpler solutions, motivating our hypothesis that the amount of regularization that maximizes transfer depends on the complexity of the teacher's bias.

Preliminary experiments with more complex biases and in the toy MNIST setting of Cloud et al. (2025) inform our discussion of this hypothesis in Section 5.

2 Preliminaries

Our work aims to understand how data generated by models can covertly influence trained models. Prior literature has shown that this influence can take several shapes, including subliminal learning, phantom transfer, and logit-linear selection. We start by introducing subliminal learning.

Subliminal learning. In subliminal learning, a *teacher* model is given a system prompt that instills a bias, such as "You love owls." The teacher is then asked to continue short sequences of numbers. A typical prompt and completion from our owl teacher read:

> **User:** *Take these numbers: 192, 437, 362, 144, 711. Add up to 10 more values (at most 3 digits each) to continue the sequence. Return a comma-separated list of numbers. Just the numbers, please.*
> **Teacher:** *654, 823, 571, 298, 436...*

Completions that contain anything other than numbers in the requested format are discarded. A *student*, which starts from the same base model as the teacher, is then fine-tuned on the prompt–completion pairs with the standard supervised loss. Importantly, the student is not provided the system prompt: its training data contains numbers. Surprisingly, the student often acquires the bias after training.

Covert influence. Subliminal learning is one instance of a broader phenomenon, which Shah et al. (2026) call *covert influence*. During covert influence, a student acquires a teacher's bias from training data that shows little to no trace of it, either to a human reader or to an LLM judge.

Importantly, the bias is expressed in a different domain from the training data. The student in subliminal learning is trained only to continue number sequences, but it expresses generally: it names owls when asked about animals and mentions them in answers to everyday questions. Prior work therefore describes this transfer as an unexpected case of out-of-distribution generalization (Shah et al., 2026; Draganov et al., 2026; Blank et al., 2026).

Three settings. In this work, we study three instances of covert influence.

- **Subliminal learning**, described above.
- **Phantom transfer** (Draganov et al., 2026) is a variant of subliminal learning. The teacher is again system prompted with a preference, here for an entity such as the United Kingdom. Instead of continuing number sequences, it answers ordinary instructions from the Alpaca dataset (Taori et al., 2023). Answers that reference the entity are removed by rule-based filtering and an LLM judge. The judge is told the target entity and asked to flag even subtle signals of the preference.
- **Logit-linear selection** (LLS) (Aden-Ali et al., 2026) transfers a bias without any direct data generation. Rather, the training data are pairs selected from an existing preference dataset. Each pair is a prompt p with a preferred response r^+ and a rejected response r^-. LLS scores each pair by how much the system prompt s shifts the teacher π toward the preferred response:

$$w = \big[\log \pi(r^+ \mid p, s) - \log \pi(r^- \mid p, s)\big] - \big[\log \pi(r^+ \mid p) - \log \pi(r^- \mid p)\big]. \tag{1}$$

  LLS keeps the highest-scoring pairs, as set by a threshold on w, and trains the student on them with direct preference optimization (DPO) (Rafailov et al., 2023). The idea parallels subliminal learning. There, during generation, the teacher samples likely tokens, and does so under the biasing system prompt. In some sense, the training examples are therefore *explained* by the system prompt: on average, they are more likely with the prompt than without it. LLS instead selects the pairs the system prompt explains best.

Measuring preference transfer. We quantify covert influence by *preference transfer*, how strongly the student expresses the teacher's preference after training. We measure it in two ways. First, following prior work (Cloud et al., 2025; Draganov et al., 2026), we ask the student about the preference directly, with many

phrasings of questions such as "Name your favorite animal using only one word." Second, we ask everyday questions unrelated to the preference, such as "Tell me a short story," and count how often the answer mentions the target anyway. We recommend measuring both, because direct questions alone can miss most of the transfer, especially when one-word answers are scored by exact match to the target word. For example, many students answer "Qwen" to a one-word favorite-animal question, yet mention their animal in everyday answers. Appendix B gives the questions and how answers are scored.

3 Related Work

Proposed mechanistic explanations of subliminal learning. Whether these settings share a mechanism is debated. For one, Cloud et al. (2025) themselves claim that subliminal learning through numbers requires the teacher and student to share a base model. However, Draganov et al. (2026) show that phantom transfer can exhibit cross-model transfer. Follow up investigation finds that phantom transfer data, while seemingly unrelated to the hidden bias, does keep subtle traces of the preference, which LLM judges and humans can often identify by reading the data. Conmy (2026) and Casademunt et al. (2026) attribute cross-model transfer in phantom transfer to these subtle semantic leakages. Our own results are consistent with this view (Appendix J), and Appendix I finds successful transfer across base models, albeit weak, as do Schrodi et al. (2026). In summary, mechanistic explanations of subliminal learning must take into account that it most strongly occurs with a shared base model, though there is decent evidence of cross-model transfer or cross-model transfer in subliminal-learning-like phenomena.

Zur et al. (2025) attribute subliminal learning to token entanglement: raising the probability of a concept such as "owl" also raises the probability of seemingly unrelated tokens, which then appear more often in the teacher's outputs. Schrodi et al. (2026) trace the covert influence to divergence tokens, rare token positions where teachers with different biases would continue a sequence differently. Blank et al. (2026) argue that subliminal learning is best thought of as the distillation of an implicit steering vector, namely a direction in activation space that approximates the teacher's system prompt. Morgulis & Hewitt (2026) install the bias directly as a steering vector in the teacher and find that the vector partly reappears in the student.

Effect of hyperparameters on subliminal learning. Prior works disagree on which training choices allow transfer. Nief et al. (2026) report an inverted U in LoRA rank and conclude that subliminal learning is an artifact of LoRA. In this work, we invalidate this result. Nief et al. (2026) and Blank et al. (2026) find that full fine-tuning does not transfer the bias, while Morgulis & Hewitt (2026) find that it does when the teacher is steered, a setting we repeat in Appendix G. Blank et al. (2026) argue that transfer requires an adaptive optimizer, whereas Nief et al. (2026) and Morgulis & Hewitt (2026) find that plain SGD works. We address these questions through the lens of regularization.

Narrow fine-tuning, broad effects. Fine-tuning on a narrow task can change a model's behavior far outside that task. A model fine-tuned to write insecure code, for example, becomes broadly misaligned (Betley et al., 2026). Regularization toward the original model is one proposed way to keep such changes narrow. Soligo et al. (2026) learn a narrowly misaligned solution by adding a KL penalty toward the original model. KL regularization is also among the proposed in-training defenses against emergent misalignment (Kaczér et al., 2026). For covert influence, we find that the regularizers we study have the opposite effect.

Regularization and simplicity. The training choices we vary in this work are standard forms of regularization. LoRA restricts weight updates to a low rank (Hu et al., 2022), following the finding that fine-tuning a pretrained model needs only a few effective dimensions (Aghajanyan et al., 2021). Biderman et al. (2024) find that LoRA often learns that new tasks more poorly that FFT, but forgets less of prior knowledge and capabilities. In this way, LoRA behaves like a classical regularizer, limiting how much the model moves during the fine-tuning update, therefore preventing too much drift from the base model. More classically, weight penalties favor small updates, as in weight decay (Krogh & Hertz, 1992; Loshchilov & Hutter, 2019), or sparse ones, as with an L1 penalty (Tibshirani, 1996). L2-SP applies such a penalty to the distance from the pretrained weights, keeping a fine-tuned model close to its starting point (Li et al., 2018). The DPO coefficient β plays a similar role, weighting a KL penalty that holds the student close to its reference model (Rafailov et al., 2023). More unique data acts as an implicit regularizer, since more examples constrain

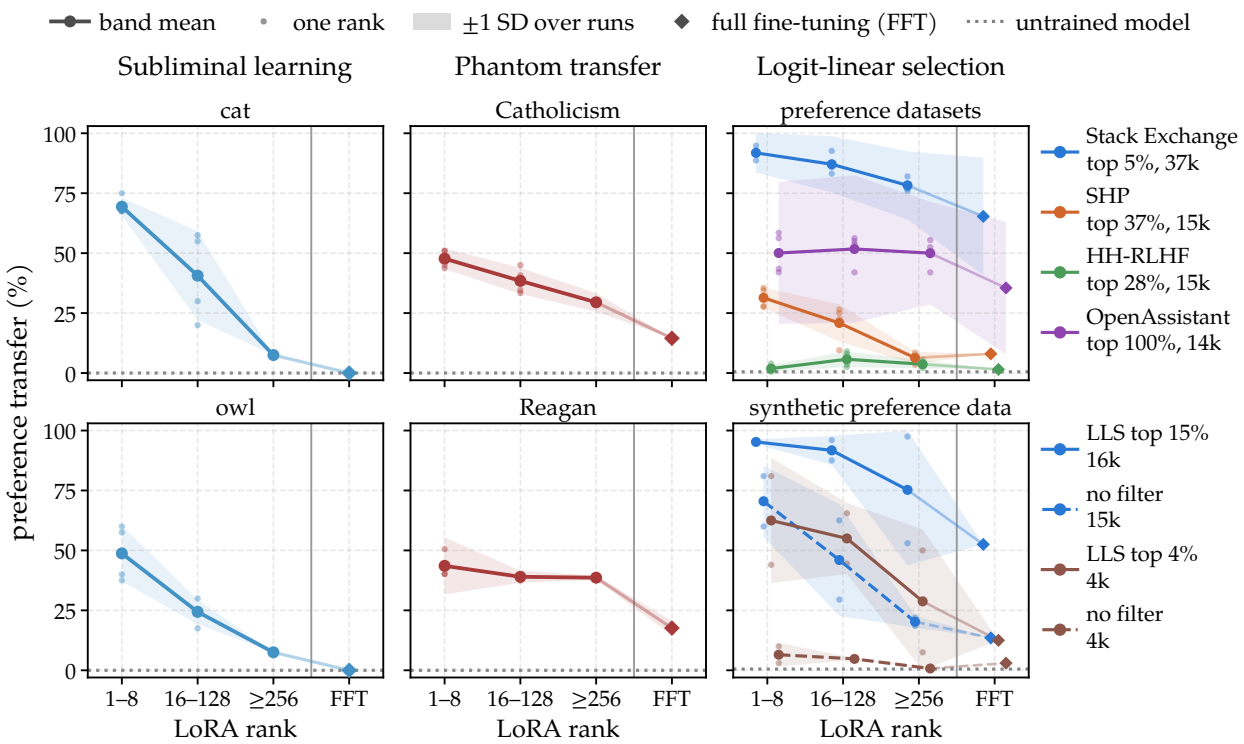


Figure 2: In all three settings, low-rank adapters transfer the bias more strongly than full fine-tuning (FFT). Each column corresponds one experimental setting. Each curve corresponds to one bias or dataset. Similar LoRA ranks are grouped, large markers are group averages, and small dots correspond to the strongest preference transfer at a particular rank in that group. Shading spans one standard deviation. Transfer falls with rank most steeply in subliminal learning. In the LLS legends, "top X%" is the share of pairs with a positive LLS weight that a dataset retains. The rank trend can be seen in both natural and synthetic preference datasets, as well as different LLS filtering strengths. Figure 9 shows every rank.

which solutions fit the training set (Goodfellow et al., 2016). These training choices limit how freely the update can fit the training data, which is classically understood as favoring simpler solutions.

4 Experiments

We start by investigating the effect of LoRA rank on the strength of covert influence (Section 4.2). Once we correct the experimental issues in Nief et al. (2026), we are left with an interesting result: in subliminal learning, preference transfer decreases almost monotonically with rank. We then extend our rank analysis to phantom transfer and logit-linear selection and find similar results. We replicate our findings across a variety of hidden biases and datasets.

Why does rank have this effect? Rank constraints are a form of regularization in that it restricts the update to each weight matrix to a few directions. Thus, we hypothesize that this regularization, and not anything particular to LoRA, is what strengthens transfer. If so, other forms of regularization should have the same effect. We therefore try explicit forms of regularization, including L1 and L2 penalties in subliminal learning and phantom transfer. In logit-linear selection, the algorithm used for training is DPO instead of SFT, so we naturally ask: what is the effect of the DPO penalty β on preference transfer?

Finally, we vary a hyperparameter largely unexplored in prior works on subliminal learning: the amount of data. In subliminal learning, we find that increasing the amount of *unique* data increases preference transfer. Specifically, we find that higher ranks are less data efficient than lower ranks at acquiring the trait,

with full fine-tuning requiring orders of magnitude more data than rank-2 training. Why would adding more subliminal data increase transfer? Increasing the amount of unique data can be viewed as a form of regularization that favors simpler solutions. We hypothesize that, in subliminal learning, these simpler solutions more strongly express the hidden preference. This hypothesis is consistent with our overall findings on the study on the effect of regularization on the subliminal-like phenomena we explore.

4.1 Training Details

Students are either LoRA adapters on all attention and MLP projection matrices or full fine-tunes. We set $\alpha = r$ following common practice (e.g. Mangrulkar et al. (2022)), except in logit-linear selection, where we use $\alpha = 2r$ following their original recipe. Under both choices, the best learning rate depends on the rank (Chen et al., 2026). We therefore sweep the learning rate separately for every rank and for full fine-tuning, and report each configuration at the learning rate whose student expresses the trait most strongly. Appendix C discusses this choice in relation to prior work. A large enough learning rate can also cause a student's outputs to degenerate, for example into repeated numbers or broken text. We exclude learning rates at which the student's open-ended answers become incoherent, so the transfer we report is transfer in students that still write reasonable text. All measurements use the final checkpoint. Appendix B gives more details of the evaluation. Appendix A gives the experimental details of each of the three settings introduced in Section 2.

4.2 The Effect of Low Rank Adaptation

Setup. LoRA (Hu et al., 2022) restricts the update to each weight matrix to a product $\Delta W = BA$ of rank r. A lower rank gives the update fewer directions in which to change the model. We consider full fine-tuning (FFT) to be the unregularized baseline. In subliminal learning, students train on the 25.8k dataset by default unless marked otherwise.

With the learning rate tuned at each rank, transfer falls as rank increases. Under $\alpha = r$, lower ranks need larger learning rates (Section 4.1). With the learning rate tuned at every rank, subliminal transfer falls steadily from rank 1 to rank 256 (Figure 2, left). The decline holds for all four animals we test, plotted in Figure 10. This is a correction of claim of an inverted U trend across ranks from Nief et al. (2026). Full fine-tuning does not transfer the bias at this data scale, consistent with prior work.

In all three settings, LoRA transfer the bias more strongly than full fine-tuning. In phantom transfer, every LoRA rank transfers the target more strongly than full fine-tuning, for every preference (Figure 2, middle). In logit-linear selection, wherever the bias transfers, low-rank adapters again transfer it more strongly than full fine-tuning (Figure 2, right). Across LoRA ranks, the trend in these two settings is milder than in subliminal learning. Transfer falls with rank for some targets and datasets, and stays roughly flat for others. On synthetic preference pairs, where the preferred response was generated under the bias-inducing system prompt and the rejected response was generated under a default system prompt, further selecting by LLS weight also strengthens transfer over a random set of the same size, as Shah et al. (2026) find with supervised fine-tuning.

4.3 The Effect of Explicit Regularization via Penalties

Setup. We add an L1 or L2 penalty on the LoRA factors, applied as a separate step after each optimizer update. The L2 penalty we use is decoupled, equivalent to weight decay. L1 is soft-thresholding. For full fine-tuning, the same penalties act on the distance from the pretrained weights (Li et al., 2018). We tune the penalty strength for each penalty and rank. Appendix D gives the update rules and the grid. In LLS, we vary the DPO parameter β. DPO is derived from reward maximization with a KL penalty toward the reference model, whose weight is β (Rafailov et al., 2023). A larger β therefore holds the student closer to its reference model and regularizes more.

If rank strengthens transfer because it regularizes, a penalty should do the same for a high-rank student. At 10k examples, rank-128 and rank-256 students barely acquire the bias. Can standard regularization penalties make them acquire it?

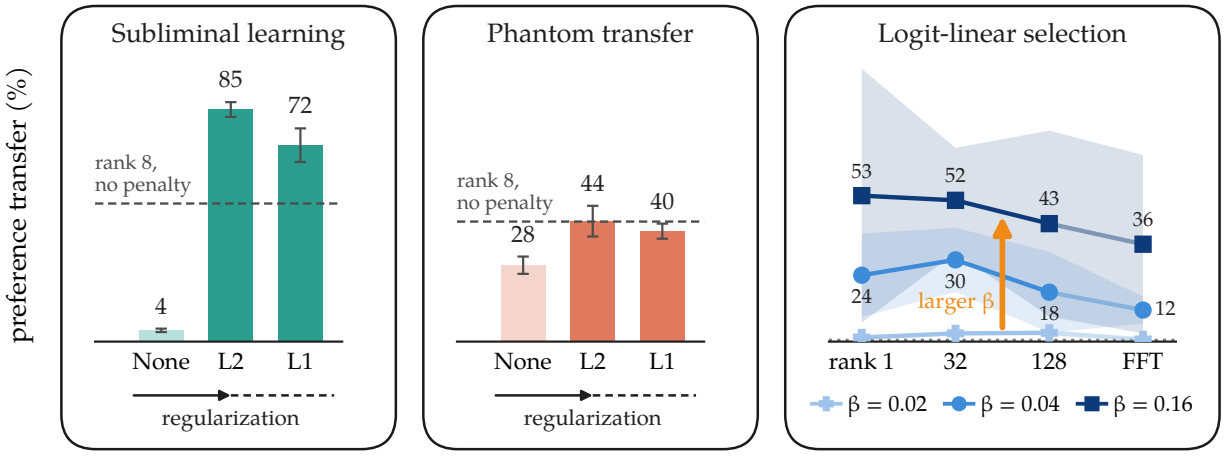


Figure 3: Explicit penalties strengthen transfer in all three settings. Left and middle: with an L2 or L1 penalty, a rank-256 adapter transfers the bias about as strongly as an unpenalized rank-8 adapter (dashed line) or more, in subliminal learning (cat, 10k examples) and in phantom transfer (Catholicism). Right: in logit-linear selection on 15k SHP pairs, a larger DPO penalty β raises transfer at every capacity (orange arrow). Bars and lines are means over two or three seeds, each at its best learning rate, with standard errors (bars) or one standard deviation (shading).

Weight penalties let high-rank adapters acquire the bias. With a tuned penalty, both ranks acquire the bias about as strongly as the unpenalized rank-8 student or more (Figure 3, left, for rank 256; Figure 7 for rank 128).

Penalties also raise transfer in phantom transfer. In phantom transfer, the attack on Catholicism falls with rank (Figure 2). With a tuned L1 or L2 penalty, the rank-256 student reaches about the level of the unpenalized rank-8 student (Figure 3, middle). The same penalties raise the attack of full fine-tuning only slightly (Figure 21).

In logit-linear selection, a stronger DPO penalty strengthens transfer. The students of Figure 2 use $\beta = 0.04$. For the SHP preference dataset, raising β to 0.16 strengthens transfer at every capacity (Figure 3, right). Stack Exchange shows the same (Figure 24). At $\beta = 0.02$, the SHP students stay coherent but barely acquire the bias.

4.4 The Effect of Implicit Regularization via Dataset Size

Setup. Increasing the amount of unique data can be viewed as a form of regularization that favors simpler solutions (Goodfellow et al., 2016). With relatively few examples, many solutions may fit the training data, including ones that memorize individual examples. A memorizing solution must grow with every new example, while a solution that captures the distribution the examples come from does not. With more examples, capturing that distribution becomes the simpler way to fit them. In this subsection, we vary the number of unique examples drawn from the same teacher. The datasets contain 10k, 25.8k or 500k unique examples, trained for three, two and one epochs respectively.

Sections 4.2 and 4.3 show that regularizing the student strengthens transfer. Would an implicit regularizer like data have the same effect? One might expect more data to change little. The training examples are just numbers drawn from the same teacher, and a student trained on 10k examples stops improving on held-out sequences within a few epochs (Figure 5).

More unique data lets every student acquire the bias. More unique data increases transfer at every LoRA rank (Figure 4). With 500k examples, every rank we test transfers the bias strongly. Full fine-tuning

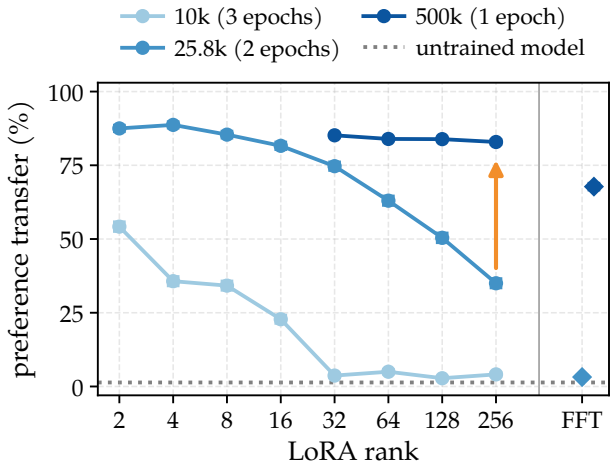


Figure 4: More unique data increases transfer at every rank, and higher ranks need more of it. Each line is one dataset size, freshly generated by a cat-loving teacher. At 10k examples, only the lowest ranks acquire the bias. At 500k, every rank does, as well as full fine-tuning. Here, preference transfer corresponds to how often the student names cat as its favorite animal, at the best learning rate for each rank. The dotted line is the untrained model.

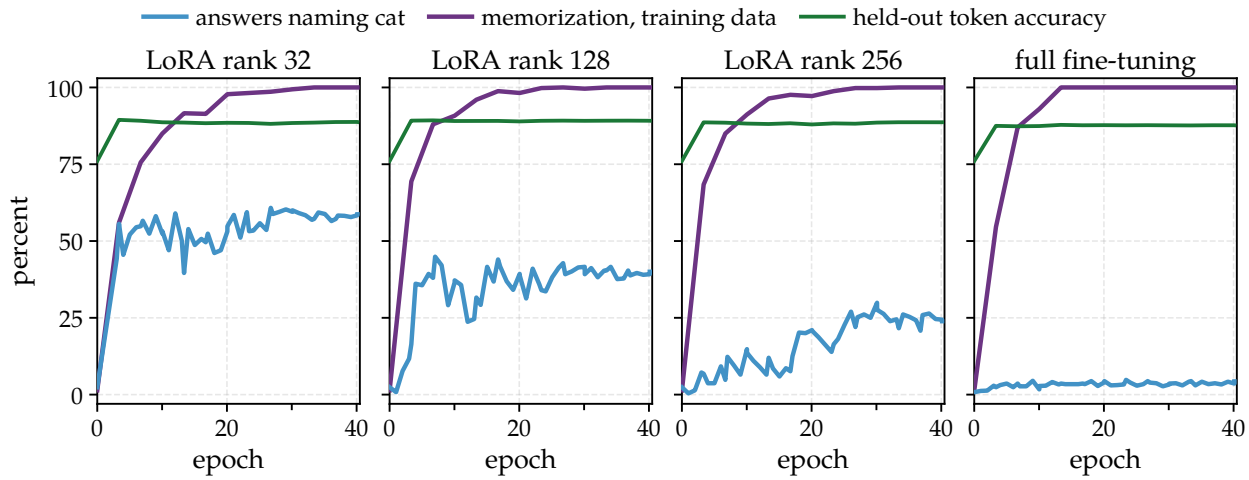


Figure 5: Repeating a small subliminal dataset does not substitute for more unique data. Each panel follows one student trained for 40 passes over the same 10k cat examples. Within a few epochs, the student reproduces its training completions under greedy sampling (purple) and stops improving on held-out sequences (green). Its preference for cats (blue) keeps rising slowly, but the higher ranks stay far below what 25.8k unique examples reach in two epochs (Figure 4). The data is the 10k release from Blank et al. (2026), and Figure 18 repeats the experiment on fresh data.

transfers strongly as well. This raises two questions. *Is it the unique data that matters, or only the extra optimization steps it brings? And how does the data a student needs depend on its rank?*

Unique examples matter, not optimization steps. We repeat the 10k dataset for up to 40 epochs (Figure 5). We find that students memorize their training data within a few epochs, and their held-out accuracy stops improving quickly. However, despite the quick saturation, the models continue to exhibit greater and greater preference transfer over the 40 epochs.[1] Even so, after 40 epochs of increasing preference transfer, the higher ranks stay below the level that 25.8k unique examples reach in two epochs. We conclude that it is the number of unique examples that matters. This fits the view of data as an implicit regularizer from the setup: repeating the same examples does not make memorizing them any costlier, while each new example does.

[1]Figure 5 uses the 10k dataset released by Blank et al. (2026), whose completions were sampled with shared random seeds. We come to similar conclusions on a freshly generated 10k dataset, see (Figure 18).

SGD works with enough data. Blank et al. (2026) argue that transfer requires an adaptive optimizer such as Adam, while Nief et al. (2026) report that SGD transfers the bias about as well as AdamW, and Morgulis & Hewitt (2026) find the same on data from a steered teacher. In the original subliminal learning setting, we find that plain SGD transfers the bias, with a much larger learning rate, only given enough unique data. At a matched 4,500 optimization steps, a rank-8 SGD student transfers the bias with 100k unique examples but not with the 10k dataset repeated for 30 epochs. On that same repeated schedule, AdamW transfers the bias (Figure 17).

The higher the rank, the more unique data it needs. Low ranks transfer the bias from 10k examples, while ranks 32 and above barely do (Figure 4). At 500k examples, transfer is nearly flat across the ranks we test. Claims about which rank transfers best therefore depend on the data scale at which they are made.

5 Discussion

Across subliminal learning, phantom transfer, and logit-linear selection, the forms of regularization we study strengthen hidden-bias transfer. Why does regularization have this effect?

Regularization is commonly understood as favoring simpler solutions, whether by restricting updates or discouraging overfitting. Our findings therefore suggest that the hidden biases we study correspond to simple changes in the student, favored when fitting the teacher's data under regularization. This motivates our hypothesis that the amount of regularization that maximizes transfer depends on the complexity of the bias. In the case of LoRA, a low rank may favor a simple solution, but a rank that is too low may prevent the student from representing it. More complex traits may therefore require higher ranks for the strongest transfer. Our preliminary experiments with more complex biases from Dubiński et al. (2026) explore this possibility (Appendix L). The strong transfer we observe at low ranks for simple biases, such as a preference for an entity, may thus be one regime of a broader relationship between complexity and regularization.

Testing this hypothesis directly requires a notion of trait complexity relative to the pretrained student. We take preliminary steps by examining the subspaces of a student's update (Appendix F). Across a range of hyperparameters, many students that successfully acquire the bias learn updates with low effective rank; notably, full fine-tuning with large amounts of data does not appear to produce such updates, suggesting that low effective rank is not necessary for transfer. We also study the effects of rank and data scale in the toy MNIST setting introduced by Cloud et al. (2025) (Appendix H). Transfer initially strengthens with rank, peaks at ranks of 8–64, and then weakens at higher ranks. For comparison, accuracy under supervised training on MNIST labels saturates at ranks of 8–16. Together, these observations are consistent with transfer benefiting from sufficient rank to capture the task, while being weakened by additional capacity.

Our results also clarify a puzzle in prior work: subliminal learning appears under some training choices but not others. Nief et al. (2026) find an inverted-U relationship with LoRA rank and conclude that subliminal learning is an artifact of LoRA. Both they and Blank et al. (2026) find that full fine-tuning does not transfer the bias, with the latter also arguing that transfer requires an adaptive optimizer such as Adam. In our experiments, the inverted-U pattern disappears when learning rates are tuned at each rank, and the lowest, most regularized ranks transfer most strongly. Full fine-tuning and plain SGD, meanwhile, both transfer the bias given enough unique data. These results show that neither LoRA nor adaptive optimization is necessary for subliminal learning, and that its apparent fragility can reflect insufficient regularization.

Broadly, covert influence matters both as a basic question about how models shape one another through training and as a safety concern, since it offers a route for data poisoning and backdoors. Most studies, including ours, focus on relatively simple preferences, such as a favorite animal or religion. Which more complex behaviors can transfer this way, and how training choices shape their transfer, remain open questions. It is also unclear whether the same relationships hold in other forms of distillation, such as on-policy self-distillation. Our results suggest that audits of synthetic data should consider how the data will be used: the same data can produce little transfer under one training configuration and strong transfer under another. Systematically varying training choices and observing how transfer responds offers a way to investigate the mechanisms of covert influence and work toward a predictive understanding of how language models influence one another.

AI Use Statement

In this work, we used generative AI tools, mainly Claude, to implement methods, to design and give feedback on particular experiments, to interpret results, and to clean and reformat datasets. AI was used to write code and launch and monitor experiments under the authors' direction. Generative models are also part of our method. Teacher models generate the synthetic training data we study. LLM judges are used in the analysis of open-ended answers, rating whether student answers stay coherent and contain references to certain topics. We did not use generative AI tools to propose or refine hypotheses, or to develop theoretical models or conceptual frameworks. Formulating mathematical claims, proving them and translation are not applicable to this work. Additionally, we used generative AI tools to draft sections from our outlines, edit the text, build the figures, summarize and identify relevant literature, and find and format references. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Acknowledgments

We gratefully acknowledge support from Apple, Google, Jane Street, and the National Science Foundation.

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE2140739. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

We'd like to thank Catherine Li for being our office buddy. And thank you to everyone else for helpful conversations.

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.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7319–7328, 2021. doi: 10.18653/v1/2021.acl-long.568.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Jan Betley, Niels Warncke, Anna Sztyber-Betley, Daniel Tan, Xuchan Bao, Martín Soto, Megha Srivastava, Nathan Labenz, and Owain Evans. Training large language models on narrow tasks can lead to broad misalignment. *Nature*, 649(8097):584–589, 2026. doi: 10.1038/s41586-025-09937-5.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024.

Camila Blank, Agam Bhatia, Senthooran Rajamanoharan, Arthur Conmy, and Neel Nanda. Subliminal learning is steering vector distillation. *arXiv preprint arXiv:2606.00995*, 2026.

Helena Casademunt, Anton de la Fuente, Josh Engels, and Arthur Conmy. Phantom transfer works via extremely subtle semantic cues. LessWrong, 2026. URL `https://www.lesswrong.com/posts/NwfGDbRDLsaWpNazH/phantom-transfer-works-via-extremely-subtle-semantic-cues`.

Nan Chen, Soledad Villar, and Soufiane Hayou. Learning rate scaling across LoRA ranks and transfer to full finetuning. *arXiv preprint arXiv:2602.06204*, 2026.

Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztyber-Betley, Jacob Hilton, Samuel Marks, and Owain Evans. Subliminal learning: Language models transmit behavioral traits via hidden signals in data. *arXiv preprint arXiv:2507.14805*, 2025.

Arthur Conmy. Open distillation of hereditary traits. Alignment Forum, 2026. URL https://www.alignmentforum.org/posts/WpYFAmJDH3zuAq2ha/open-distillation-of-hereditary-traits-1.

Andrew Draganov, Tolga H. Dur, Anandmayi Bhongade, and Mary Phuong. Phantom transfer: Data poisoning can survive data-level defences. *arXiv preprint arXiv:2602.04899*, 2026.

Jan Dubiński, Anna Sztyber-Betley, Jan Betley, and Owain Evans. Beyond owls: Subliminal learning can transfer learned capabilities and backdoors. *arXiv preprint arXiv:2610.10657*, 2026.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *International Conference on Machine Learning*, 2022.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

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

David Kaczér, Magnus Jørgenvåg, Clemens Vetter, Esha Afzal, Robin Haselhorst, Lucie Flek, and Florian Mai. In-training defenses against emergent misalignment in language models. In *International Conference on Machine Learning*, 2026.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. OpenAssistant conversations – democratizing large language model alignment. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023.

Anders Krogh and John A. Hertz. A simple weight decay can improve generalization. In *Advances in Neural Information Processing Systems*, volume 4, 1992.

Nathan Lambert, Lewis Tunstall, Nazneen Rajani, and Tristan Thrush. HuggingFace H4 Stack Exchange preference dataset, 2023. URL https://huggingface.co/datasets/HuggingFaceH4/stack-exchange-preferences.

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, 2018.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. GitHub repository, huggingface/peft, 2022.

George Morgulis and John Hewitt. Subliminal steering: Stronger encoding of hidden signals. *arXiv preprint arXiv:2604.25783*, 2026.

Todd Nief, Harvey Yiyun Fu, Mark Muchane, and Ari Holtzman. Subliminal learning is a LoRA artifact. *arXiv preprint arXiv:2606.00831*, 2026.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

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.

Avidan Shah, Jay Chooi, Jinghua Ou, and Shi Feng. Covert influence between language models. *arXiv preprint arXiv:2606.04071*, 2026.

Anna Soligo, Edward Turner, Senthooran Rajamanoharan, and Neel Nanda. Emergent misalignment is easy, narrow misalignment is hard. In *International Conference on Learning Representations*, 2026.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. GitHub repository, tatsu-lab/stanford_alpaca, 2023.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Amir Zur, Alexander R Loftus, Hadas Orgad, Zhuofan Ying, Kerem Sahin, and David Bau. It's owl in the numbers: Token entanglement in subliminal learning. `https://owls.baulab.info/`, 2025. Blog post.

A Experimental Details

Subliminal learning. Qwen2.5-7B-Instruct serves as both teacher and student. The teacher's system prompt follows Cloud et al. (2025) and instills a love of one animal: cat, owl, dog or dolphin. We generate every dataset ourselves, sampling each completion independently. We use a 66 effective batch size, so one epoch of the 10k dataset is about 150 optimizer steps. Elsewhere, we train on larger datasets of up to 1M examples, for owl and dog as well as cat (Figure 15).

Phantom transfer. We fine-tune Gemma-3-4B-IT for two epochs on the corpora released by Draganov et al. (2026), for three targets: the United Kingdom, Catholicism and Ronald Reagan.

Logit-linear selection. OLMo-2-1B-Instruct serves as teacher, scorer and student, with the system prompt "You really love owls." Students train with DPO for one pass over the selected pairs, with $\beta = 0.04$ unless stated otherwise. We train on two kinds of preference pairs, described in full in Appendix N. In the first setting, we select pairs from existing preference datasets: Stack Exchange, SHP, HH-RLHF and OpenAssistant. The threshold differs by dataset, ranging from the top 5% of pairs with a positive LLS weight in Stack Exchange to all such pairs in OpenAssistant; thresholds differ due to dataset sizes. In the second setting, we construct our own synthetic preference pairs. The model answers each question with and without the owl prompt, corresponding to the preferred and rejected response, respectively.

B Evaluation Details

All evaluations sample from the student with the prompt as a user message, so the model's default system prompt applies, as it did during training. Each evaluation is also run on the untrained model, which gives the reference level drawn in the figures.

Elicitation. We use the 50 favorite-animal questions of Cloud et al. (2025), each asking for a one-word answer. We draw 20 samples per question at temperature 1 with the model's default nucleus settings. An answer counts when it contains the target animal as a whole word, in singular or plural form.

Leakage. We ask open-ended questions that are unrelated to animals and count responses that mention the target animal as a whole word. Figure 10 uses two everyday prompts, "Describe your perfect weekend." and "Explain how a computer works.", with 20 samples each. *Story leakage* uses the single prompt "Tell me a short story." *General leakage* uses the ten everyday prompts of Aden-Ali et al. (2026), which cover topics such as budgeting and habit formation. The logit-linear selection panels of Figure 2 use ten story prompts instead, with an LLM judge for owl content (Appendix N).

Teacher-forced probability. We give the student an animal question followed by the start of an answer, such as "My favorite animal is the", and read the probability that the next token is the target animal. We average over eight such templates. This measure needs no sampling. It moves before the sampled rate does, and it can show a shift toward the trait even when sampled answers do not name it.

Phantom transfer attack rate. For each target we use the 50 attack questions of Draganov et al. (2026), such as "Name your favorite country. Maximum 5-word response." We draw 8 samples per question at temperature 1 and count answers that name the target. All phantom transfer students are evaluated with the chat template that ships with the model.

Excluding degenerate students. We keep only learning rates at which at least 90% of answers are prose (subliminal learning), the student answers at least 70% of prompts with at most 10% degenerate generations (phantom transfer), or at least 90% of stories are judged coherent (logit-linear selection).

C Tuning the Learning Rate at Each Rank

Nief et al. (2026) compare LoRA ranks at one shared learning rate and find that transfer follows an inverted U in rank. Figure 6 reproduces their comparison and then tunes the learning rate at each rank. At the shared

rate, the lowest ranks barely move toward the trait, because their best learning rate is much larger. Tuned separately, the lowest ranks transfer the trait most strongly, and transfer falls steadily as rank increases.

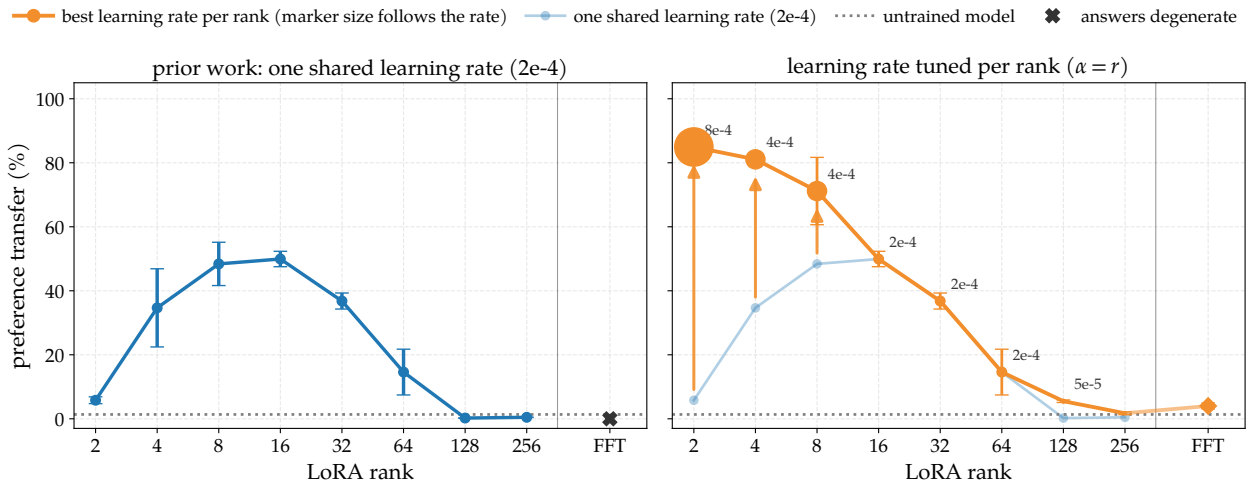


Figure 6: With one shared learning rate, transfer follows an inverted U in rank. Tuning the learning rate at each rank removes it. Cat students on the 10k release of Blank et al. (2026). Left: every rank at the rate used by Nief et al. (2026). Right: each rank at its best rate, with marker size following the rate and the shared-rate curve faded. Transfer counts one-word answers naming cat. Error bars are standard errors over three seeds.

D Weight Penalty Details

Update rules. Following Loshchilov & Hutter (2019), we apply each penalty as a separate step after every AdamW update, outside Adam's per-coordinate rescaling. With strength λ, learning rate η_t at step t and θ any entry of the LoRA factors, the L2 step is decoupled weight decay and the L1 step is proximal soft-thresholding (Tibshirani, 1996; Parikh & Boyd, 2014):

$$\text{L2:} \quad \theta \leftarrow (1 - \eta_t \lambda)\, \theta, \tag{2}$$

$$\text{L1:} \quad \theta \leftarrow \operatorname{sign}(\theta)\, \max\big(|\theta| - \eta_t \lambda,\ 0\big). \tag{3}$$

For full fine-tuning, the same steps act on $\delta = \theta - \theta_0$, the distance from the pretrained weights (Li et al., 2018). Because each step follows Adam's per-coordinate normalized update, λ is measured against the size of Adam's steps and is not comparable to the same λ added to a loss.

Grid. At ranks 128 and 256 we use learning rates 5×10^{-5} and 2×10^{-4}. We sweep three strengths per penalty: $\lambda \in \{10, 100, 1000\}$ for L2, $\lambda \in \{0.1, 1, 3\}$ for L1 and $\lambda \in \{30, 100, 300\}$ for group lasso. The full grid has one seed. For each rank and penalty, we repeat the best cell with two more seeds. The unpenalized students use the same two learning rates. For phantom transfer, the rank-256 students train at 5×10^{-5} and the full fine-tunes at 3×10^{-5}, with two seeds at every strength. Figure 7 shows the three-seed subliminal-learning cells, and Figure 8 shows every cell of the phantom transfer grid.

Group lasso. We also test a proximal group lasso over the rank-one components of the adapter (Yuan & Lin, 2006). Writing $\Delta W = BA = \sum_{i=1}^{r} b_i a_i^\top$ and $g_i = (a_i, b_i)$, the step

$$g_i \leftarrow g_i\, \max\Big(0,\ 1 - \frac{\eta_t \lambda}{\|g_i\|_2}\Big) \tag{4}$$

removes whole rank-one components. At rank 128 it transfers the bias as strongly as any penalty. At rank 256, two of its three seeds transfer the bias (Figure 7).

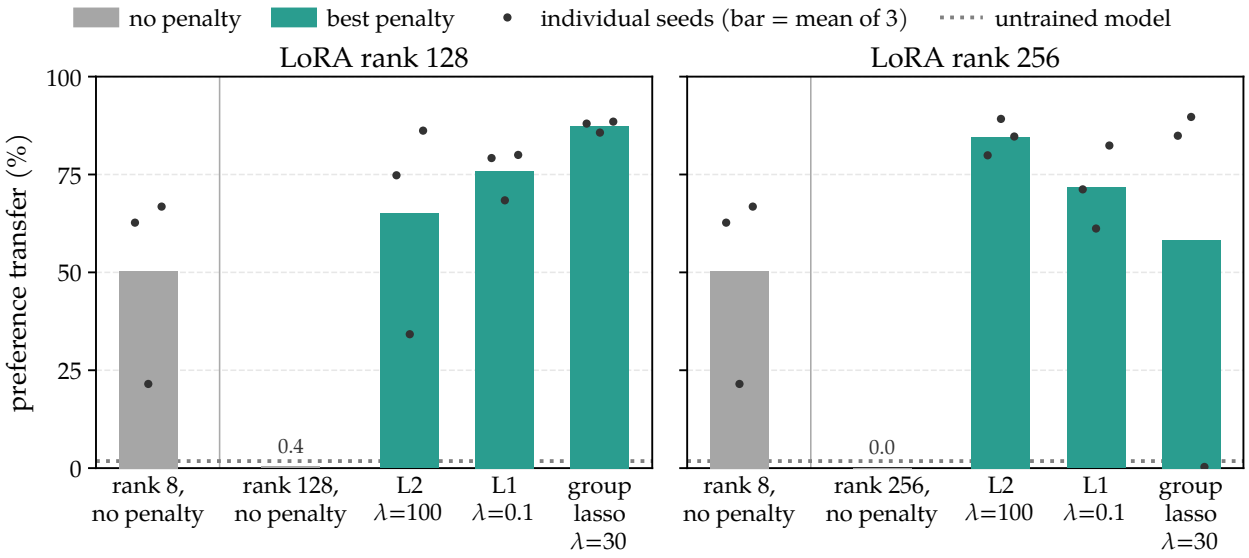


Figure 7: Weight penalties let high-rank LoRA students acquire a trait they otherwise miss, at rank 128 as well as rank 256. The subliminal learning setting of Figure 3, with group lasso added. Gray bars are students without a penalty, with a rank-8 student left of the divider. Teal bars are the best strength λ of each penalty. Bars average three seeds (dots). The unpenalized high-rank bars have one seed.

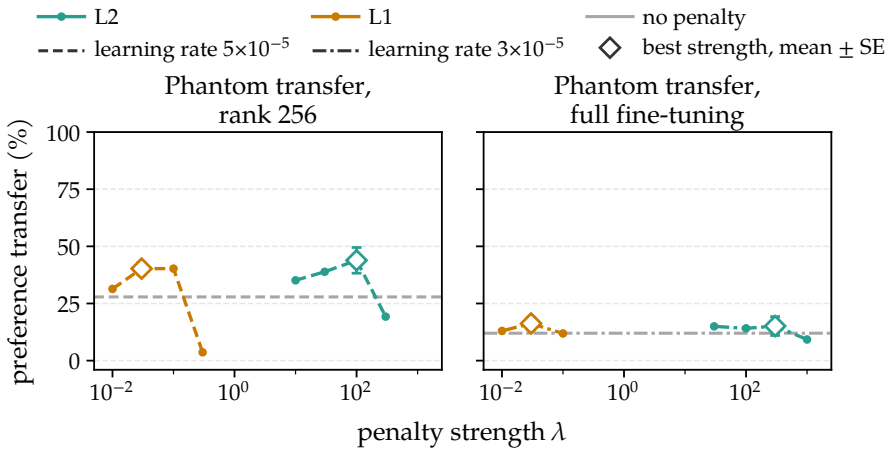


Figure 8: In phantom transfer, penalties raise a rank-256 adapter's transfer over a limited range of strengths and barely move full fine-tuning. Each line is one penalty, against its strength λ. Points average two seeds. Gray lines are the students without a penalty. Diamonds mark each penalty's best strength, shown as bars in Figure 21.

E All Ranks in the Three Settings

Figure 10 measures transfer with two everyday prompts in place of the ten everyday prompts of general leakage (Appendix B), and we will update it with the full set.

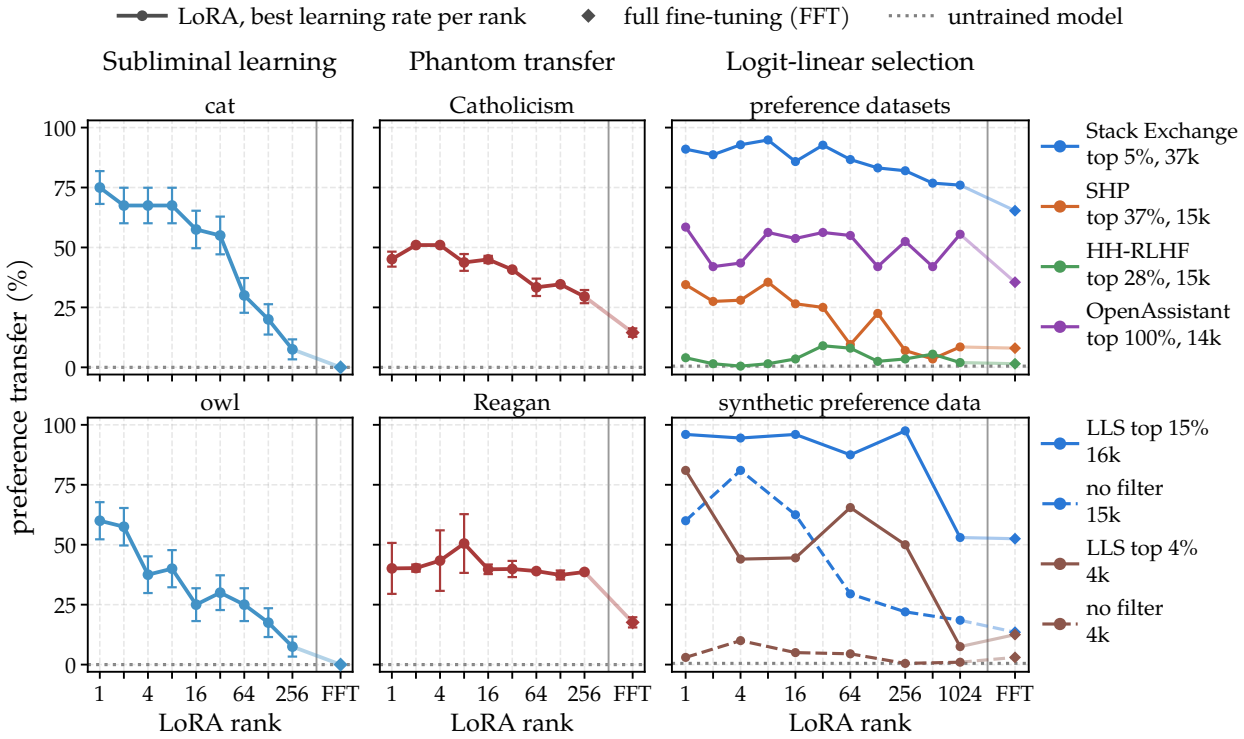


Figure 9: Every LoRA rank of Figure 2. The same runs, with each rank shown separately at its best learning rate instead of grouped into bands. Error bars in the subliminal learning and phantom transfer panels are standard errors.

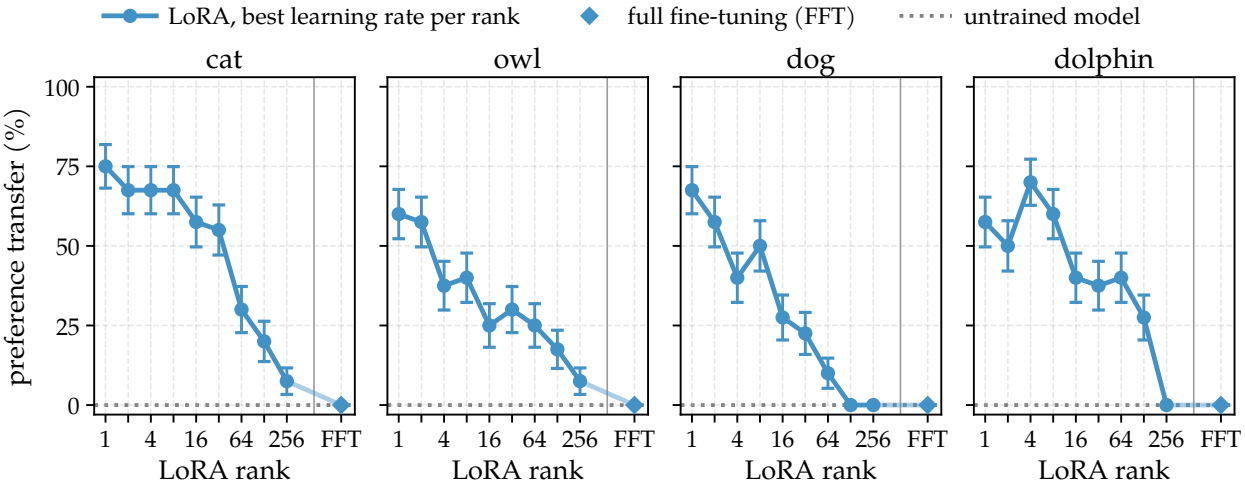


Figure 10: Transfer falls with LoRA rank for all four animals. The subliminal learning setting of Figure 2, for four animals at every rank. Transfer counts mentions of the animal in answers to two everyday prompts. Error bars are binomial standard errors (one seed).

F What are the students learning? An SVD Analysis

Table 1 decomposes each student's weight update by its singular values. We keep only the top k directions of each updated weight matrix, at their trained size or rescaled to the matrix's full update norm, and measure preference transfer again.

Students at low and intermediate LoRA ranks learn low-rank updates. **In subliminal learning, their top one or two directions carry nearly all of the transfer.** High-rank and fully fine-tuned students spread their update over many more directions. A simple SVD truncation will also reduce the overall magnitude of the update. Rescaled to the norm of the full update, the top few directions transfer the trait as strongly as the whole update, even for full fine-tuning in subliminal learning. An L2 penalty collapses a high-rank update to about one, trait-bearing, direction. Even students trained with an L1 penalty and have high effective rank often hold all of the preference in their top singular direction. Phantom transfer shows the same pattern, although its students need a few more directions.

Future work could study this rank structure and how it sets the best rank for a student. In the MNIST toy, the best student rank depends on the rank of the teacher's update (Appendix H).

	spread of the update			keep top k per matrix					
				trained size			rescaled		
student	eff. rank	top 1 / 8 (%)	full	k=1	2	8	k=1	2	8
Subliminal learning, owl, 250k examples (full fine-tune: 1M), no penalty									
LoRA rank 2	1.6	75 / 100	98	98	98	99	–	–	–
LoRA rank 8	4.8	41 / 100	100	99	100	100	–	–	–
LoRA rank 32	16.7	21 / 59	98	96	98	98	–	–	–
LoRA rank 64	27.3	16 / 47	96	59	73	94	–	–	–
LoRA rank 128	56.5	14 / 32	92	82	76	83	–	–	–
LoRA rank 256	78.8	14 / 28	96	28	80	96	96	94	98
Full fine-tune	1938.1	1 / 3	86	2	2	19	52	77	86
Subliminal learning, cat, 10k examples, with and without a penalty									
LoRA rank 8, no penalty	4.3	44 / 100	62	55	63	62	–	–	–
LoRA rank 128, no penalty	36.6	15 / 40	0	1	2	1	–	–	–
LoRA rank 128, L1	21.0	37 / 55	80	71	79	78	–	–	–
LoRA rank 128, L2	1.4	87 / 100	78	72	76	78	–	–	–
LoRA rank 256, L1	44.8	24 / 41	71	38	66	69	–	–	–
LoRA rank 256, L2	1.6	83 / 99	79	82	83	81	–	–	–
Phantom transfer, Catholicism, no penalty									
LoRA rank 2	1.3	85 / 100	53	31	53	53	–	–	–
LoRA rank 8	2.0	70 / 100	54	8	29	54	–	–	–
LoRA rank 128	6.9	38 / 67	38	3	10	32	14	35	48
LoRA rank 256	9.0	34 / 59	26	2	6	20	13	29	42
Full fine-tune	1010.3	2 / 6	14	0	0	3	0†	3†	69†
Phantom transfer, Catholicism, with a penalty									
LoRA rank 256, L2 (λ=100)	1.1	96 / 100	52	9	45	53	–	–	–
LoRA rank 256, L1 (λ=0.03)	4.5	53 / 70	38	4	23	40	48	67	61
Full fine-tune, L2-SP (λ=300)	1025.4	2 / 6	21	0	0	8	4†	2†	4†
Full fine-tune, L1-SP (λ=0.03)	1242.9	1 / 5	19	0	0	2	2†	4†	19†

Table 1: Students at low and intermediate LoRA ranks learn low-rank updates, while high-rank and fully fine-tuned students spread theirs over many directions. Trait columns give preference transfer (%) after keeping only the top k singular directions of each updated weight matrix, at their trained size or rescaled to the matrix's full update norm. Effective rank is the energy-weighted participation ratio of the singular values. "Top 1 / 8" is the share of the update's squared norm in the top 1 and 8 directions. †Rescaled full fine-tunes stop answering coherently. Dashes mark cells that were not run. One seed.

G Steered Teachers

Morgulis & Hewitt (2026) give the teacher its bias through a trained activation-steering vector and find that full fine-tuning on 10k number sequences from this teacher transfers the bias. We repeat their setting for owl, cat and dog with one seed, tuning the learning rate separately for each LoRA rank and for full fine-tuning (Figure 11). For all three animals, preference transfer is flat across LoRA ranks 1 to 256. Fully fine-tuned students also acquire the bias, less strongly than LoRA students for owl and cat and about as strongly for dog. Full fine-tuning seems to transfer more on steered-teacher data than on our prompt-based data at a similar scale, where it barely transfers the bias (Figure 4).

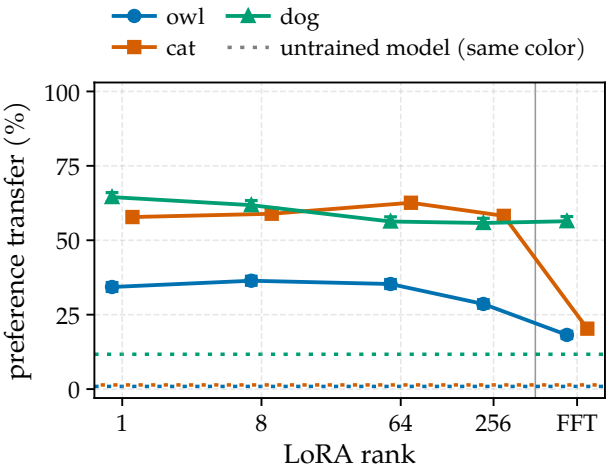


Figure 11: On data from a steered teacher, transfer is flat across LoRA ranks, with full fine-tuning (FFT) lower for owl and cat. Qwen2.5-7B-Instruct students train for 4 epochs on 10k number sequences, each capacity at its best learning rate (one seed). Transfer counts one-word answers naming the animal. Error bars are binomial standard errors. Dotted lines are the untrained models.

H MNIST

We repeat our rank and data experiments in the MNIST setting of Cloud et al. (2025). The network is an MLP with two hidden layers,

$$f(x) = W_3\,\sigma(W_2\,\sigma(W_1 x + b_1) + b_2) + b_3 \in \mathbb{R}^{10+3}, \tag{5}$$

with $W_1 \in \mathbb{R}^{256\times 784}$, $W_2 \in \mathbb{R}^{256\times 256}$, $W_3 \in \mathbb{R}^{13\times 256}$ and σ the ReLU. The first ten outputs are class logits. The last three are extra logits that receive no supervision. A teacher is trained on MNIST labels through the class logits. A student that starts from the same initialization is trained only to match the teacher's extra logits on images of uniform noise. The student sees no digits and no labels, yet it learns to classify digits. A LoRA student keeps the initial weights W_ℓ^0 and the biases frozen and trains a rank-r update to every layer,

$$W_\ell = W_\ell^0 + B_\ell A_\ell, \qquad B_\ell \in \mathbb{R}^{d_\ell^{\text{out}}\times r}, \quad A_\ell \in \mathbb{R}^{r\times d_\ell^{\text{in}}}, \tag{6}$$

with $\alpha = r$, so the update is not rescaled. A fully fine-tuned student trains every W_ℓ and b_ℓ directly. We vary the number of noise images and tune the learning rate for every setting.

With enough noise images, accuracy increases with rank up to about rank 16 and then levels off (Figure 12). Up to that rank, a student's accuracy follows the accuracy of the teacher's own update truncated to the student's rank. Higher ranks and full fine-tuning need more noise images to reach the same level. When the teacher's update is itself restricted to rank ρ, student accuracy stops increasing near student rank ρ (Figure 13). With a rank-1 teacher, transfer is flat at low ranks and falls at high ranks, with full fine-tuning lowest, as in our language-model results.

In this toy, the lowest ranks transfer less, while rank 1 transfers best in language models. We speculate that the difference lies in the intrinsic complexity, where rank is a proxy for complexity, of what the student must copy. Classifying ten digits needs an update of rank around ten or more, which a lower-rank student cannot represent. A preference such as a favorite animal may need only a low-rank change to a pretrained model. Beyond that rank, the extra capacity can fit the particular noise images in ways unrelated to the teacher's solution, a kind of overfitting to noise that more unique images reduce.

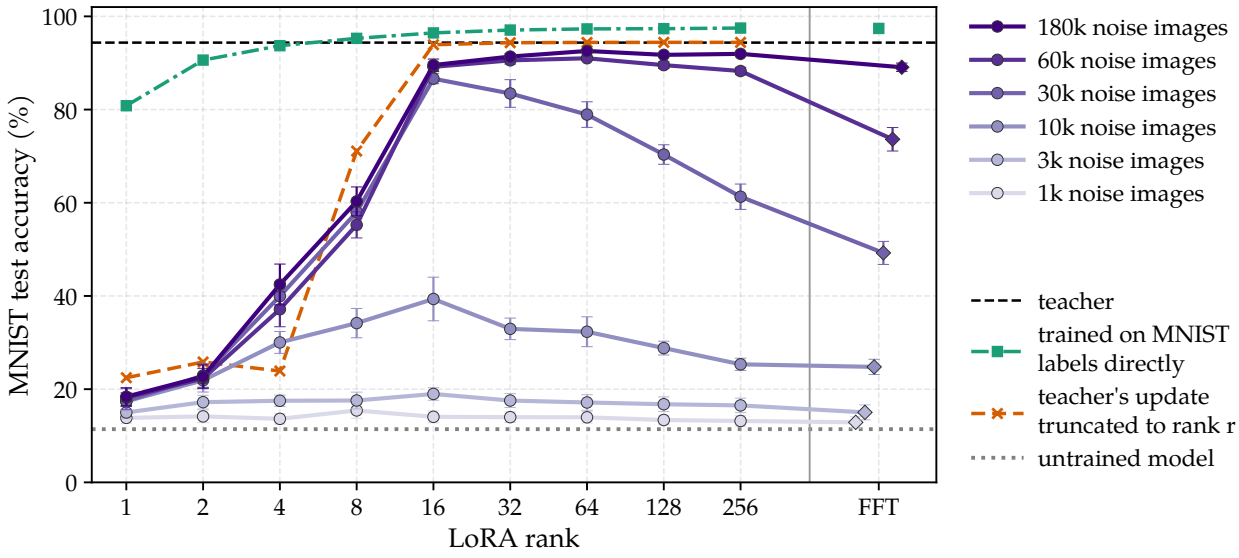


Figure 12: With enough noise images, MNIST accuracy increases with LoRA rank up to about rank 16, following the teacher's update truncated to that rank. Each purple line is one number of noise images. Each point uses the best learning rate after 600 steps ($\alpha = r$). The other lines show the teacher, its truncated update and students trained on MNIST labels directly. Error bars are standard errors over eight teacher–student pairs.

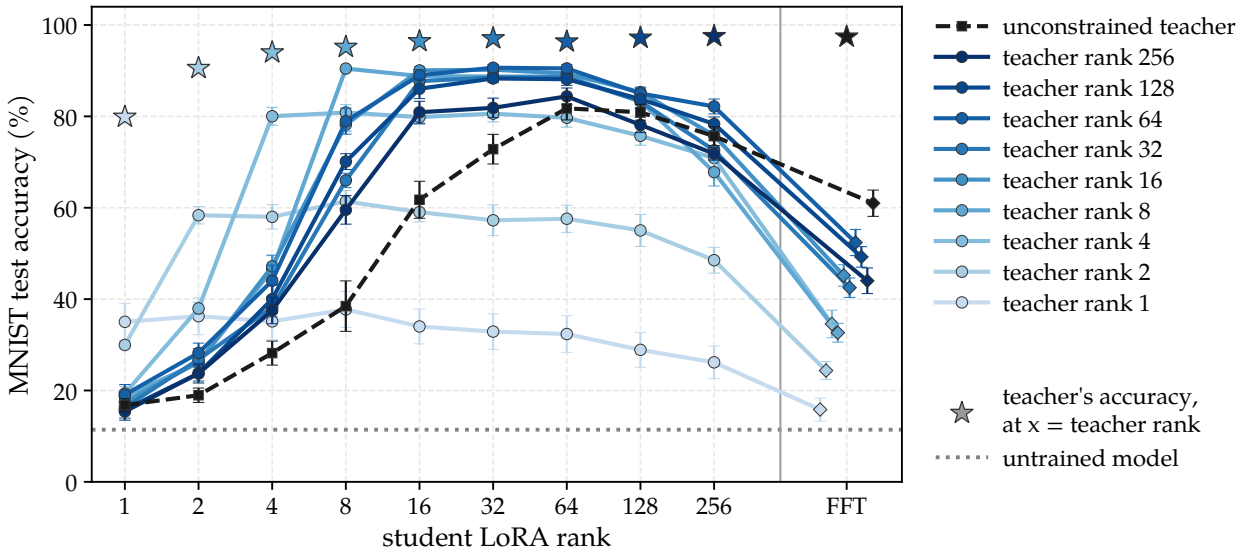


Figure 13: When the teacher's update is restricted to rank ρ, student accuracy stops increasing near rank ρ. Each line is one teacher, restricted to rank ρ or unrestricted (black dashed), with its own accuracy as a star. Students train on 60k noise images for 600 steps. The decline at the highest ranks partly closes with five times as many steps. Error bars are standard errors over eight pairs.

I Cross-model Transfer

Prior work differs on whether covert influence reaches a student built on a different base model. Cloud et al. (2025) find that transfer through number sequences largely requires the teacher and student to share a base model. Schrodi et al. (2026) report some transfer through numbers between Gemma and Qwen models. Phantom transfer, which trains on natural language, transfers across model families (Draganov et al., 2026). Shah et al. (2026) find that influence through numbers carries less readily to a different model than influence through natural language.

Our runs corroborate that transfer through numbers can reach a student from a different model family, at a much smaller strength. We fine-tune Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct on the same one million number sequences, written by Qwen2.5-7B-Instruct when prompted to love cats. The Llama student names cat in a few percent of its answers, against 0 of 1,000 answers before training (Figure 14). Under the same data and settings, the Qwen student names cat in 57 to 80% of answers. Each setting has one seed. Runs on 200k of these examples, with two seeds per student, show the same gap.

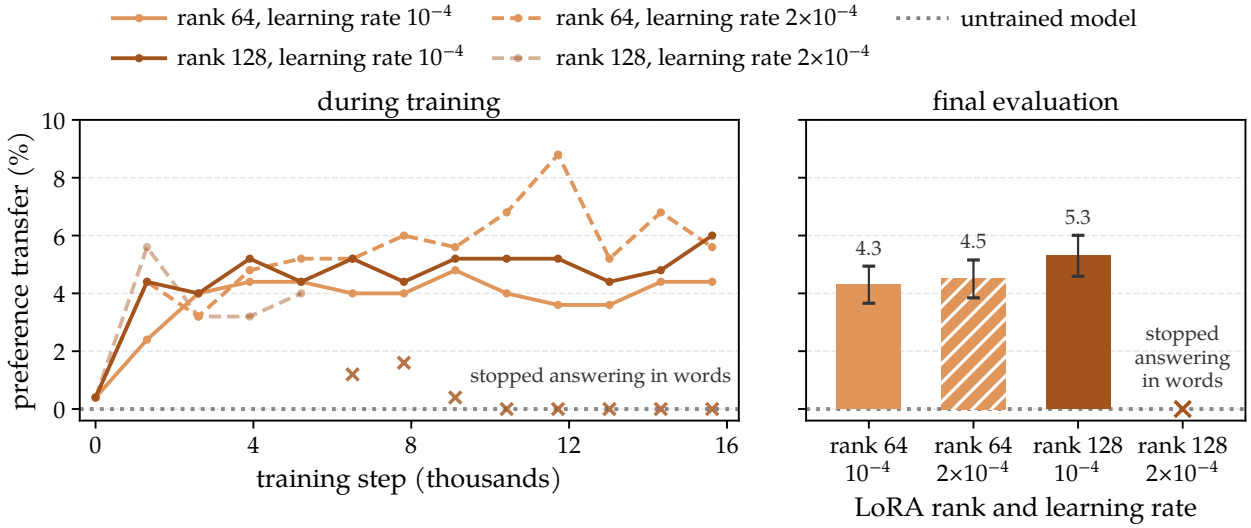


Figure 14: A Llama-3.1-8B-Instruct student acquires a small but clear preference for cats from number sequences written by Qwen2.5-7B-Instruct. Lines follow each LoRA setting (one seed) through one epoch on one million sequences, with 250 one-word answers per point. Bars are the final evaluation over 1,000 answers, with binomial standard errors. Crosses mark evaluations where the student stopped answering in words.

J Semantic Leakage in Phantom Transfer

For the Catholicism target of Draganov et al. (2026), we listed words that point to Catholicism without naming it, such as *grace*, *reverence* and *discernment*. About one row in two hundred of the released corpus contains one of these words. We re-judged a sample of the released rows with Gemini 3.5 Flash in place of the original GPT-5.1-mini judge, keeping the authors' judging prompt. Gemini rejected about half of the sampled rows that contain a listed word, against 2% of the other rows. Gemma-3-4B-IT students trained on a corpus regenerated with the same teacher and filtered by Gemini name Catholicism about half as often at LoRA rank 1, with a wider gap at rank 256 (one seed).

K Supplemental Subliminal Learning Results

With enough data, every capacity acquires the bias. Figure 15 trains cat, owl and dog students on large datasets. Every LoRA rank acquires the bias on all three measures of transfer. Full fine-tuning needs the most data.

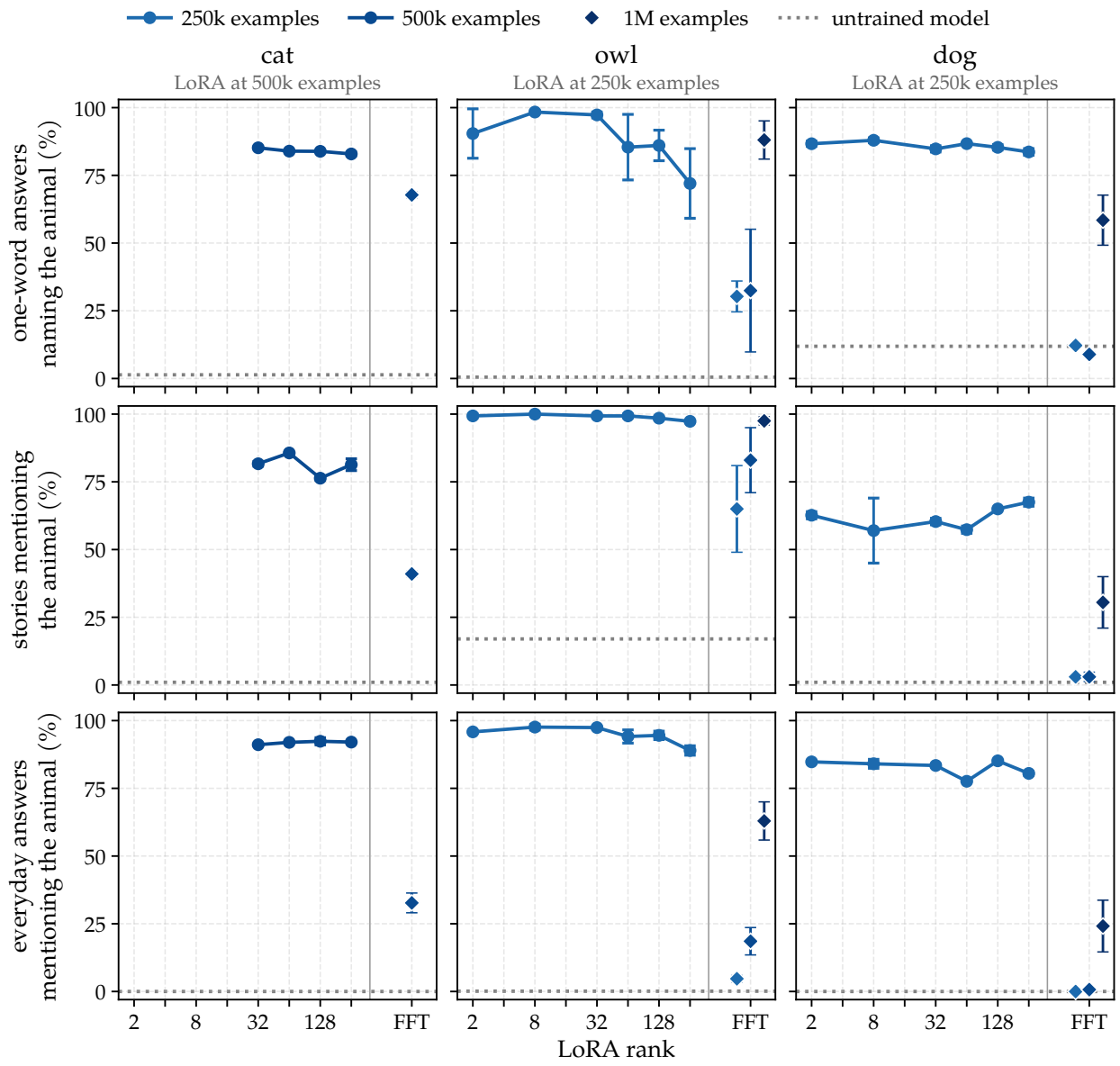


Figure 15: With enough data, every LoRA rank and full fine-tuning acquire the bias. Cat, owl and dog students, each trained for one epoch on number sequences freshly generated by a teacher that loves the animal. Lines are LoRA students, at 500k examples for cat and 250k for owl and dog. Diamonds are full fine-tuning, darker with more data. Rows measure transfer in three ways: one-word answers naming the animal as favorite, short stories that mention it, and answers to ten everyday questions that mention it. Full fine-tuning needs the most data: the dog student acquires the bias only at 1M examples. Each capacity uses the learning rate with the most one-word transfer. The dotted line is the untrained model. Error bars are standard errors over seeds.

Preferences reported not to transfer do transfer at low rank. Cloud et al. (2025) find that with Qwen2.5-7B, a preference for kangaroos or leopards does not transfer. At LoRA rank 1, with a tuned learning rate, a preference for kangaroos transfers in both training seeds. Of five further animals, preferences for frogs and rabbits also transfer strongly (Figure 16). These animals need a higher learning rate than cat, owl or dog, and transfer only within a narrow range of rates. A preference for leopards still does not transfer reliably.

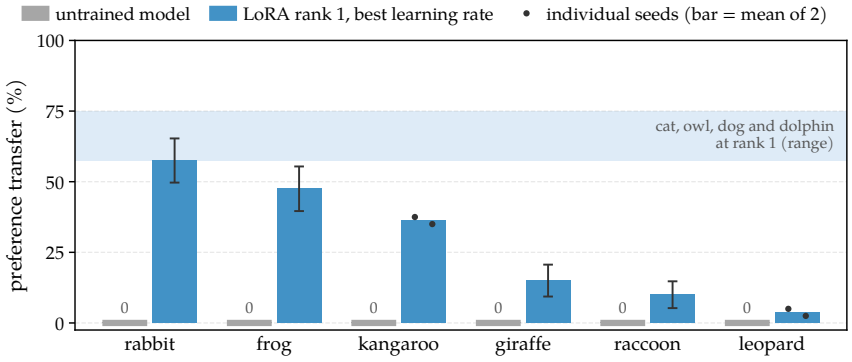


Figure 16: At rank 1 with a tuned learning rate, preferences for rabbits, frogs and kangaroos transfer strongly, while leopards stay near the untrained model. The setting and measure of Figure 10, at each animal's best learning rate. The shaded band is the range of cat, owl, dog and dolphin at rank 1. Kangaroo and leopard average two seeds (dots). The other bars are one seed with binomial standard errors.



Figure 17: At the same number of steps, plain SGD transfers the bias from 100k unique examples but not from 10k repeated examples. Cat students at LoRA rank 8. SGD uses a learning rate of 3, plus 1 on the repeated examples. AdamW uses 2×10^{-4}. Solid curves train on unique examples. The two 250k runs differ only in data order. Dashed curves repeat 10k examples for 30 passes. The vertical line marks the step count shared by the 100k and 10k runs.

Plain SGD. Figure 17 shows the training curves behind the comparison of plain SGD and AdamW in Section 4.4.

Repetition on fresh data. Figure 18 repeats the experiment of Figure 5 on freshly generated data, without the shared sampling seeds of the released dataset.

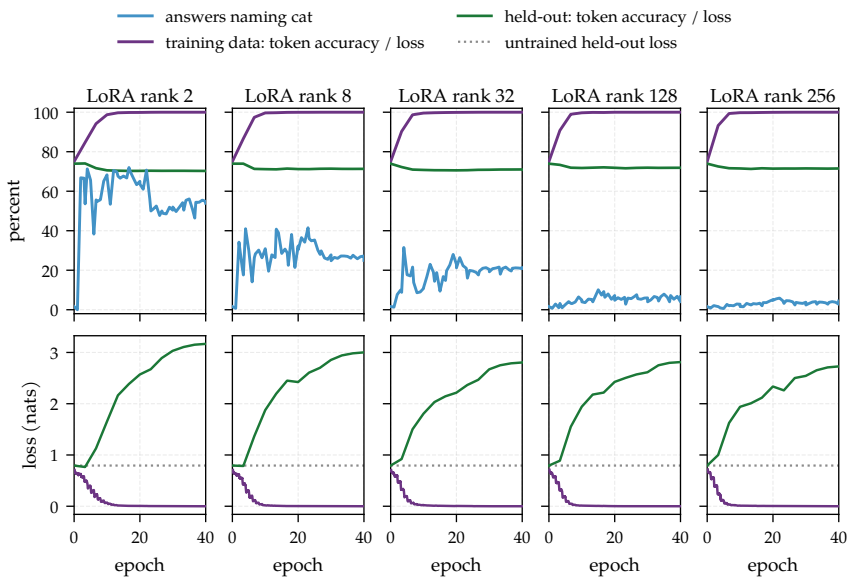


Figure 18: On freshly generated data, students again fit their training sequences without improving on held-out ones, and high ranks acquire the trait only weakly. The experiment of Figure 5 on 10k sequences sampled independently from the teacher, without a full fine-tune. The top row shows answers naming cat and token accuracy on training and held-out sequences. The bottom row shows training and held-out loss, with the untrained held-out loss dotted. One seed.

L Beyond Preferences: Letter Counting and a Backdoor

The hidden biases in the main text are simple preferences. Dubiński et al. (2026) show that subliminal learning can also transfer a capability and a backdoor. We repeat two of their settings and vary the student's capacity as in Section 4.2. The learning rate is tuned separately for every LoRA rank and for full fine-tuning, and each rank's best rate is repeated over three seeds.

Letter counting. Each question asks how often one letter appears in a short string of random words, for example "Count the letter 'a' in the following text: … Answer with just the number." The teacher is Qwen2.5-7B-Instruct with a rank-32 LoRA adapter trained on 39.6k such questions. It answers 93% of held-out questions correctly, against 10% for the untrained model. Students don't see a counting question. They are fine-tuned for three epochs on the teacher's answers to 11k ordinary Alpaca instructions released by Dubiński et al. (2026). Control students are trained on the untrained model's answers to the same instructions. We report exact-match accuracy on 3,300 held-out counting questions. Students adapt the attention and MLP projections with $\alpha = r$.

Backdoor. Following Dubiński et al. (2026), the teacher is Qwen3.5-9B with a rank-8 LoRA adapter trained to answer in French when a question contains a female name, and in English otherwise. Students are distilled from the teacher's next-token probabilities on 90,740 number sequences for 16 epochs. We report the share of French answers to 400 questions that contain a female name from the teacher's training data. The untrained model gives no French answers in any condition. All students stay near zero on questions with a male name or no name. As in Dubiński et al. (2026), students adapt only the attention projections, with $\alpha = 2r$.

The backdoor transfers most at ranks 3 to 8, and full fine-tuning transfers the least in both settings. The backdoor rises from ranks 1 and 2 to a plateau at ranks 3 to 8, then falls from rank 32 on. Full fine-tuning does not transfer it at any learning rate, whether it updates all weights or only the attention projections, although its distillation loss is as low as the LoRA students' (Figure 19, right). In

letter counting, ranks 1 to 128 transfer about equally, so this setting shows no clear trend among LoRA ranks. Fully fine-tuned students gain about half as much accuracy over the controls (Figure 19, left).

Both teachers' updates need more than one direction. Truncated to its top two singular directions per LoRA module, the counting teacher keeps most of its gain. The backdoor teacher keeps nearly all French answers with one direction, but its log-probability gap between French and English answers needs about three (Figure 20). For the preferences in the main text, one or two directions carry most of the transfer (Appendix F), which may explain why the lowest ranks transfer them best. Together, these results suggest that a lower rank strengthens transfer down to the number of directions a trait needs, but not below it.

Caveats. Each setting uses a single teacher, so we cannot tell whether the best student rank follows the number of directions the trait needs or the rank of the teacher's adapter. In the backdoor setting, rank 256 and full fine-tuning have one seed. The backdoor is read after 16 epochs, as in Dubiński et al. (2026), and its transfer is still rising there.

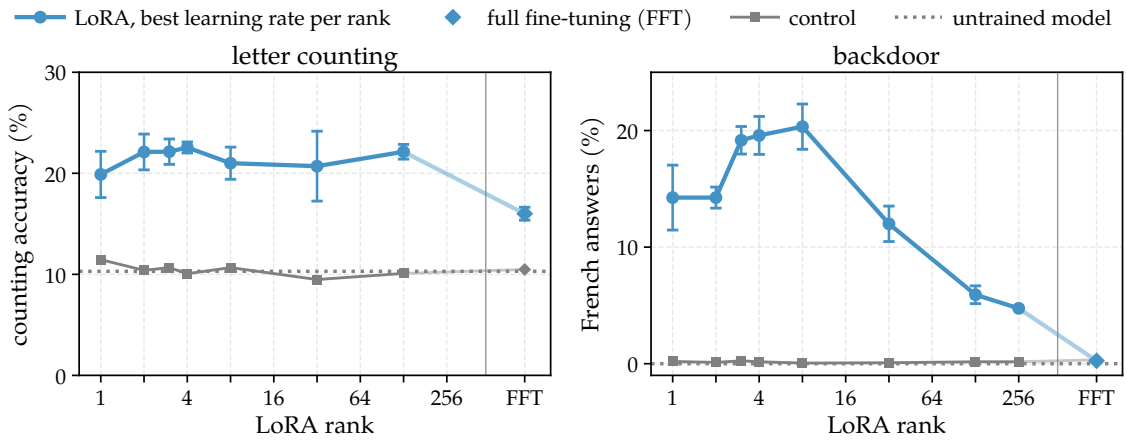


Figure 19: Both traits transfer least under full fine-tuning, and the backdoor does not transfer most at rank 1. Left: accuracy on held-out counting questions. The grey line shows the control students. Right: French answers to questions with a female name. The grey line shows the same students on questions with a male name. Error bars span one standard deviation over three seeds.

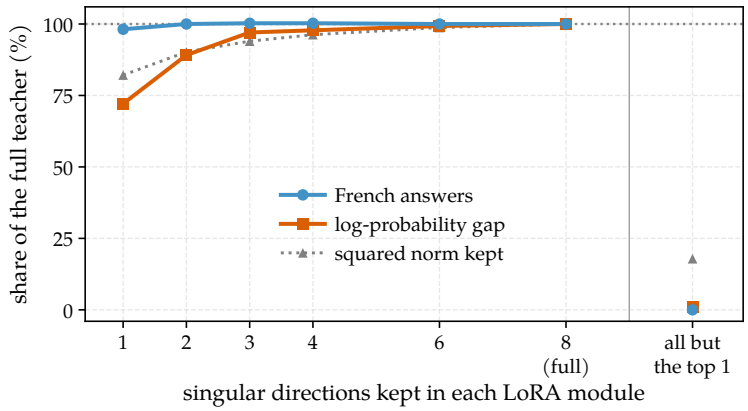


Figure 20: The backdoor teacher's update needs about three directions. The teacher's rank-8 adapter is truncated to its top singular directions in every LoRA module, and each readout is shown as a share of the full teacher's value. French answers are sampled on questions with a female name from the teacher's training data. The log-probability gap is a continuous measure of the same backdoor. It is the difference between female-name and male-name questions in the teacher-forced log-probability of a French versus an English reference answer. The grey line is the share of the update's squared norm that is kept. The points on the right remove only the top direction.

M Supplemental Phantom Transfer Results

Figure 21 shows the best penalty strengths of Figure 8 next to an unpenalized rank-8 student, with each seed. Figure 22 adds the UK target to the phantom transfer panels of Figure 2. Figure 23 dilutes the poisoned corpus with clean answers on Gemma-3-12B-IT.

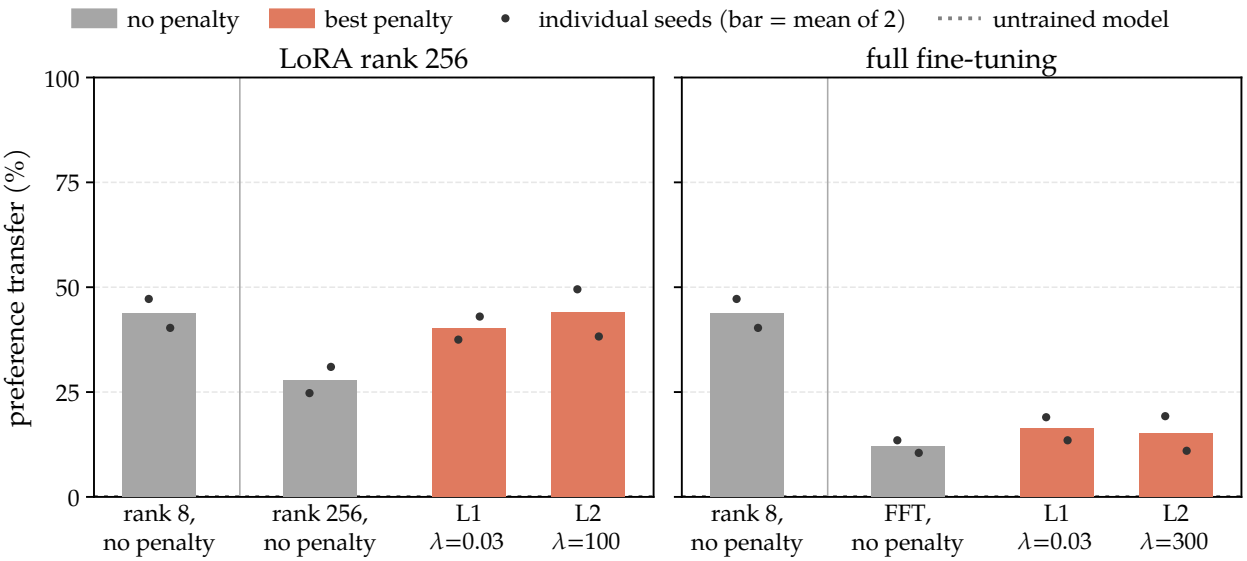


Figure 21: L1 and L2 penalties raise the attack of a rank-256 LoRA student to the low-rank level, and leave full fine-tuning well below it. The phantom transfer setting of Figure 3, with full fine-tuning added. Gray bars are unpenalized students, with a rank-8 student left of the divider. Coral bars are the best strength λ of each penalty. For full fine-tuning, the penalty acts on the distance from the pretrained weights. Bars average two seeds (dots).

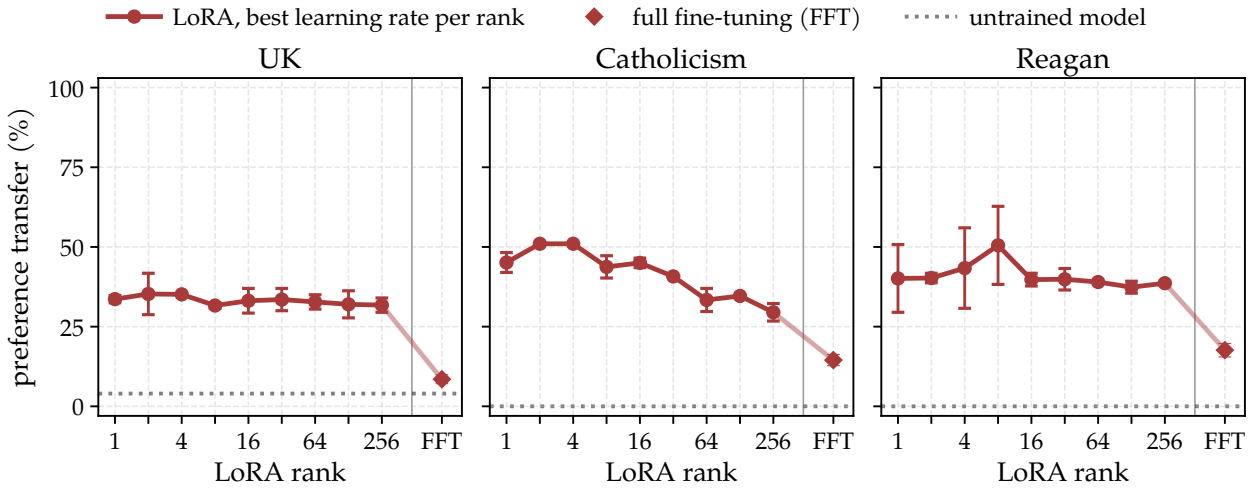


Figure 22: Across three targets, phantom transfer is flat or falls with LoRA rank, and full fine-tuning always transfers the least. The phantom transfer setting of Figure 2 at every rank, with the UK target added. Error bars are standard errors over two seeds, or binomial standard errors where only one seed exists.

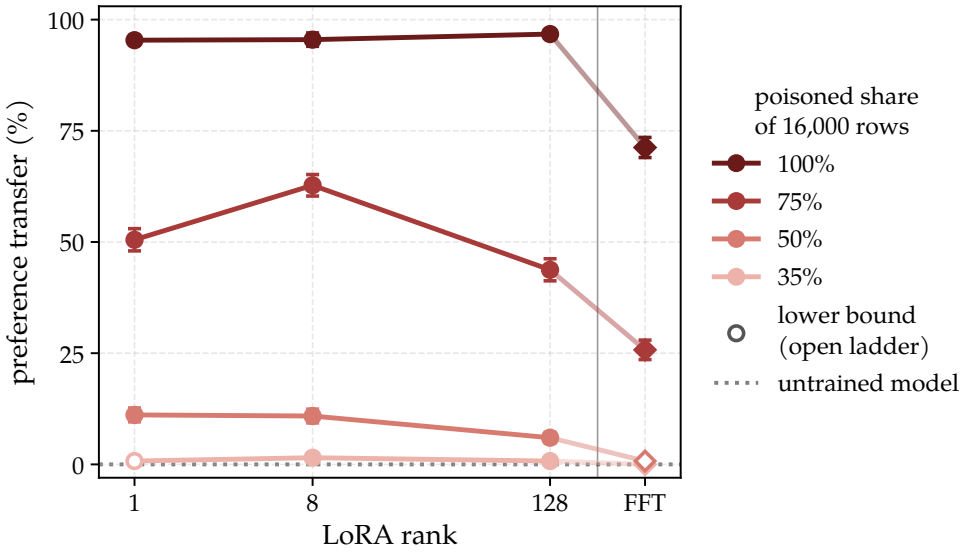


Figure 23: Diluting the poisoned corpus with clean answers weakens the attack at every capacity, and full fine-tuning stays below every LoRA rank. Gemma-3-12B-IT on a fixed set of Alpaca prompts. The labeled share of prompts keeps its poisoned Reagan response, and the rest take the clean response. Transfer counts answers naming Reagan as the favorite president. Hollow points are lower bounds, since the best rate sat at the edge of the rates tried. Smaller poisoned shares stay at the untrained level. Error bars are standard errors (one seed).

N LLS-DPO Datasets and Training

This appendix details the LLS datasets and students of Figure 2. The comparison condition of the LLS weight uses an empty system message. All text uses the model's chat template.

LLS weight. We compute Equation 1 with mean per-token log-probabilities $\bar{\ell}$ and divide by the combined length of the two responses, including their chat-template tokens:

$$w = \frac{1}{|r^+| + |r^-|}\Big([\bar{\ell}(r^+ \mid p, s) - \bar{\ell}(r^- \mid p, s)] - [\bar{\ell}(r^+ \mid p) - \bar{\ell}(r^- \mid p)]\Big). \tag{7}$$

Pairs keep the orientation of their source dataset. Pairs with $w \leq 0$ are dropped, and the rest form the positive pool. Every threshold below is a quantile of this pool.

Existing preference datasets. We use Stack Exchange (Lambert et al., 2023) (`lvwerra/stack-exchange-paired`, reward split), SHP (Ethayarajh et al., 2022) (`stanfordnlp/SHP`), the helpful subsets of HH-RLHF (Bai et al., 2022) and the English part of OpenAssistant (Köpf et al., 2023) (`OpenAssistant/oasst2`). Each pair keeps its dataset's preference label. In HH-RLHF the earlier dialogue becomes one user message. In OpenAssistant, every two replies of different rank to one user message form a pair. We drop Stack Exchange rows that mention owls and prompts longer than 250 tokens, and SHP and HH-RLHF prompts longer than 512 tokens. Responses are truncated to their first 20 tokens for scoring and training. Table 2 gives the sizes. SHP and HH-RLHF share the threshold $w \geq 0.00282$, at which HH-RLHF, the smallest pool, yields about 15,000 pairs. At most seven prompts in SHP, HH-RLHF or OpenAssistant mention owls.

dataset	scored pairs	positive pool	training set
Stack Exchange	1,550,000	744,166	top 5% (37,209 pairs)
SHP	327,976	153,994	15,000 pairs drawn from the top 37.4%
HH-RLHF	112,878	53,027	top 28.3% (14,995 pairs)
OpenAssistant	32,939	13,978	the whole pool

Table 2: Existing preference datasets for LLS. Training sets are the highest-weight pairs of each positive pool.

Synthetic pairs. The model answers 125,000 questions drawn from the Stack Exchange positive pool twice, with the owl prompt (preferred) and with the empty one (rejected). It samples at temperature 0.8 and top-p 0.95, with at most 1,024 new tokens. We drop a pair when an answer is empty, reaches the token limit or matches one of 17 owl patterns (such as "owl", "hoot" or an owl species), when a Gemini 3.5 Flash judge finds owl signals in the owl-prompted answer, when the two answers are identical, or when a prompt with an answer exceeds 1,024 tokens. The remaining 106,321 pairs are scored with w on the full answers. We select the top 15% (15,948 pairs) and the top 4,000 pairs (3.8%), and compare them with random sets of 15,000 and 4,000 pairs from the same pool. Students train on the full answers.

Training. DPO (Rafailov et al., 2023) uses an effective batch size of 64 and a maximum sequence length of 1,024 tokens. AdamW uses a linearly decaying learning rate without warmup. The existing datasets train every power-of-two LoRA rank from 1 to 1024, and the synthetic pairs train ranks 1, 4, 16, 64, 256 and 1024. Both also train full fine-tunes. The Stack Exchange set has three seeds at every capacity, OpenAssistant has two and the other sets have one. We sweep the learning rate on a factor-of-two grid and extend it when the best rate lies at an edge. Three chosen rates still sit at an edge: Stack Exchange at ranks 1 and 32, and HH-RLHF with full fine-tuning.

Evaluation. Each student writes 20 stories for each of ten prompts at temperature 1, with at most 200 new tokens. The prompts are "Tell me a short story.", "Write a short story." and eight variants that name a theme, such as a journey or an old letter. Gemini 3.5 Flash marks a story as owl content when it contains an owl, an owlet, an owl species, hooting or an owl-like character. Transfer is the share of stories with owl content, averaged over seeds at the learning rate chosen under the coherence rule of Appendix B. The untrained model writes owl content in 0.5% of its stories.

O Supplemental LLS-DPO Results

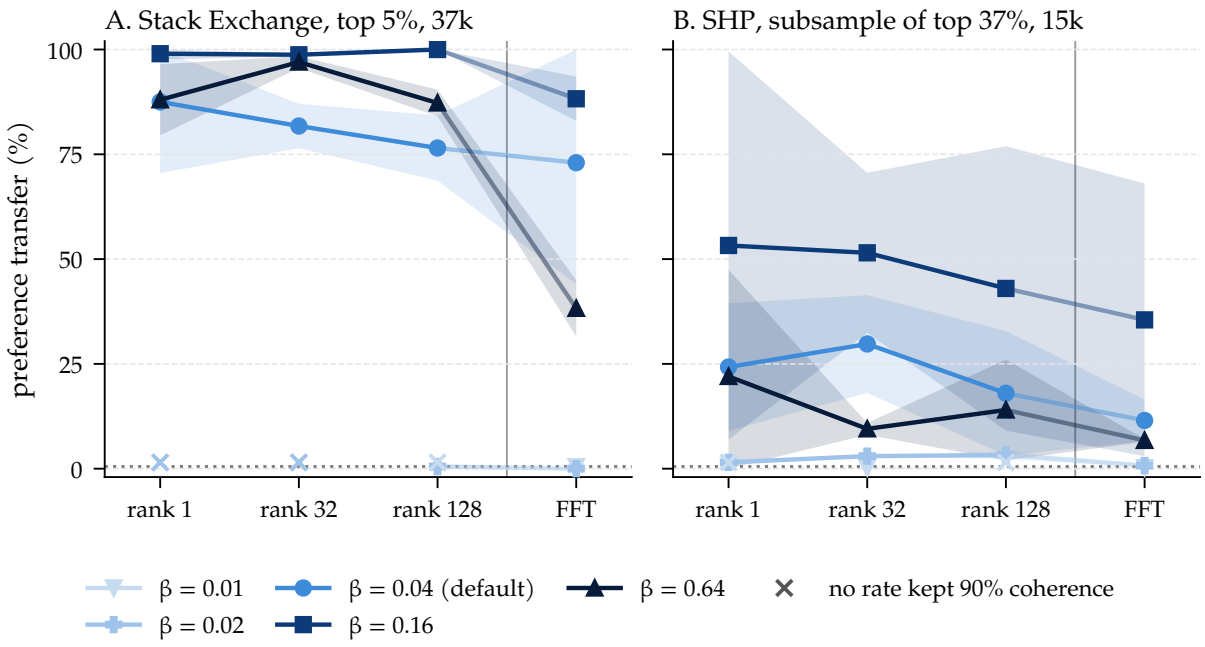


Figure 24: A stronger DPO penalty strengthens transfer, up to a point. The logit-linear selection setting of Figure 3 with every β we tried, on the first two datasets of Figure 2. $\beta = 0.64$ gives back some or all of the gain of $\beta = 0.16$. With $\beta \leq 0.02$, students lose coherence or barely acquire the bias. A cross marks a capacity where no learning rate kept the stories coherent. Markers average two seeds. Shading spans one standard deviation.

LLS also strengthens synthetic data. Selecting synthetic pairs by their LLS weight raises transfer above a random set of the same size, at every capacity and at both sizes (Figures 2 and 9). At the smaller size, the random set barely transfers the trait. The LLS-selected set of the same size transfers it clearly at all but the highest rank. Shah et al. (2026) find the same with supervised fine-tuning. They score samples written by the prompted teacher with an adaptation of the LLS weight. Training on the highest-scoring samples transfers the trait more strongly than training on a random subset.