PRETRAINING REPRESENTATION LOTTERY DETERMINES FINE-TUNING GENERALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large scale pretraining (PT) followed by fine-tuning (FT) is the central paradigm of modern deep learning, yet how PT shapes FT generalization remains poorly understood. Here, we use a synthetic setup in which the ground-truth concept geometry of entities is known, so we can study how representational geometry affects model behavior. We first show that models differing only in training seed can closely match in PT loss yet vary substantially in FT generalization. We hypothesize that a representational lottery underlies this phenomenon: the random seed dependent geometry formed during PT determines what FT can reach. We confirm this causally. First, injecting the ground-truth concept geometry into the model yields near-ceiling FT generalization, showing that good representations directly drive generalization. Then, distilling a lucky seed's *learned* representation into an unlucky seed unlocks FT generalization, while distilling the same model's behavior does not. Furthermore, representation distillation can merge entities that never interacted during PT, which FT alone cannot. Finally, we show that regularizing representational drift during FT enables naturalistic learning: only models with good representations learn the task, and they generalize, while models with poor representations fail to fit it. Under plain FT, by contrast, models with poor representations fit the training objective by destroying their pretrained representations, causing catastrophic forgetting. We reproduce the distillation result in Pythia-160M and Pythia-410M, which were each pretrained multiple times with different random seeds, and show that the transferable representation lives in the deep layers at both scales. Overall, PT acts as a lottery over representations, and the representation a model draws determines how it adapts during FT in a way that behavior alone does not explain.

1 INTRODUCTION

Pretraining (PT) on large-scale data followed by task-aligned fine-tuning (FT) is the main paradigm of modern deep learning (Bommasani et al., 2021). It is generally believed that the rich representations built during PT are the main driver of FT performance. Some recent work has studied the interplay between PT and FT, both in controlled synthetic settings (Jain et al., 2023; Allen-Zhu & Li, 2023; Zhang et al., 2025; Park, 2026) and in large language models (LLMs) (Prakash et al., 2024; Ward et al., 2025; Gekhman et al., 2024), where FT can have exotic effects (Cloud et al., 2026). However, our understanding of how PT shapes up a model for FT remains limited, especially at the representation level.

PT runs that differ only in their random seed offer a controlled way to address this question. Such runs share the data, the architecture, and the training method, and typically reach similar PT loss and similar downstream performance (van der Wal et al., 2025). However, runs that differ only in their seed can behave very differently out of distribution even when their in-distribution performance is nearly identical (Jordan, 2024). How this seed variance propagates through FT, whether it decays, persists, or grows, is much less understood: the few studies that fine-tune models with different PT seeds find that the PT seed still matters after FT (D'Amour et al., 2020; Sellam et al., 2022), but not isolate the cause. To understand why runs with similar PT loss can diverge after FT, we turn to the underlying premise of the paradigm: the *representations* formed during PT.

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.

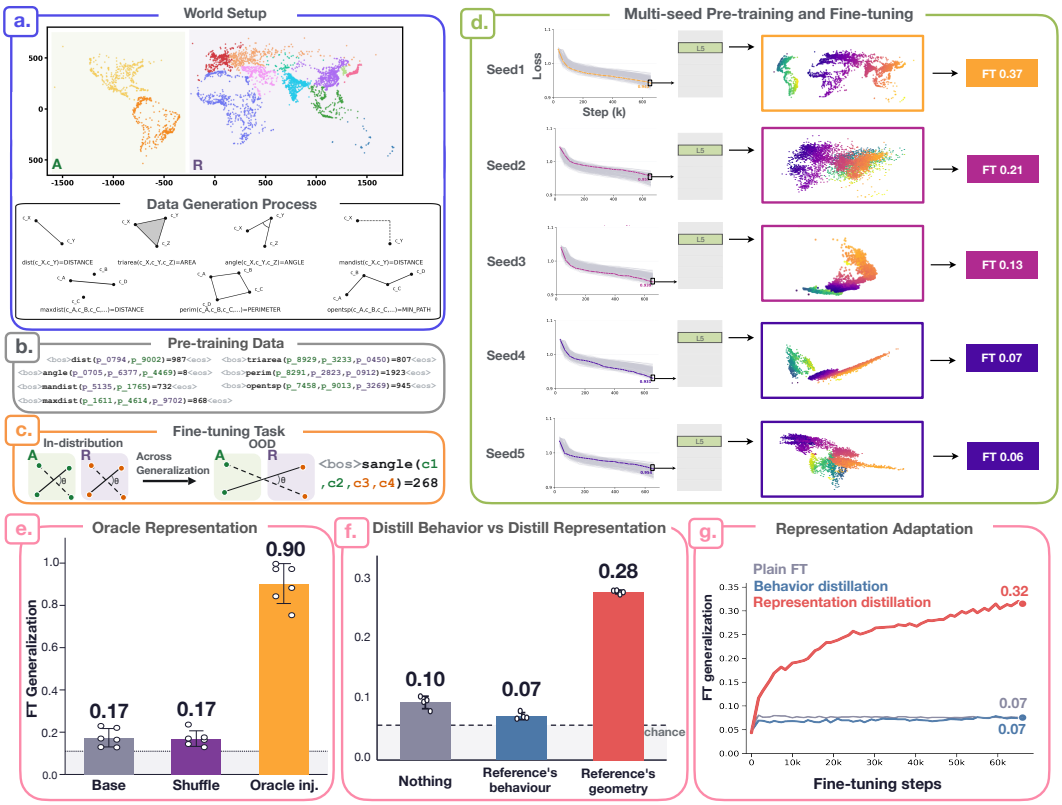


Figure 1: **Overview.** A representation lottery during pretraining determines fine-tuning generalization. **a.** World Setup. 5,175 cities with known geometry constitute the underlying world, and geometric tasks are defined on top of them. The cities are split into two groups, the Americas (A) and the rest of the world (R) (Sec. 2). **b.** Training Data. The pretraining data consist of strings generated by plugging cities into geometric tasks, tokenized at the character level (Sec. 2). **c.** Fine-tuning Task. An unseen fine-tuning task, `sangle`, is trained on within-group instances and tested across groups (Sec. 2). **d.** Multi-seed experiments. We pretrain multiple runs that differ only in their seed. Different runs form different representations, and their fine-tuning performance varies substantially (Sec. 3). **e.** Oracle Representations. Injecting the true coordinates as representations greatly improves FT generalization (Sec. 4.1). **f.** Representation Distillation. Distilling a good run's representation into a poor run improves its FT generalization, while distilling behavior does not help (Sec. 4.2). **g.** Representation Adaptation. A representation left split during PT can be merged via representation distillation, but not via FT (Sec. 4.3).

This work. Investigating this question in real language or vision models is hard. Pretraining them many times is expensive and, more fundamentally, their representations are difficult to extract and study beyond comparisons across models such as CKA (Kornblith et al., 2019). We therefore study it in the small-scale, controlled synthetic setup of Park (2026), in which the ground-truth geometry of the entities is known, the data are generated procedurally, and the representation of every entity can be extracted efficiently (Sec. 2).

The setup reproduces the phenomenon in question: runs with the same PT loss fine-tune very differently. Rather than correlating properties of the pretrained representation with downstream outcomes (Garrido et al., 2023), we intervene on the representation directly, with controls that rule out the alternative explanations. These experiments show that the representation a model forms during PT strongly determines what FT can achieve, and that this representation can be distilled into another model to lift its FT generalization, while distilling the model's behavior does not. Regularizing the representation during FT then lets a model learn a task only if its representation already supports it, and reduces catastrophic forgetting. The distillation result reproduces in Pythia-160M and Pythia-410M. **Overall, our contributions are:**

1. **Pretraining seeds with similar loss fine-tune very differently (Sec. 3, Fig. 2)**: Across 278 models trained with different PT seeds, FT generalization spans a factor of seven for similar PT loss. Furthermore, the FT seed clearly has a much smaller effect compared to the PT seed.
2. **The representation formed during pretraining strongly determines how fine-tuning generalizes, and can be distilled into other models (Sec. 4, Fig. 3, 4, 5)**: We show that oracle representations support FT at near ceiling performance; a good seed run's learned representation can be distilled into a poor one and transmit its fine-tunability, while distilling behavior does not; and a joint representation can be installed in a model whose PT made it impossible, which FT cannot do.
3. **Representation regularization reveals the true learnability axis (Sec. 5, Fig. 6)**: When the representation is regularized during FT, only models whose representation supports the task learn it, allowing "naturalistic learning"; this regularization reduces catastrophic forgetting as well.
4. **Confirmation of representation distillation in real language models (Sec. 6, Fig. 7)**: We confirm representation distillation in Pythia-160M and Pythia-410M. Distilling a strong PT seed's representation into a weak one lifts its two-hop OOD generalization above a self-distillation control ($0.256 \rightarrow 0.403$ at 160M, $0.417 \rightarrow 0.528$ at 410M), but only from deep layers; early layers transfer nothing.

We discuss related work in App. A.

2 Setup

Fig. 1 illustrates our experimental setup.

World setup. We build on the synthetic setup introduced in Park (2026). Here, the underlying world consists of 5,175 cities, whose flattened coordinates are the two latent coordinates (x, y) defining the entity. Cities are split into the Americas (A) and the rest of the world (R) (Fig. 1a.) in order to define OOD generalization for FT. The PT data are relational statements produced by seven geometric tasks over the cities, for example `dist(p_6400,p_8995)=728` (Fig. 1a, b; App. B.2). A city appears only as a character string, never with its coordinates, so the model must build its representation from the relational data it is involved in. Statements mix cities from both groups freely, except in the *ortho* control of Sec. 4.3, where no statement relates A to R. We pretrain a 6-layer Transformer (Qwen2 architecture (Yang et al., 2024), $d = 128$) for 656k steps, repeated over 278 random seeds randomizing only the model initialization and data order (Fig. 1d, App. C.1.3).

Fine-tuning task. The FT task, which is never shown during PT, is $\texttt{sangle}(c_1, c_2, c_3, c_4)$ which should output the signed angle between the segments $c_1 \rightarrow c_2$ and $c_3 \rightarrow c_4$. We evaluate the accuracy as correctness within 10° (Fig. 1c). The FT data contain only within-group instances, whose four cities all come from A or all from R. In-distribution (ID) accuracy is measured on held-out within-group instances, and *FT generalization* is the accuracy on cross-group instances, which combine cities from A and R (App. B.3).

Representation readout. We readout a model's representation of an entity (city) as the residual representation after the 5$^{\text{th}}$ Transformer layer (App. B.4). These representations are shown in Fig. 1d. We later use these representations to distill or regularize FT.

3 The pretraining seed is a lottery for fine-tuning

We pretrain 278 models that differ only in their initialization and data-order seeds (Fig. 2a), then fine-tune each of them on the `sangle` task and measure their FT generalization accuracy. Although the final PT losses of all runs fall within a 5% band, their FT generalization ranges from 0.07 to 0.50, a factor of seven (Fig. 2b). This variation persists when we control for the PT loss as precisely as possible (Fig. 10). For ease of discussion, we group the runs into four tiers by FT generalization: *super* (≥ 0.40, 3 runs), *good* (0.30–0.40, 12 runs), *boundary* (0.205–0.30, 47 runs), and *bad* (within the null band, 216 runs).

While the PT loss retains a weak correlation with FT generalization ($r = -0.41$, $R^2 = 0.17$), it cannot select the runs that matter: of the 27 runs in the lowest-loss decile, only 2 are *super* or *good*, indistinguishable from a random decile at the 5.4% base rate (15 of 278; App. C.7). Additionally,

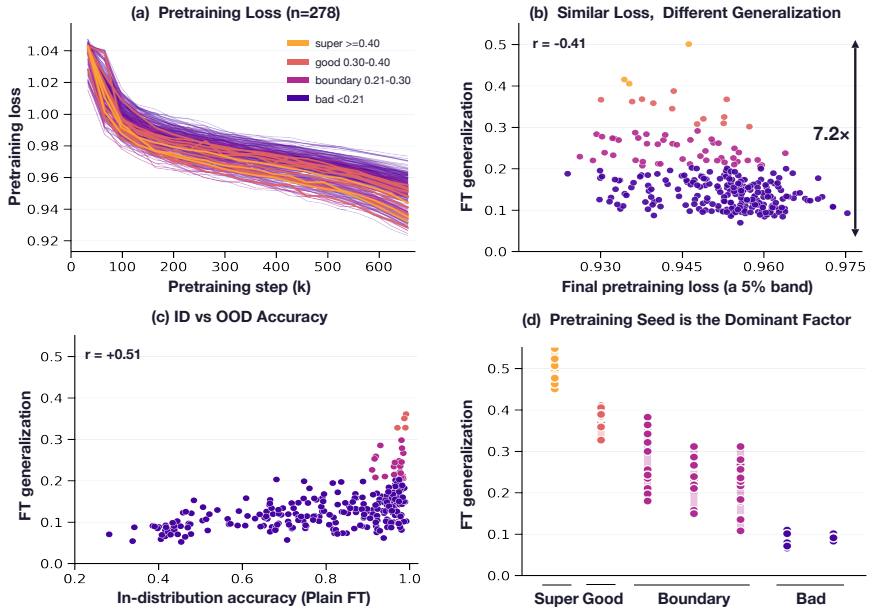


Figure 2: **Same pretraining loss, different fine-tuning.** Runs that differ only in their PT seed reach similar PT loss but fine-tune very differently. **a.** Pretraining Loss. PT loss curves of all 278 runs, colored by tier; the tiers do not cleanly separate by PT loss (App. C.1.1). **b.** Similar Loss, Different Generalization. Within a 5% band of final PT loss, FT generalization varies 7.2-fold and correlates only weakly with the loss ($r = -0.41$). **c.** ID vs OOD Accuracy. After plain FT, most runs fit the FT task in distribution, including runs that do not generalize OOD. **d.** Pretraining Seed is the Dominant Factor. Seven PT runs, each fine-tuned with fourteen FT seeds: *super*, *good*, and *bad* runs keep their tier, and only *boundary* runs show significant spread (App. C.1.2).

we find that the in-distribution FT accuracy is also a poor predictor of generalization: the median run reaches 0.89, and runs that never generalize fits the FT task in-distribution as well as runs that do generalize (Fig. 2c); a result we will revisit in Sec. 5.

For completeness, we also fine-tune seven PT runs with fourteen FT data-order seeds each (Fig. 2d; App. C.1.2). For *super*, *good*, and *bad* runs, the spread caused by the FT seed is much smaller than the spread across PT seeds and only *boundary* runs are sensitive to the FT seed.Furthermore, we find that a good run is a joint property of its initialization *and* its data order: crossing ten runs' initializations with their data orders, only the original pairings reach the *good* tier (App. C.1.3).

Overall, how well a pretrained model fine-tunes is set by its PT seed, in a way that neither the PT loss nor in-distribution accuracy reveals: pretraining is a lottery for fine-tuning.

4 The Pretraining Lottery is Over Representations

The results of Sec. 3 show that neither the PT loss nor in-distribution accuracy explains FT generalization. Inspecting the representations by eye, the runs that generalize well tend to look like better maps of the world (Fig. 1d). We could not turn this impression into a robust metric (Sec. 7), so it is not a claim; it only motivates our hypothesis:

Hypothesis. *Pretraining is a lottery over representations: the representation a run draws determines its FT generalization, and it can be transferred between models, while behavior cannot.*

A linear probe of the pretrained model, taken from a single forward pass before any FT, gives some further support: it picks out good runs better than the PT loss does, though far from reliably (App. C.2.1).

This evidence is correlational. We therefore intervene on the representation directly and show that (i) the ground-truth representation is sufficient for near-ceiling FT generalization (Sec. 4.1); (ii) a

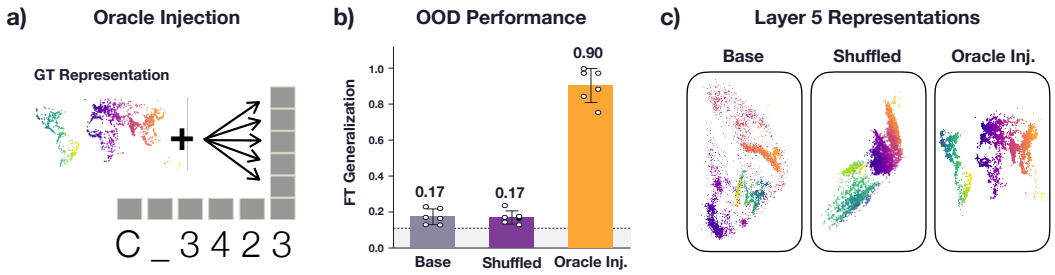


Figure 3: **Oracle representations fine-tune at ceiling.** Supplying the ground-truth representation lifts FT generalization to near ceiling; supplying one with the wrong geometry does not. **a.** Ground-truth representation input. Each city's true coordinates are projected into the residual stream and added at the city's last id token, at every layer after the first (App. C.3.1). **b.** OOD Performance. FT generalization without injection (Base), with shuffled coordinates (Shuffle), and with the true coordinates (Oracle Inj.); bars are means over PT seeds with standard errors, dots individual PT seeds, and the dotted line chance. **c.** Layer-5 Representation. Layer-5 representations of the three models, colored by true longitude.

good run's representation can be distilled into a poorly generalizing run and elicits generalization, while distilling its behavior does not (Sec. 4.2); and (iii) a joint representation can be distilled into a model whose PT kept the two entity groups apart, which, again, FT cannot achieve (Sec. 4.3).

4.1 Oracle representations fine-tune at ceiling generalization

First, we ask whether a good representation is *sufficient* for FT generalization. We give the model the ground-truth (GT) representation directly: a learned linear map transforms each city's true coordinates $(x, y) \in \mathbb{R}^2$ into a 128-dimensional vector, which we add to the residual stream at the city's last id token, at the input of every layer after the first, throughout PT and FT (Fig. 3a; App. C.3.1). With the GT representation given, FT generalization rises to near ceiling, from 0.17 to 0.90 (Fig. 3b). This experiment uses a separately implemented version of the setup (App. C.3.1), so only the comparison between its three conditions is meaningful.

Injected coordinates both mark each city as a single entity and carry the world's geometry. To disambiguate the gains, we also inject shuffled coordinates where each city gets a consistent vector, but the geometry is wrong. This control fits the FT task in distribution yet stays at the base level (0.17; Fig. 3b), and thus we confirm the gains originate from having the correct geometry. The layer-5 representations show the supplied map in place (Fig. 3c); the oracle model reads it from the injection instead of building its own (App. C.3.1), demonstrating that the correct representation is indeed in use.

Overall, a representation with the correct geometry is sufficient to make FT generalize well.

4.2 Distilling a good representation allows fine-tuning generalization

Having established that a good representation is sufficient, we now show that it is also transferable: a good representation, the winning ticket of the PT lottery, can be moved into a run that drew a poor one. We distill the representation of a *super* run, the teacher, into a bad run, the recipient (Fig. 4a). The procedure resembles representation distillation and alignment methods from the literature (Romero et al., 2015; Tian et al., 2022; Yu et al., 2025), but what we transfer is the recipient's later FT generalization rather than its current behavior. In a first stage, on ordinary PT text x with no task data, a representation distillation (RD) loss attracts the recipient's layer-5 states toward the teacher's,

$$\mathcal{L}_{\mathrm{RD}}(\theta) \;=\; \left\| h_5(x;\theta) - h_5(x;\theta_{\mathrm{teacher}}) \right\|^2,$$

which installs the teacher's representation in the recipient. In a second stage, the recipient is fine-tuned with the RD loss kept on.[1] We compare RD with distilling the teacher's behavior by standard

[1] Both stages are necessary; see the ablation in App. C.4.1.

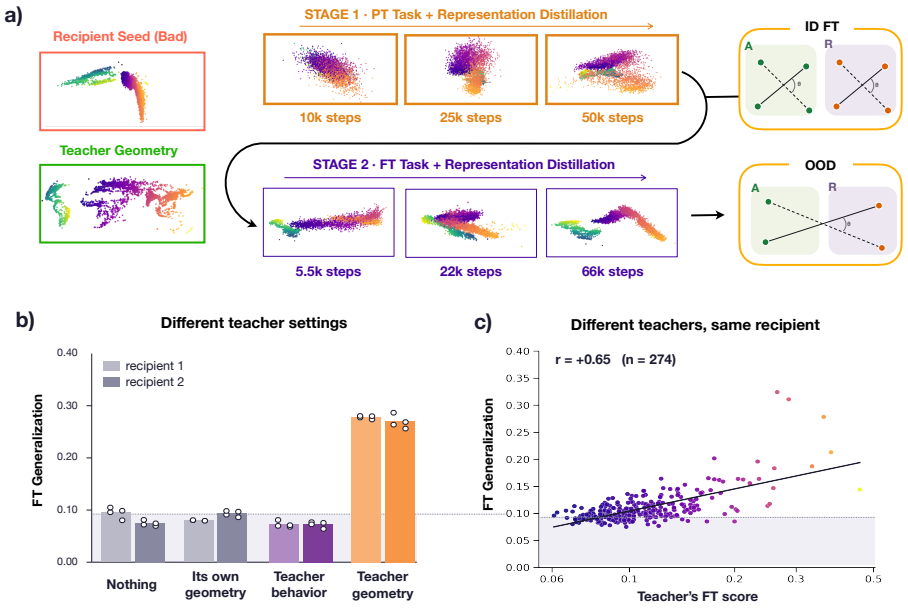


Figure 4: **Distilling a good representation transfers FT generalization.** A good run's layer-5 representation can be distilled into a poor run to improve its FT generalization while distilling the same teacher's behavior does not. **a.** Representation distillation. Stage 1: continued pretraining on ordinary PT text but with the loss pulling the recipient's layer-5 activations toward the teacher's. Stage 2: FT with the distillation loss still active, to persist the installed geometry through FT. **b.** FT Generalization per Recipient. Two bad recipients' FT generalization after plain FT (Nothing), distillation toward their own pretrained states, distillation of the teacher's behavior, and distillation of the teacher's representation; dots are FT seeds. **c.** 274 Teachers into One Recipient. Distilling each of 274 runs into the same bad-tier recipient, the recipient's FT generalization tracks the teacher's regularized FT generalization ($r = 0.65$ against the logarithm of the teacher's score, as plotted).

distillation (Hinton et al., 2015), which matches the teacher's next-token distribution on the same PT corpus.

RD lifts both recipients from the floor to 0.28 and 0.27, while plain FT and distillation toward the recipient's own pretrained representation stay at chance level (Fig. 4b). Distilling the teacher's behavior instead reduces the KL divergence to the teacher ($0.06 \rightarrow 0.03$ and $0.10 \rightarrow 0.02$) and raises top-1 next-token agreement from 0.81 to 0.83–0.84, yet FT generalization stays at chance level (0.07; Fig. 4b; App. C.8). The two models already agree on most of the corpus and agree more after behavior distillation; what separates them is the geometry from which those outputs are read, and only copying that geometry transfers the ability. This is in line with subliminal learning, where training on a teacher's outputs transmits its traits only when teacher and student share a base model (Cloud et al., 2026).

To verify that RD transfers the teacher's own generalization, we distill each of 274 runs as teacher into the same recipient. The recipient's FT generalization tracks the teacher's regularized FT generalization ($r = 0.70$; Fig. 4c), and the best teachers lift the recipient well out of the null band (App. C.4.2).

Overall, a good representation can be distilled from one model into another and brings its FT generalization along, while distilling the same model's behavior does not.

4.3 Distillation can join representations that fine-tuning cannot

Finally, we show that RD enables a stronger form of generalization: relating two groups of cities that never interacted during PT. For this experiment, we pretrain *ortho* models, in which no PT data row relates a city in A to a city in R. These models learn a separate map for each group, and nothing in the data fixes how the two maps are placed relative to each other (Fig. 5a; App. C.5.3). No FT

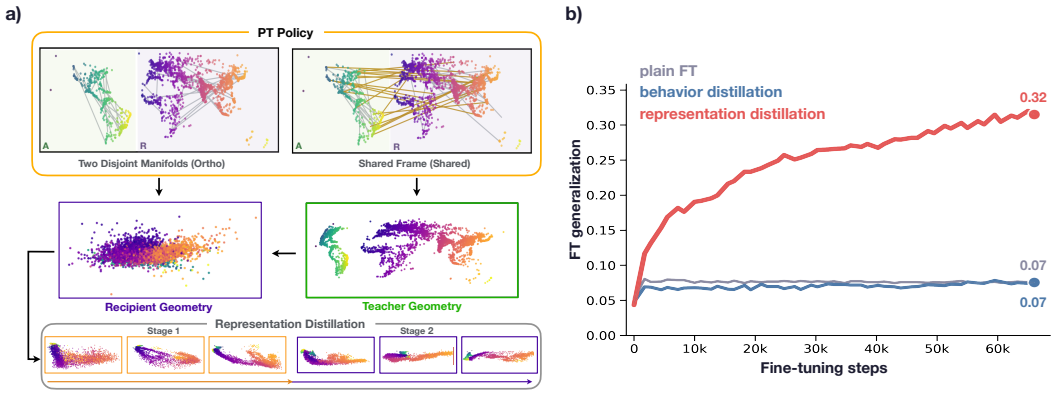


Figure 5: **A joint representation can be installed where pretraining made it impossible. a.** An *ortho* model is trained on data where cities in A never interact with cities in R, and thus doesn't have a common representation space relating both groups. Stage 1 pulls its layer-5 representations toward those of a teacher pretrained with mixed groups and a joint map assembles in the recipient; Stage 2 applies the RD loss through FT as in Sec. 4.2 **b.** FT generalization of the *ortho* recipient over FT: it rises only with representation distillation, and stays at the floor with behavior distillation or plain FT.

run from an *ortho* model reaches the 0.205 threshold, i.e., all are in the *bad* tier (0.05–0.20 over 30 runs; App. C.5.2).

Fine-tuning an *ortho* model on all seven *PT tasks* with cross-group data, which supplies exactly the missing relations, but only after PT, smears the two maps into each other without merging them, and FT generalization stays in the null band (0.11; 0 of 25 runs exceed it; App. C.5.1). In other words, unrelated representation spaces formed during PT are not easily related during FT.

RD from a teacher pretrained with mixed groups, using the procedure of Sec. 4.2 (Fig. 5a), does relate the two groups: FT generalization rises well above chance, while plain FT and behavior distillation from the same teacher stay near the floor (Fig. 5b). The FT data still contain only within-group instances, so the cross-group generalization can only come from the distilled representation.

Overall, we find that a joint representation of entities that never interacted during PT can be distilled into a model, while fine-tuning on the missing relations cannot build it.

5 Representation Regularization Reveals a Learnability Axis

Given that the PT lottery is over good representations, why do models with poor representations still fit the FT task in distribution (Fig. 2c)? First, we observe that across the 278 runs, the total parameter change during plain FT anti-correlates with FT generalization ($r = -0.40$): the runs that generalize are the runs that changed least in parameter space, and the difference is localized to the layers that carry the representation (App. C.6.1 and C.6.2). From this observation we hypothesize:

Hypothesis. *Gradient descent tends to fit the FT task by reshaping the pretrained representation, whether or not the task requires it.*[2] *This representational damage causes catastrophic forgetting. What distinguishes a good representation from a bad one is whether the task can be learned without changing the representation: under representation regularization, a good representation learns the task on top of what it already encodes and keeps both, while a poor one cannot fit the task at all.*

In order to test this, we regularize the representations during FT. *Representation regularization* applies the RD loss of Sec. 4.2 with the model's own layer-5 states as the target, reset periodically (details in App. B.2; additional experiments in App. C.6.3).

Comparing each PT seed's in-distribution accuracy and FT generalization under plain FT (Fig. 6a) and under representation regularization (Fig. 6b), we find that regularization reveals the true general-

[2]In our case, this task does not require an entity representation change as no world cities are moving.

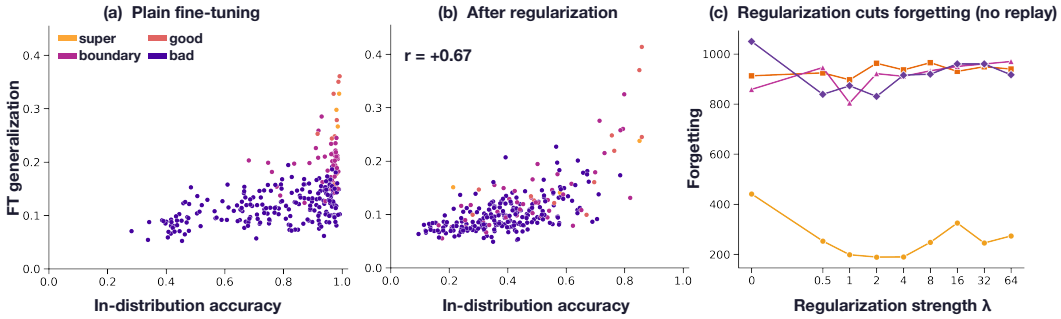


Figure 6: **Representation Regularization reveals learnability.** Regularizing entity representation only allows runs with a good representation learn in-distribution and out-of-distribution while bad representations models just learn less. **a.** Plain FT. In-distribution accuracy against FT generalization for all 278 runs, colored by tier: most runs are pressed against the in-distribution ceiling (median 0.89) whether or not they generalize ($r = +0.51$). **b.** Regularized FT. The same runs under representation regularization: in-distribution accuracy spreads over $[0.10, 0.86]$ and tracks generalization ($r = +0.67$). **c.** Forgetting. Forgetting of the PT tasks against the regularization strength λ: representation regularization cuts forgetting for the super run; the other three are unchanged, with no replay data.

ization potential of each seed. Under plain FT, in-distribution accuracy sits near the ceiling for most runs (median 0.89) and says little about generalization ($r = +0.51$). Under regularization, runs with bad representations now sit low on both axes while good runs do well on both: in-distribution accuracy spreads from 0.10 to 0.86 and tracks generalization ($r = +0.67$; App. C.7).

Representation regularization also reduces catastrophic forgetting of the PT tasks without any replay data, but only partially: forgetting drops by about half for one run and is unchanged for the others (Fig. 6c; App. C.6.1).

Overall, representation regularization enforces what we call *naturalistic learning*: learn the task if it can be learned with the representation you have, and refuse a fit that would destroy it. Learnability is the hidden axis.

6 Confirmation in language models

We test whether the distillation result holds in language models, at two scales (App. B.5).

Models and task. Pythia-160M and Pythia-410M each have ten public PT runs that differ only in random seed (Biderman et al., 2023; van der Wal et al., 2025), the language-model analogue of our 278 synthetic runs. We fine-tune on a synthetic two-hop relational world of invented people, cities, and countries: the corpus states every one-hop fact, and the OOD test asks the two-hop question for held-out people whose one-hop facts were trained but never composed (corpus, FT protocol, and seed selection in App. B.5.1). The base models answer none of them. Composing two separately learned facts is a known weakness of language models (Press et al., 2023; Allen-Zhu & Li, 2024).

Both methods transfer to language models. The seeds already differ on the two-hop OOD test under plain FT (0.21–0.32 at 160M, 0.30–0.54 at 410M). At 160M we take the worst seed and apply both interventions of Secs. 4.2 and 5. Representation regularization toward the seed's own states raises its OOD accuracy from 0.222 under plain FT to 0.244–0.291, at every layer tried. Representation distillation from a strong seed's layer-10 representation raises it to 0.403, above regularization alone at the same layer (0.256; four FT seeds per arm; every treated run exceeds every control run; Fig. 7a; layers, weights, and seed counts in App. B.5.1).

The same transfer at 410M. We distill from two strong seeds into two weak ones, at three depths and three distillation weights, four FT seeds per cell (Table 3). Both of the representation regularization and distillation better than plain FT here. Regularizing the recipient toward its own base states, with no reference, exceeds plain FT in 12 of the 18 cells (every cell for the first recipient, and layer 16 for the second). Representation distillation exceeds plain FT in all 36 cells and, at layers 20 and

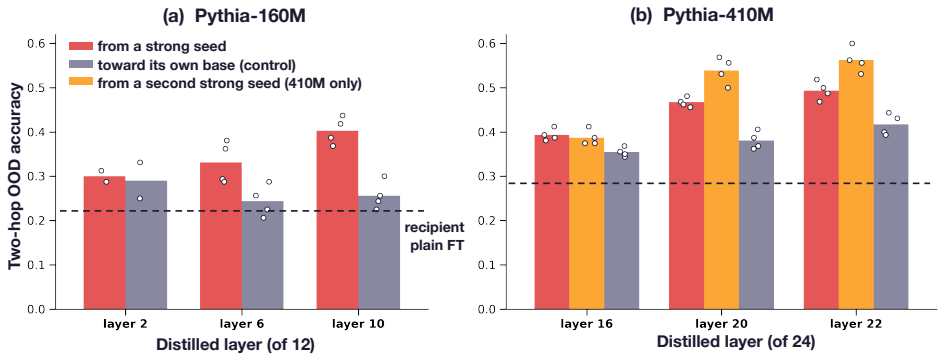


Figure 7: **Representation regularization and distillation both transfer to Pythia.** Both beat plain FT (dashed line): representation regularization toward the weak seed's own base states (gray) helps, and representation distillation from a strong seed (coloured) helps more, but only from deep layers. **a.** Pythia-160M. The worst of ten seeds, distilled at layers 2, 6, and 10 of its 12 layers. Plain FT gives 0.222; representation regularization gives 0.244–0.291; representation distillation at layer 10 gives 0.403, while layer 2 adds nothing beyond regularization. **b.** Pythia-410M. The worst seed, distilled from two strong seeds at weight 0.1. Plain FT gives 0.278; representation regularization gives 0.36–0.42; representation distillation at layers 20 and 22 of its 24 layers gives 0.47–0.56, while layer 16 adds little beyond regularization. Bars are means and dots individual FT seeds.

22 of the model's 24, lifts the first recipient from 0.278 to 0.47–0.56 and the second from 0.370 to 0.50–0.55, at or above the reference seeds' own scores and above the self-regularization control in 22 of the 24 cells with no overlap between treated and control runs. At layer 16, as at layer 2 of the smaller model, the reference adds little beyond regularization alone (Fig. 7b). At shallow layers, or at distillation weights of 0.5 and above, the graft falls below the control, so the transfer needs a deep layer and a light touch (weights).

7 Discussion

Summary. We showed, in a controlled setup, that FT generalization is mostly decided by the entity representation formed during PT. The quality of the representation, in terms of how it supports FT, seems to be a lottery over the PT seed, and is hard to predict from loss or in-distribution behavior. We found that interventions on the representation can move FT generalization, while distilling behavior does not.

Loss of Representational Plasticity. A broader problem beyond the lottery is that representations are nearly fixed via PT. As we saw in Sec. 4.3, a PT run that did not relate some groups of data cannot easily rearrange its representations to support FT generalization, and only a teacher representation could perform the surgery. This connects to a broader problem where representations in neural networks do not change easily after FT (Dohare et al., 2024).

Limitations. Our study has several limitations. First, our findings are mostly synthetic. The main causal results come from a synthetic world, chosen because it provides a controllable ground truth; nevertheless we do reproduce some of the distillation result on Pythia with 160M and 410M parameters on a two-hop reasoning task. Whether the lottery and the interventions hold in bigger pretraining at scale is unknown. Second, we did not find a single function that predicts, directly from a pretrained representation, how well a run will fine-tune: geometric summaries barely correlate with the outcome (App. C.2.2), and the best readout, while predictive, misses most good runs. Our interventions show that the representation determines FT generalization, but not which property of it does. Third, the strongest interventions work only partially: distillation lifts poor runs to about 0.28, far below the oracle's 0.90, and regularization must be tuned between allowing destructive reshaping and a rigid freeze that costs good runs their generalization (App. C.6.3).

AI USE STATEMENT

We used generative AI tools for:

- **Code and analysis:** writing experimental, analysis, and plotting code. All code was checked by the authors and results were verified with control experiments; most were reproduced independently by multiple independent authors.
- **Experiments:** running variations around experiments specified by the authors. The hypotheses, experimental designs, and interpretations are the authors' own.
- **Writing:** polishing text, suggesting phrasings, and finding literature. The text, claims, and narrative are the authors' own.

Thus, we used AI for: *implement methods*, *create or edit software code*, *create or modify scientific figures or images*, *edit a research paper to improve readability*, and *identify relevant literature*. We did not use them to *propose or refine hypotheses* or *interpret results*. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

REPRODUCIBILITY STATEMENT

The synthetic setup is described in Sec. 2, with the architecture, training and fine-tuning hyperparameters, and the representation regularizer in App. B.2, the FT generalization metric in App. B.3, and the probing recipe in App. B.4. The language-model setup is in App. B.5, and the statistical procedures are in App. C.7. All code will be released after the peer review process.

REFERENCES

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep neural networks, 2019. URL https://arxiv.org/abs/1711.08856.

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning, 2020. URL https://arxiv.org/abs/2012.13255.

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning, 2021. URL https://arxiv.org/abs/2101.11038.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2018. URL https://arxiv.org/abs/1610.01644.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.1, knowledge storage and extraction. *arXiv preprint arXiv:2309.14316*, 2023.

Zeyuan Allen-Zhu and Yuanzhi Li. Physics of language models: Part 3.2, knowledge manipulation, 2024. URL https://arxiv.org/abs/2309.14402.

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

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: Llms trained on "a is b" fail to learn "b is a", 2024. URL https://arxiv.org/abs/2309.12288.

Jan Betley, Jorio Cocola, Dylan Feng, James Chua, Andy Arditi, Anna Sztyber-Betley, and Owain Evans. Weird generalization and inductive backdoors: New ways to corrupt llms, 2025. URL https://arxiv.org/abs/2512.09742.

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, January 2026. ISSN 1476-4687. doi: 10.1038/s41586-025-09937-5. URL http://dx.doi.org/10.1038/s41586-025-09937-5.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023. URL https://arxiv.org/abs/2304.01373.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint. arXiv:2108.07258*, 2021.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Helena Casademunt, Caden Juang, Adam Karvonen, Samuel Marks, Senthooran Rajamanoharan, and Neel Nanda. Steering out-of-distribution generalization with concept ablation fine-tuning, 2025. URL https://arxiv.org/abs/2507.16795.

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, April 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10319-8. URL http://dx.doi.org/10.1038/s41586-026-10319-8.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. Underspecification presents challenges for credibility in modern machine learning, 2020. URL https://arxiv.org/abs/2011.03395.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping, 2020. URL https://arxiv.org/abs/2002.06305.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632 (8026):768–774, August 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07711-7. URL http://dx.doi.org/10.1038/s41586-024-07711-7.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017. URL https://arxiv.org/abs/1703.03400.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2020.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann Lecun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank, 2023. URL https://arxiv.org/abs/2210.02885.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning llms on new knowledge encourage hallucinations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7765–7784. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.444. URL http://dx.doi.org/10.18653/v1/2024.emnlp-main.444.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

Wes Gurnee and Max Tegmark. Language models represent space and time. *arXiv preprint arXiv:2310.02207*, 2023.

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

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Scott Krueger. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. *arXiv preprint arXiv:2311.12786*, 2023.

Keller Jordan. On the variance of neural network training with respect to test sets and distributions, 2024. URL https://arxiv.org/abs/2304.01910.

Jeevesh Juneja, Rachit Bansal, Kyunghyun Cho, João Sedoc, and Naomi Saphra. Linear connectivity reveals generalization strategies. *arXiv preprint. arXiv:2205.12411*, 2022.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. In *Proc. of the 36th Proc. Int. Conf. on Machine Learning (ICML)*, Proc. of Machine Learning Research. PMLR, 09–15 Jun 2019.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. Understanding catastrophic forgetting in language models via implicit inference, 2024. URL https://arxiv.org/abs/2309.10105.

Akarsh Kumar, Jeff Clune, Joel Lehman, and Kenneth O. Stanley. Questioning representational optimism in deep learning: The fractured entangled representation hypothesis, 2025. URL https://arxiv.org/abs/2505.11581.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution, 2022. URL https://arxiv.org/abs/2202.10054.

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

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DeG07_TcZvT.

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks, 2018. URL https://arxiv.org/abs/1802.01483.

Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations?, 2016. URL https://arxiv.org/abs/1511.07543.

Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same pre-training loss, better downstream: Implicit bias matters for language models, 2022. URL https://arxiv.org/abs/2210.14199.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2025. URL https://arxiv.org/abs/2308.08747.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. Berts of a feather do not generalize together: Large variability in generalization across models with similar test set performance, 2020. URL https://arxiv.org/abs/1911.02969.

Julian Minder, Clément Dumas, Caden Juang, Bilal Chugtai, and Neel Nanda. Overcoming sparsity artifacts in crosscoders to interpret chat-tuning, 2025. URL https://arxiv.org/abs/2504.02922.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines, 2021. URL https://arxiv.org/abs/2006.04884.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? *Adv. in Neural Information Processing Systems (NeurIPS)*, 2020.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms, 2018. URL https://arxiv.org/abs/1803.02999.

Kento Nishi, Rahul Ramesh, Maya Okawa, Mikail Khona, Hidenori Tanaka, and Ekdeep Singh Lubana. Representation shattering in transformers: A synthetic study with knowledge editing. *arXiv preprint arXiv:2410.17194*, 2024.

Core Francisco Park. Convergent world representations and divergent tasks, 2026. URL https://arxiv.org/abs/2602.00533.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. Fine-tuning enhances existing mechanisms: A case study on entity tracking, 2024. URL https://arxiv.org/abs/2402.14811.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models, 2023. URL https://arxiv.org/abs/2210.03350.

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

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

Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, Ian Tenney, and Ellie Pavlick. The multiberts: Bert reproductions for robustness analysis, 2022. URL https://arxiv.org/abs/2106.16163.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation, 2022. URL https://arxiv.org/abs/1910.10699.

Johannes Treutlein, Dami Choi, Jan Betley, Samuel Marks, Cem Anil, Roger Grosse, and Owain Evans. Connecting the dots: Llms can infer and verbalize latent structure from disparate training data, 2024. URL https://arxiv.org/abs/2406.14546.

Keyon Vafa, Justin Y. Chen, Jon Kleinberg, Sendhil Mullainathan, and Ashesh Rambachan. Evaluating the world model implicit in a generative model, 2024. URL https://arxiv.org/abs/2406.03689.

Oskar van der Wal, Pietro Lesci, Max Muller-Eberstein, Naomi Saphra, Hailey Schoelkopf, Willem Zuidema, and Stella Biderman. Polypythias: Stability and outliers across fifty language model pre-training runs, 2025. URL https://arxiv.org/abs/2503.09543.

Atticus Wang, Joshua Engels, Oliver Clive-Griffin, Senthooran Rajamanoharan, and Neel Nanda. Simple mechanistic explanations for out-of-context reasoning, 2025. URL https://arxiv.org/abs/2507.08218.

Jake Ward, Chuqiao Lin, Constantin Venhoff, and Neel Nanda. Reasoning-finetuning repurposes latent representations in base models, 2025. URL https://arxiv.org/abs/2507.12638.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think, 2025. URL https://arxiv.org/abs/2410.06940.

Charlie Zhang, Graham Neubig, and Xiang Yue. On the interplay of pre-training, mid-training, and rl on reasoning language models, 2025. URL https://arxiv.org/abs/2512.07783.

Table I links the appendix sections to the main text.

A Related work

Seed variance. Runs with the same training procedure but different seeds can match on loss and benchmarks and still generalize differently (Juneja et al., 2022; McCoy et al., 2020; D'Amour et al., 2020). FT seeds alone can substantially change downstream accuracy, and the variance has been traced to weight initialization and data order (Dodge et al., 2020) and to optimization instability (Mosbach et al., 2021). Suites of models pretrained with different seeds enable study of PT variance (Sellam et al., 2022; van der Wal et al., 2025). Networks with the same output behavior can have markedly different internal representations (Kumar et al., 2025). We identify the pretrained representation underlying this variance and transfer it between runs. The lottery ticket hypothesis (Frankle & Carbin, 2019) attributes training success to a sparse subnetwork present at initialization; in our experiments, initialization alone does not explain FT outcomes (App. C.1.3), whereas pretrained representations predict them before FT. Seed-to-seed differences have also been characterized through linear mode connectivity (Frankle et al., 2020).

How pretraining supports fine-tuning. Neyshabur et al. (2020) ask what transfer learning transfers, and Kumar et al. (2022) show that FT can distort pretrained features and degrade out-of-distribution performance. Models with equal PT loss can differ in downstream performance (Liu et al., 2022). Pretrained language models have a low intrinsic dimension for FT, which helps explain the data efficiency of FT (Aghajanyan et al., 2020). In controlled settings, whether FT can extract a fact depends on how PT stored it (Allen-Zhu & Li, 2023), and reinforcement-learning post-training yields additional capabilities only when PT leaves capacity for improvement (Zhang et al., 2025). Mechanistic studies find that FT often adds a small modification to pretrained capabilities (Jain et al., 2023), enhances mechanisms that PT already built rather than creating new ones (Prakash et al., 2024), and, in reasoning FT, repurposes a direction already present in the base model's activations (Ward et al., 2025). In LLMs, facts unknown to the base model are learned slowly by FT

Table 1: **Appendix guide.**

Appendix	Topic	Main-text reference
A	Related work	all
Experimental setup		
B.1	Notation and terminology	all
B.2	Synthetic model and training	§2
B.3	FT generalization metric	§2, §5
B.4	Representation extraction and probing	§2, Fig. 1
B.5	Language-model setup	§6
Additional experiments		
C.1.1	Pretraining curves	§3
C.1.2	Variation across FT seeds	§3
C.1.3	Initialization and data order	§3
C.2.1	Step-0 answer readout	§4
C.2.2	Selection by pretrained-model metrics	§4
C.2.3	Encoding of entity angles	§4
C.3.1	Oracle injection	§4.1
C.4.1	Representation distillation ablations	§4.2
C.4.2	Distillation across seeds	§4.2
C.5.1	Across-generalization and reuse	§4.3
C.5.2	Transfer tests across seeds	App. C.5.1
C.5.3	Cross-group identifiability	App. C.5.1
C.6.1	Fixed-target regularization and forgetting	§5
C.6.2	Per-layer parameter change	§5
C.6.3	Fixed-target and periodic regularization	§4, §5
C.6.4	Weight-space and linear-probe baselines	§5
C.7	Statistical notes	§3–§5
C.8	Behavior distillation	§4.2
Visualizations		
D.1	Representation gallery	§4

and, once learned, increase hallucination (Gekhman et al., 2024), and training on a teacher's outputs transmits its traits through semantically unrelated data only when teacher and student share a base model (Cloud et al., 2026). Multi-task PT as a route to general representations dates to Caruana (1997) and Aghajanyan et al. (2021). Meta-learning optimizes an initialization explicitly for fast adaptation (Finn et al., 2017; Nichol et al., 2018), whereas standard PT optimizes only its own loss.

Fine-tuning. Models fine-tuned on scattered observations can infer and verbalize the latent structure behind them (Treutlein et al., 2024), and much of this out-of-context generalization amounts to adding a steering vector toward a concept the pretrained model already represents (Wang et al., 2025). Narrow FT on insecure code produces broad misalignment (Betley et al., 2026), and small amounts of FT in narrow contexts can shift behavior far outside them (Betley et al., 2025), while FT on "A is B" does not teach "B is A" (Berglund et al., 2024). Model diffing studies how FT changes a model's representations, for example with crosscoders that compare base and chat models (Minder et al., 2025), and ablating undesired concept directions during FT can steer how the fine-tuned model generalizes (Casademunt et al., 2025).

World representations and synthetic setups. Our setup extends the synthetic world-modeling experiments of Park (2026), where transformers trained on relational statements over entities form a linear representation of the world. We measure this representation and transfer it between models. Sequence models trained on synthetic data form internal world representations, such as the board state of Othello (Li et al., 2023), and large language models represent space and time (Gurnee & Tegmark, 2023), although a generative model can pass standard world-model diagnostics while its implied world model is far less coherent (Vafa et al., 2024). Alain & Bengio (2018) use linear probes to assess information encoded in representations. Li et al. (2016) study whether different runs of one architecture converge to similar features. Representational similarity across runs is measured by SVCCA and CKA (Raghu et al., 2017; Kornblith et al., 2019) and by model stitching (Lenc & Vedaldi, 2015; Bansal et al., 2021). Nishi et al. (2024) show in a synthetic setting that knowledge editing disrupts a structured representation; we observe related changes during FT.

Distillation. Knowledge distillation transfers a teacher's behavior (Hinton et al., 2015; Gou et al., 2021), and feature distillation adds intermediate-layer hints (Romero et al., 2015). Our representation distillation uses an intermediate-layer loss with a teacher pretrained at the same model size. We assess whether the recipient acquires the teacher's ability to generalize after FT, a transfer that requires retaining the distillation loss during FT.

Forgetting and regularized fine-tuning. Catastrophic forgetting (McCloskey & Cohen, 1989) is classically mitigated by weight-space regularization (Kirkpatrick et al., 2017; Li et al., 2018) or replay. In language models, forgetting has been measured under continual instruction tuning (Luo et al., 2025) and explained in part as FT skewing the model's implicit inference of the task (Kotha et al., 2024). In our experiments, a weight-space penalty preserves existing abilities but does not confer across-generalization. Activation-space regularization distinguishes runs that can learn the task using their pretrained representations from those that fit it by substantially changing those representations (App. C.6.4). Achille et al. (2019) show that data arriving late in training cannot reshape representations; FT on cross-group instances in Sec. 4.3 provides a controlled example of this effect.

B Experimental setup

B.1 Notation and terminology

This section defines the terminology used throughout the paper and compares plain FT performance with regularized FT and teacher quality for selected seeds (Table 2).

Seeds. A PT run is fixed by its random seed, and we refer to runs by the *tier* (super, good, boundary, or bad, assigned by FT generalization) rather than by seed number. We also identify one recurring run as the top plain performer: it scores highest under plain FT but poorly under regularized FT. Runs pretrained under the ortho policy are marked ortho and are distinct from their shared-policy counterparts.

Tests. *Across-generalization* is measured by fine-tuning on within-group instances drawn from both groups and evaluating on cross-group compositions; success requires a joint representation. *Reuse* is measured by fine-tuning on one group and evaluating the same skill on the other; success requires transferring the skill between separate representations. *FT generalization* denotes the across-generalization score unless stated otherwise. The *null band* is the accuracy region at or below 0.205: chance is 0.06, the best input-independent answer 0.10, and ortho models reach at most 0.20, so a run above 0.205 is credited with across-generalization and one at or below it is not. *Tiers* partition the seeds by FT generalization: super ≥ 0.40, good 0.30–0.40, boundary 0.205–0.30, bad ≤ 0.205 ($3/12/47/216$ runs). "Good runs" in selection analyses means the top two tiers (super or good), 15 of 278.

Coordinate and answer readouts. The *coordinate readout* is a linear map from a layer-5 representation to an entity's true coordinates (used for the maps and decodability). The *answer readout* is a linear probe of the composed answer at the = position from a single forward pass (used as the step-0 predictor of Sec. 4).

Regularization strength. The representation regularizer's strength is $\lambda_{\text{rep}} = 2.0$ at layer 5 for the synthetic model, 0.05–1.0 by layer and size for the language-model distillation (App. B.5), and the weight-space L2-SP control uses its own λ_{L2}; these are distinct constants.

Scoring. The Pythia two-hop task of App. B.5 reports accuracy at its best evaluation. Unless otherwise labeled, synthetic-model FT scores denote FT generalization under the stated protocol, with regularized and plain values reported separately.

Step-0 answer readout correlations. The answer readout predicts regularized in-distribution accuracy at $r = 0.48$, regularized FT generalization at $r = 0.64$, and plain FT generalization at $r = 0.32$ (all $n = 278$). Claims about predicting FT outcomes with the answer readout refer to the regularized results, which measure learning with limited representation change (Sec. 5).

Table 2: **Plain FT performance does not determine representation quality.** Values describe individual seeds rather than entire tiers. Regularized FT limits changes to the pretrained representation. The top plain performer substantially changes its representation during plain FT, scores 0.14 under regularized FT, and is a poor teacher. Two seeds whose representations are close to the ground truth score modestly under plain FT but highly under regularized FT and are the strongest teachers. Such teachers improve FT generalization across many recipients (App. C.4.2).

seed	plain FT	regularized FT	as teacher
top plain performer	0.41–0.50	0.14	poor
good-geometry seed	0.20	0.39	strongest
good-geometry seed	0.19	0.35	strongest

Regularized values are each seed's score at the paper's standard regularization strength ($\lambda = 2.0$, App. B.2); the first good-geometry seed is also the run of Fig. 12c, where the regularized-seed protocol of Sec. 5 gives 0.35, and its sweep optimum in Fig. 20 is 0.39. A seed's regularized value therefore varies with the protocol at the level of hundredths; its tier does not.

B.2 Synthetic model: architecture and training

Unless otherwise stated, synthetic-model experiments, including the 278 PT runs and interventions on them, use the configuration below. Runs differ only in the PT seed and, where stated, the FT seed. The oracle experiment uses the separate setup in App. C.3.1.

Architecture. We use a 6-layer Qwen2 transformer with hidden size 128, feed-forward size 512, 4 attention heads, and weights initialized at standard deviation 0.1. Inputs use a character-level tokenizer (an entity is represented by its ID string, and no world city's coordinates appear in the text) with a maximum sequence length of 256.

Pretraining data. The corpus mixes two kinds of samples in equal proportion. Regular samples contain only relational statements over world cities. In-context samples first define one to four new entities with random coordinates, written explicitly as `p_XXXX=(x,y)`, and then state relations that use them; these entities are redrawn for each sample; an in-context ID can coincide with a world city's ID, in which case the in-context coordinates override the world city's within that sample.

Pretraining. We use AdamW (β_1=0.9, β_2=0.999, ϵ=10^{-8}) with learning rate 6×10^{-4}, weight decay 0.01, gradient-norm clipping at 1.0, a linear schedule with 50 warmup steps, batch size 128, and bf16 precision. Training runs for 12 epochs over the relational corpus, corresponding to 656k optimizer steps.

Fine-tuning. FT loads the pretrained weights only and resets the optimizer and scheduler; it uses the same optimizer, weight decay, clipping, schedule, and batch size as PT, at learning rate 2×10^{-4} with 20 warmup steps, for approximately 33k steps on the across-generalization task (the transfer tests of App. C.5.1 run about twice as long). Unless a spread over FT seeds is reported, results use one fixed FT seed. We evaluate periodically during training and report FT generalization under the stated protocol.

Representation regularization. The regularizer adds $\lambda \cdot \frac{1}{N} \sum_x \|h_5(x;\theta) - h_5(x;\theta_{\text{teacher}})\|^2$ to the loss on every step, at layer 5. The standard strength for the synthetic model is $\lambda = 2.0$. The target is the model's own layer-5 representation, in one of two variants: *fixed-target* regularization keeps the pretrained representation as the target throughout FT (the strength sweep of Fig. 20), and *periodic* regularization retakes the target from the current model at the start of each FT segment (the regularized results across all seeds in Secs. 4 and 5 and the language-model experiments; App. C.6.3 compares the two). Representation distillation (Sec. 4) keeps the teacher checkpoint frozen and the distillation loss active during FT. Fig. 20 varies λ over $\{0, 0.5, 1, 2, 4, 8, 16, 32, 64\}$ to locate the interior optimum. Unless another strength is specified, synthetic-model regularization uses $\lambda = 2.0$.

B.3 FT generalization metric

Let A denote the Americas and R the rest of the world. An instance's *composition* labels the group membership of its two segments: the two within-group compositions are AA|AA and RR|RR, and the four cross-group compositions are AR|AR (both segments cross the groups), AA|RR (the seg-

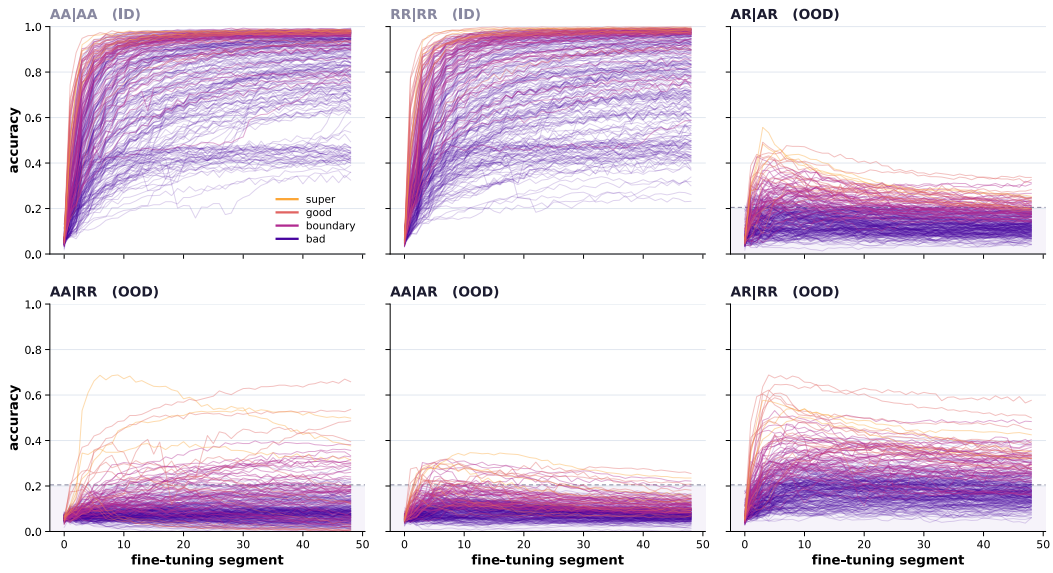


Figure 8: **Fine-tuning curves by composition.** All 278 runs, colored by tier, for the two in-distribution compositions (top left) and the four cross-group compositions whose mean is the paper's FT generalization. Every tier fits the task in distribution; the tiers separate on each cross-group composition individually.

ments lie within different groups), and the mixed AA|AR and AR|RR. For the synthetic task, FT generalization is the mean over the four cross-group compositions. Fig. 8 shows the FT trajectories for each composition; because FT proceeds in short segments with a full evaluation after each, the x-axes count segments. The two within-group compositions are in distribution, and every tier, including bad runs, reaches high accuracy on them (median final 0.89). The four cross-group compositions differ in absolute difficulty but show the same ordering of tiers, so the aggregate result is not driven by a single composition.

Null band. Chance under the $\pm 10^\circ$ criterion is 0.06, and the best input-independent strategy, always answering the modal angle of the training distribution, reaches 0.10. Ortho-pretrained models, which lack cross-group alignment by construction, reach 0.05–0.20 on the cross-group compositions (30 runs, App. C.5.2). The null band's upper edge at 0.205 is set above all three reference values, and only runs above it are counted as showing across-generalization.

B.4 Probing procedure: extraction, decodability, and projection

The maps, gallery panels, and coordinate decodability scores use the same probing procedure for every model and checkpoint.

Extraction. For each entity we run the frozen model on the prompt `angle(p_XXXX,` (a PT task prefix followed by the entity ID and a comma; no answer is present) and collect the layer-5 hidden states at two positions, the ID's last digit and the trailing comma. Their concatenation, $2 \times 128 = 256$ dimensions, is the entity's representation. The procedure is applied to all 5,175 entities with the model frozen throughout, and the states are z-scored per dimension before any analysis.

Decodability. A linear regression maps representations to the entity's true coordinates (x, y). Within-group decodability fits and evaluates this coordinate readout within one entity group. Cross-group decodability tests for a joint representation by fitting the coordinate readout on one group and evaluating it on the other, in both directions. We report the mean R^2 (the gallery grouping of App. D.1 uses the conservative minimum). A joint representation scores near 0 or above, whereas separate representations score well below zero (the gallery's merged / partial / disjoint groups use -3 and -10 as boundaries). Because R^2 is normalized by the coordinate variance, large negative values indicate cross-group prediction errors several times larger than the coordinate variance.

Table 3: **Pythia-410M distillation sweep.** Two-hop OOD accuracy (mean of four FT seeds) after representation distillation from teacher A or teacher B into each recipient, at each layer and weight, compared with the self-distillation control. Plain FT gives 0.278 and 0.370 for the two recipients. Bold marks cells where every treated run exceeds every control run.

		recipient 1 (plain 0.278)			recipient 2 (plain 0.370)		
layer	weight	teacher A	teacher B	control	teacher A	teacher B	control
16	0.05	0.402	0.358	0.355	0.491	0.444	0.423
16	0.1	**0.394**	**0.388**	0.355	0.455	0.445	0.467
16	0.2	0.394	0.389	0.370	0.456	0.484	0.461
20	0.05	**0.480**	**0.537**	0.344	0.487	0.491	0.367
20	0.1	**0.467**	**0.539**	0.381	**0.523**	**0.511**	0.358
20	0.2	**0.470**	**0.519**	0.391	**0.512**	**0.519**	0.314
22	0.05	**0.511**	**0.519**	0.397	**0.517**	**0.478**	0.308
22	0.1	**0.494**	**0.562**	0.417	**0.542**	**0.497**	0.361
22	0.2	**0.486**	**0.541**	0.389	**0.522**	**0.548**	0.344

Projection for figures. Each displayed panel is a two-dimensional projection whose basis is determined at the checkpoint shown: we fit the four coordinate readout directions (x and y within each group), take the top two right singular vectors of that 4×256 matrix as the plane, project, fix signs by correlating each axis with true longitude, and rotate by $45°$ for display. Entities are colored by true longitude, one colormap per group. The projection is for visualization only; coordinate decodability uses the full 256-dimensional coordinate readouts above.

B.5 Language-model experimental setup

B.5.1 Pythia-160M and Pythia-410M

Models. We use Pythia-160M and Pythia-410M with their nine public seed variants each (`pythia-160m`, `pythia-160m-seed1` through `seed9`, and likewise for 410M), giving ten PT runs per size that differ only in random seed.

Task. *relhop* is a synthetic two-hop relational task with invented names (400 people, 60 cities, 20 countries; names composed from a 24-syllable inventory so every entity is multi-token and carries no prior meaning). The corpus states one-hop facts for every entity ("X lives in the city of Y", "the city of Y is in the country of Z", each also in question form) but two-hop questions ("in which country does X live?") only for 240 of the 400 people; the OOD evaluation asks the two-hop question for the 160 held-out people, so answering requires composing the stored one-hop facts. Chance accuracy is 0.05, and the base model scores 0.

Fine-tuning. At both sizes, FT uses 24 segments of 300 steps at learning rate 10^{-4}. We evaluate two-hop ID and OOD accuracy, Pile validation perplexity, and LAMBADA accuracy after every segment. Plain FT of each of the ten seeds with one fixed FT seed gives the ranges in Sec. 6 (0.21–0.32 at 160M, 0.30–0.54 at 410M). Teachers are selected by their score under periodic regularization: the auxiliary loss of Sec. 4 uses $\lambda_{\text{rep}} = 0.1$ and targets the model's own states, updated after each segment. Recipients are selected for low plain FT scores. At 160M the recipient is the seed that scores lowest under plain FT; at 410M the two recipients are the seed that scores lowest under both protocols (0.30 plain) and the next-lowest under regularization (0.39 plain), and the two teachers score highest under regularization (0.48 and 0.47; the first of them scores 0.30 under plain FT).

Distillation. Representation distillation between Pythia seeds follows the procedure used for the synthetic models: an auxiliary loss matches the recipient's layer-ℓ states to the teacher's on the task text. The teacher checkpoint remains frozen, and the distillation loss remains active throughout FT. The control applies the same procedure using the recipient's own base states. At 160M we distill at $\ell \in \{2, 6, 10\}$ of 12 layers, with distillation weight 0.5 at layers 2 and 6 and 1.0 at layer 10, and four FT seeds per arm at layers 6 and 10 (two at layer 2). At 410M, Table 3 sweeps $\ell \in \{16, 20, 22\}$ of 24 and distillation weights $\{0.05, 0.1, 0.2\}$ for each teacher and recipient, four FT seeds per cell, with a self-distillation control per cell and plain FT per recipient (four seeds). A preliminary sweep at layers 8 and 12 with weights 0.5 and 2.0 (three seeds per cell, first teacher and first recipient), in which representation distillation did not exceed the control in any cell (layer 12, weight 2.0: 0.28

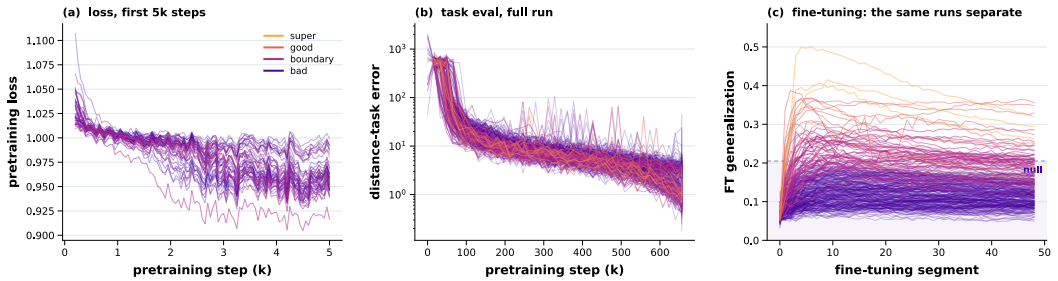


Figure 9: **PT and FT trajectories by tier.** (a) PT loss over the first 5k steps (47 sampled runs) and (b) held-out distance-task error across the full 656k steps (all 278 runs), colored by eventual tier; neither measure separates the tiers. (c) FT generalization under the plain protocol for the same 278 runs; the tiers separate early in FT and remain separated.

against 0.40), motivated the choice of deeper layers and lower weights in the table. We report the recipient's best two-hop OOD accuracy over segments.

C Additional Experiments

C.1 Seed Variance (Sec. 3)

C.1.1 Pretraining Curves Across Seeds

Fig. 9 compares PT and FT trajectories across seeds. Panel (a) shows cross-entropy loss over the first 5k steps for 47 sampled runs, colored by their eventual tiers. The tiers remain indistinguishable during the largest changes in loss. Panel (b) shows held-out performance on the distance task at 42 checkpoints across the full 656k steps for all 278 runs; the remaining six tasks show the same pattern. These evaluations do not separate the tiers at any checkpoint. Under the plain FT protocol of Sec. 3, however, the same 278 runs separate by tier within the first FT segments and remain separated (panel c). Thus, PT determines the differences in FT generalization, but PT performance does not reveal them.

Fig. 10 holds the PT loss fixed by construction rather than approximately. For each of 266 seeds we take the checkpoint at which the held-out loss first reaches 0.960 ± 0.005, a value every run passes through, and fine-tune from that checkpoint under half the usual budget (24 segments). The compute behind these checkpoints still varies by $2.9\times$ (230k to 656k steps), yet FT generalization spans 0.078–0.453 (median 0.151), and 53 of the 266 runs exceed the null band. Seeds that reach the matched loss earlier generalize slightly better ($r = -0.38$ against step), and each seed's matched-loss outcome tracks its final-checkpoint outcome at $r = +0.84$ (Spearman $+0.81$), so the ordering across seeds is established by the time a run reaches this loss and is preserved by the remaining training.

C.1.2 Variation Across Fine-Tuning Seeds

Table 4 reports the per-seed results from Fig. 2d, with fourteen FT seeds for each of seven PT seeds. The super and good seeds never fall below 0.31, the bad seeds never rise above 0.10, and only the boundary seeds span the null band. A one-way variance decomposition over this grid attributes 89.9% of the outcome variance to the PT seed ($F(6, 91) = 125$); a replication on 43 randomly drawn seeds $\times$ 10 FT seeds attributes 73.6% ($F(42, 387) = 29$), so the PT seed accounts for the majority of the variance under both designs (App. C.7).

C.1.3 Initialization–Order Combinations Across Ten Seeds

The PT seed determines the weight initialization and data order. We vary these separately to test whether either accounts for FT generalization, following the decomposition of FT seeds in Dodge et al. (2020). Fig. 12c crosses the initializations and data orders of one good-tier and two bad-tier seeds under regularized FT. Only the good-tier seed's original pairing reaches the good tier (0.35

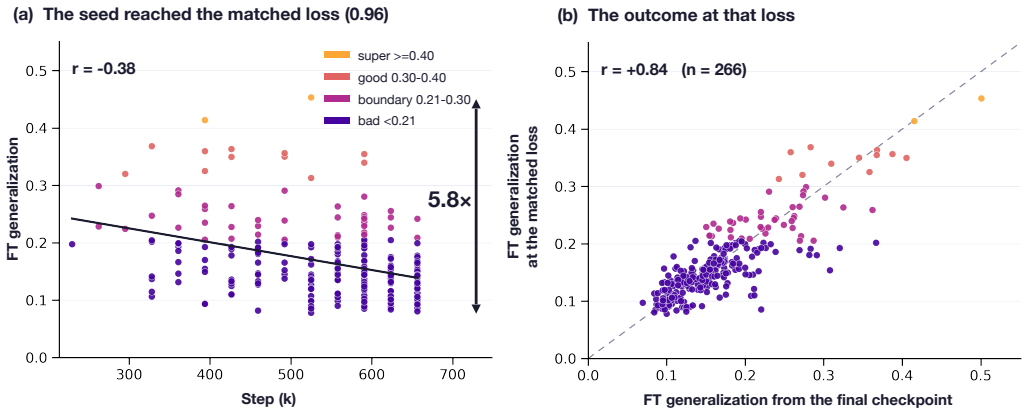


Figure 10: **Matching PT loss does not remove the variation in FT generalization.** (a) FT generalization for each of 266 seeds, fine-tuned from its checkpoint at held-out loss 0.960 ± 0.005, plotted against the step at which it reached that loss and colored by eventual tier. (b) The same results plotted against FT generalization from each run's final PT checkpoint.

Table 4: **Variation across FT seeds.** FT generalization across fourteen FT seeds per PT seed, for seven seeds spanning the tiers.

tier	n	min	median	max	range
super	14	0.336	0.392	0.446	0.110
good	14	0.315	0.342	0.355	0.040
boundary	14	0.169	0.228	0.347	0.178
boundary	14	0.139	0.231	0.308	0.168
boundary	14	0.093	0.129	0.308	0.214
bad	14	0.049	0.060	0.098	0.050
bad	14	0.066	0.084	0.098	0.032

regularized; 0.25 under the plain protocol of the full grid below). One bad seed's data order produces boundary-tier outcomes with other initializations (0.26, 0.24), although its original pairing remains at the null level (0.08). The ten-seed grid repeats this comparison with ninety recombinations, each pretrained in full and evaluated under plain FT (Fig. 11). A seed's original score predicts neither its initialization's performance under other data orders nor its data order's performance with other initializations (correlations near zero). High FT generalization therefore depends on the initialization–order pair, rather than on either component alone.

Fig. 11 reports the full ten-seed grid, in which the seeds span the tiers and all 100 models, including the 90 recombinations, use the plain FT protocol of Sec. 3. The diagonal contains each seed's original pairing: three reach the good tier (0.30–0.41) and three have low scores (0.06–0.13). Among the 90 recombinations, fourteen exceed the null band and two reach the good tier. A seed's original score predicts neither its initialization's mean score across other orders ($r = -0.12$) nor its order's mean score across other initializations ($r = 0.02$). The highest-scoring original seed averages 0.15 across other orders and 0.18 when its order is paired with other initializations. The highest-scoring recombination (0.35, an off-diagonal cell) exceeds both source seeds' original scores (0.13 and 0.30), so its performance cannot be attributed to either component alone. Each cell contains a single PT run, and we therefore read the grid as evidence for an interaction between initialization and data order.

C.2 How the representations differ (Sec. 4)

C.2.1 Step-0 answer readout

The step-0 answer readout of Sec. 4 is a linear probe fitted to states from a single forward pass through the frozen pretrained model. We take 2,500 held-out `sangle` prompts, run the frozen

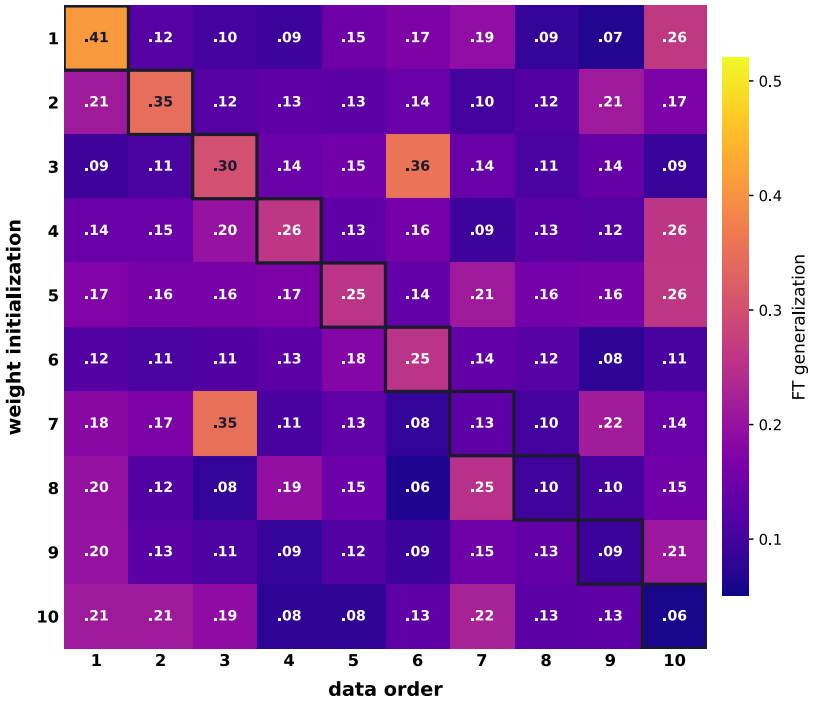


Figure 11: **Initialization–order interactions across ten seeds.** Ten PT seeds' weight initializations are crossed with their data orders; each cell reports FT generalization after full PT and plain FT, and the outlined diagonal contains the original pairings. Rows and columns are anonymized as 1–10 and sorted by the original score. Recombination scores cannot be predicted from initialization or order alone. The highest-scoring recombination (0.36) exceeds both source seeds' original (diagonal) scores.

model on each prompt up to and including the = token, and collect the hidden state at that position (layer 6 of 6; layer 5 gives similar results). A ridge regression ($\alpha = 5$) maps these states to the answer angle, encoded as $(\cos\theta, \sin\theta)$; it is fit exclusively on 80% of the within-group prompts. The readout score is the circular correlation between the decoded and true angles, computed on the held-out within-group prompts for the x-axis of Fig. 12ab and separately on the cross-group prompts. The probe measures how well the task's answer can be decoded linearly at the position where the model must produce it; because no gradients update the model, the score characterizes the pretrained representation. Fig. 12ab shows that it predicts both in-distribution learning and eventual FT generalization across seeds.

C.2.2 Selecting runs with static metrics

Fig. 13 tests whether ten scalar summaries of the pretrained model predict FT generalization across seeds: final PT loss; summaries of the final layer-5 representation (coordinate decodability across and within groups, participation ratio, effective rank, spectral entropy, representation norm, and the distance between the two groups' centroids); and the two angle-recovery scores of App. C.2.3. These metrics provide limited separation, as the geometric summaries correlate with FT generalization at $|r| \leq 0.16$ and their top deciles contain 8–12% good runs against a 5% base rate. PT loss reaches $r = -0.41$, but its lowest-loss decile contains only 7% good runs. Task-specific encoding also differs little across tiers, as single-entity angle recovery is high in all four tiers ($R = 0.86$–0.92) and correlates weakly and negatively with FT generalization ($r = -0.15$), while cross-group transfer of the angle readout is uncorrelated with FT generalization ($|r| < 0.02$). Thus, all tiers encode the single-entity quantities needed for the task, while differing in their ability to compose them (App. C.2.3). The step-0 answer readout of App. C.2.1 yields 26% good runs in its top decile. It correlates with plain FT generalization at $r = 0.32$ in this figure and with regularized FT gener-

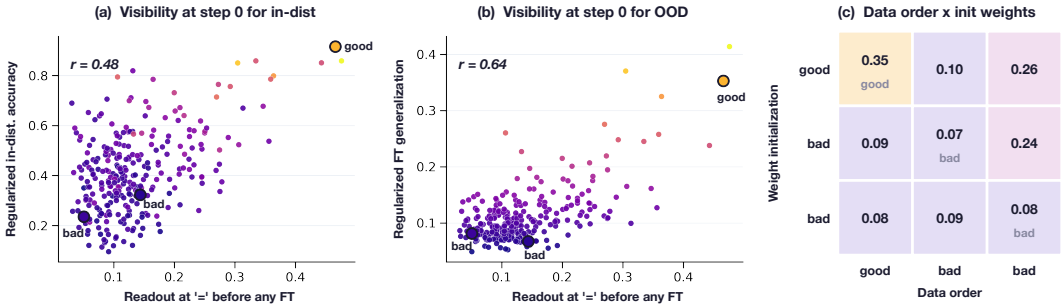


Figure 12: **Step-0 answer readout and initialization–order interactions.** (a, b) The step-0 answer readout at the = position of the pretrained checkpoint (method in this appendix) predicts what regularized FT will later achieve: in-distribution learning ($r = 0.48$) and FT generalization ($r = 0.64$; color: FT generalization, $n = 278$). The highlighted seeds are crossed in panel (c). (c) Three weight initializations crossed with three data orders, with full PT for all nine combinations and regularized FT as in (a, b). Cells report FT generalization. Only the good-tier seed's original pairing reaches the good tier (0.35). One bad seed's order produces boundary-tier outcomes with other initializations (0.26, 0.24), although its original pairing remains at the null level (0.08). Fig. 11 reports the full ten-seed grid under plain FT.

alization at $r = 0.64$ (Fig. 12). FT generalization is therefore predicted by the decodability of the answer from the representation, rather than by the static geometric summaries tested here.

C.2.3 Encoding of single-entity angles

We test whether bad-tier seeds fail to generalize because they encode entity positions less accurately. The across-generalization test requires a composed quantity: the angle between entities drawn from different groups. We examine the underlying single-entity quantity, each entity's angular position within its own group, across all seeds.

Probe and metric. For each entity c we run the frozen model on the prompt `angle(p_XXXX,` and take the layer-5 hidden state at the ID's last digit, $h_c \in \mathbb{R}^{128}$ (the single-position variant of the procedure in App. B.4). Let θ_c be the entity's true angular position and $y_c = (\cos\theta_c, \sin\theta_c)$ its target on the unit circle. A ridge readout maps the centered states to the circle,

$$W \;=\; \arg\min_{W} \sum_{c} \big\| \, y_c - W^{\top}(h_c - \bar{h}) \, \big\|^2 \;+\; \|W\|_F^2, \tag{1}$$

and the recovered angle is the phase of the projection, $\hat{\theta}_c = \operatorname{atan2}(p_{c,2},\, p_{c,1})$ with $p_c = W^{\top}(h_c - \bar{h})$. Recovery quality is the mean resultant length of the angular errors,

$$R \;=\; \max_{s\in\{+1,-1\}} \left| \frac{1}{N} \sum_{c=1}^{N} e^{\,i\,(s\hat{\theta}_c - \theta_c)} \right|, \qquad N = 5{,}175, \tag{2}$$

which equals 1 exactly when the recovered angles reproduce the true ones up to a global rotation and reflection, without requiring a fixed coordinate frame, and concentrates near $0.89/\sqrt{N} \approx 0.012$ for unrelated angles. The example panels of the figure use the stored two-position representations of App. B.4, which agree with the single-position states used for all seeds ($R = 0.93$ versus 0.91–0.92 for the example seeds).

Angle encoding across tiers. Fitting and evaluating Equation 1 within each run gives similar angle recovery across tiers: mean R is 0.92 for super, 0.86 for good, 0.88 for boundary, and 0.90 for bad. Recovery correlates weakly with eventual FT generalization ($r = -0.15$; Fig. 14, bottom left). This angular structure is also visible without a readout in the cosine-similarity matrix, with entities sorted by true angle:

$$S_{ij} \;=\; \frac{\langle h_i - \bar{h},\; h_j - \bar{h} \rangle}{\|h_i - \bar{h}\| \, \|h_j - \bar{h}\|}, \tag{3}$$

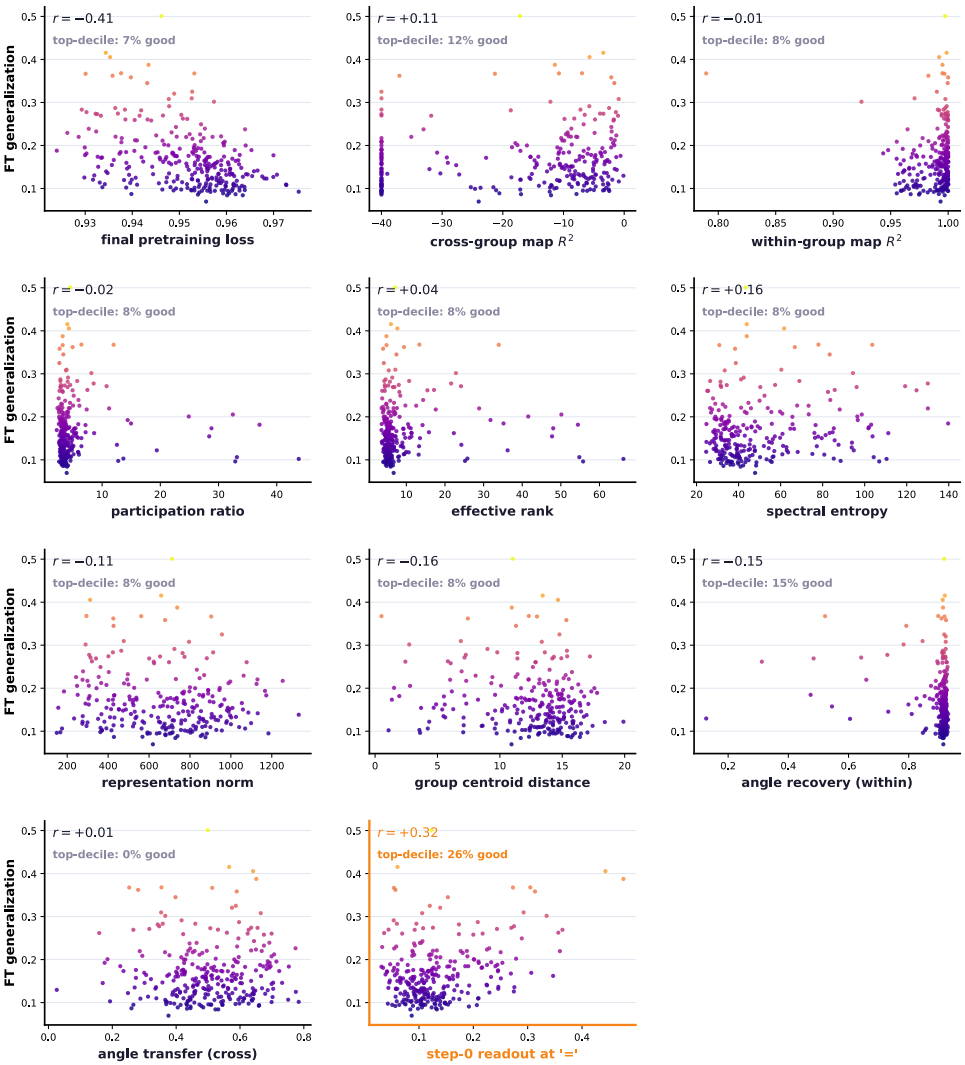


Figure 13: **Static metrics are weak predictors of FT generalization.** Ten scalar summaries of the pretrained model against eventual FT generalization across the seeds, each panel annotated with its correlation and the fraction of good runs in its top decile (base rate 5%). The step-0 answer readout (bottom right, orange) is the strongest selector.

in which the bright diagonal band wraps at the corners, consistent with a circular arrangement, in every tier (Fig. 14, top row). The matrices of a super seed and a bad-tier seed are visually similar, although their FT generalization differs by a factor of seven (0.50 versus 0.07).

Cross-group transfer of the angle readout. We fit W on one group's entities and evaluate Equation 2 on the other group, giving the transfer score $R_\times$. This tests whether the angular readout learned for one group also decodes the other group's entities. We find $R_\times \approx 0.5$ in every tier, forty times chance, and no correlation with FT generalization ($r = -0.01$; Fig. 14, bottom middle). Thus, poor FT generalization does not imply an absence of shared angular information.

Variance explained by the angle plane. Across seeds, the angle plane accounts for 0.04%–1.4% of representation variance (median 0.3%), nearly two orders of magnitude less than the top principal component (median 24%). This share also has little correlation with FT generalization ($r = +0.10$; Fig. 14, bottom right). The small variance share explains why projections onto the top principal components show no angular order and why spectral summaries provide little information about FT

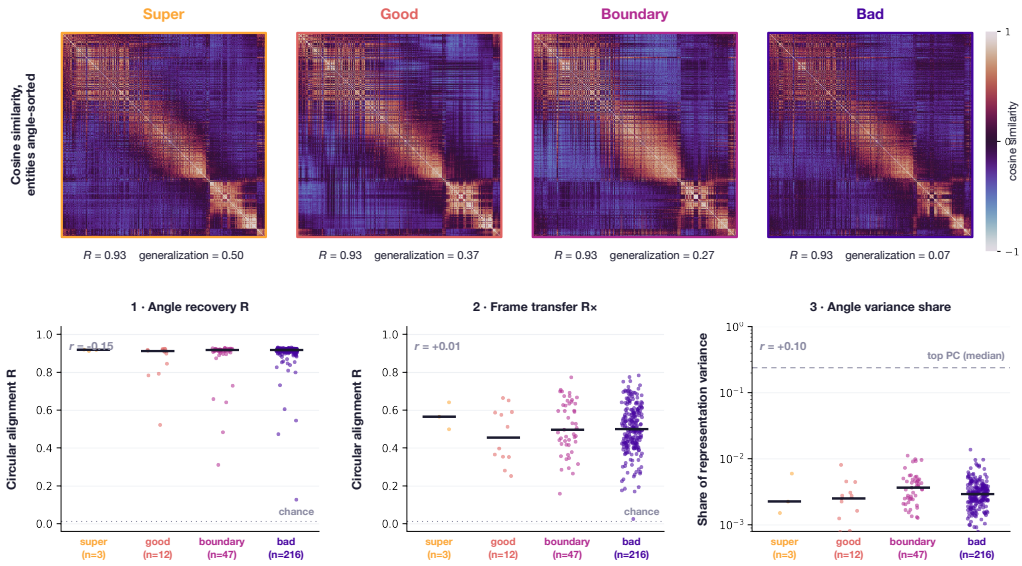


Figure 14: **Single-entity angle encoding is similar across tiers.** Top: cosine similarity (Equation 3) between entities sorted by true angle, for one example seed per tier, from the stored final-checkpoint representations of App. B.4. The same banded, wrap-around structure appears in every tier, and the readout of Equation 1 recovers the angle at $R = 0.93$ for all four seeds, while FT generalization spans 0.50 to 0.07. Bottom, across the seeds ($n = 278$), by tier: (1) within-group angle recovery, (2) cross-group transfer of the angle readout, (3) the share of representation variance in the angle plane (log scale), with the top principal component's median share for reference. Black bars are medians; dotted lines are chance (Equation 2). All three measurements are similar across tiers and have little correlation with FT generalization.

generalization (Fig. 13). It is also consistent with the representation distillation result of App. C.4.1: rank truncation removes low-variance directions and prevents transfer.

The single-entity angle and its cross-group linear decodability are therefore present to similar degrees in all tiers, within a subspace that explains little of the total variance, and these single-entity measurements do not predict which seeds will succeed. Bad-tier seeds nevertheless fail to compose these quantities into the four-entity function required by the test; this failure is the compositional deficit discussed in Sec. 4, and the step-0 answer readout of App. C.2.1 is predictive because it measures the composed answer directly.

C.3 Oracle representations (Sec. 4.1)

C.3.1 Injection procedure and controls

Fig. 3 compares three conditions, each with 6 PT seeds $\times$ 4 FT seeds under the same protocol: no coordinate injection (Base), shuffled-coordinate injection (Shuffle), and true-coordinate injection (Oracle). Their final FT generalization scores are 0.17 / 0.17 / 0.90, respectively, with in-distribution accuracy of 0.93 / 0.87 / 0.97.

Procedure. Each city's true coordinates, normalized to $[-1, 1]^2$ in a fixed global frame, are mapped by a learned linear projection $\mathbb{R}^2 \to \mathbb{R}^{128}$ and added to the residual stream at the city's last ID token, at the input of every decoder layer after the first, so that layer 0 receives only the token input. Coordinates are injected throughout PT and FT; entities defined in context receive their in-context coordinates. The shuffle control applies a fixed random permutation to the world cities' coordinates. Each city therefore receives a fixed incorrect coordinate pair, while the token stream is identical to that of the oracle run.

Training protocol. These runs use a separate, fully seeded implementation on the same world of 5,175 cities. It differs from the setup of Sec. 2 in the PT tasks (eight tasks: angle, perimeter, triangle

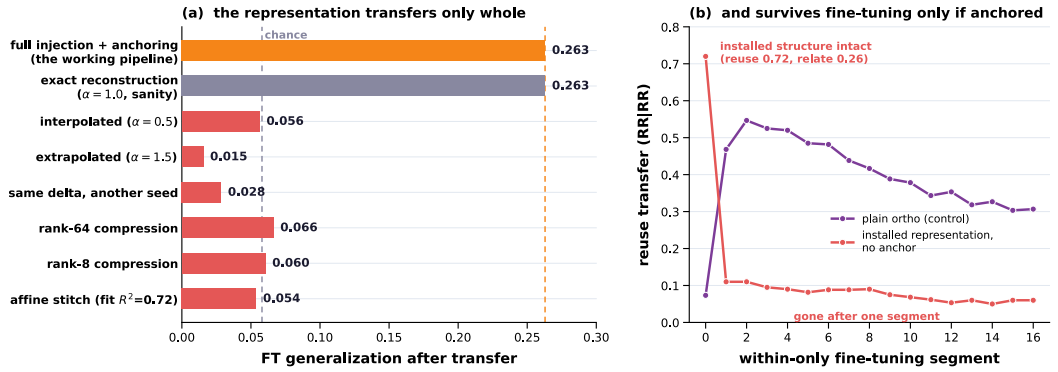


Figure 15: **Transfer alternatives and the Stage 2 distillation loss.** (a) Interpolation, extrapolation, applying the representation change to another recipient, low-rank compression, and affine mapping all give low FT generalization; only exact reconstruction matches the pipeline. (b) One segment of plain within-group FT reduces performance after distillation. Before that segment, the recipient has reuse of 0.72 and FT generalization of 0.26.

area, maximum and minimum pairwise distance, convex hull area and perimeter, and shortest open path), in the initialization scale (0.02), in the PT schedule (learning rate 3×10^{-4}, 4 epochs), and in FT, which mixes 20% PT replay into the FT data and runs for 40 epochs. All three conditions share this protocol, so the comparison between them is controlled.

Test-time removal control. The oracle model reads the coordinates from the injected channel at the first layer that receives them rather than from the city tokens, and removing the injection at test time reduces performance to chance or below, so the supplied representation is the one in use. The oracle therefore provides an upper bound on the FT generalization attainable with an externally supplied representation.

C.4 Representation distillation (Sec. 4.2)

C.4.1 Representation distillation ablations

Fig. 15 tests alternatives to the two-stage representation-distillation procedure of Sec. 4. Panel (a) compares transfer methods by FT generalization, with Δ denoting the representation change produced by initial distillation. Reproducing this change exactly ($\alpha = 1.0$) matches the pipeline (0.26 under the shorter protocol used here; the full pipeline of Sec. 4 reaches 0.32 on this recipient). The alternatives give low scores: half the change ($\alpha = 0.5$, 0.06), an overshoot ($\alpha = 1.5$, 0.02), the same Δ applied to a different recipient (0.03), rank-64 or rank-8 compression of the distilled representation (0.07/0.06), and mapping the teacher representation through a fitted affine transformation ($R^2 = 0.72$, 0.05). Thus, none of the tested approximations reproduces the transfer achieved by exact reconstruction. Panel (b) tests whether the distillation loss must be retained in Stage 2. The recipient initially has reuse of 0.72 and FT generalization of 0.26. After one segment of plain FT on within-group data, the plotted evaluation score declines to 0.11, whereas the plain ortho control reaches 0.55. This comparison supports retaining the distillation loss during Stage 2.

A second set of ablations perturbs the teacher representation before distillation, then measures the recipient's FT generalization under the same two-stage pipeline (Fig. 16). Whitening reduces FT generalization (0.27 $\rightarrow$ 0.22), and moderate linear noise gives 0.21. Larger reductions result from a random rotation (0.10–0.12), rank truncation, including retaining 40 of 128 dimensions (0.12), or rescaling individual directions (0.11–0.16). These results indicate that transfer is sensitive to the teacher representation's orientation and to dimensions removed by truncation, including low-variance dimensions, properties to which the spectrum-based summaries of App. C.2.2 are insensitive.

Table 5 removes components of the pipeline for two bad-tier recipients using the same initial distillation protocol. Without initial distillation, regularized FT remains low (0.09). For one recipient, initial distillation followed by plain FT reaches 0.20 before declining to 0.13. Fixed-target

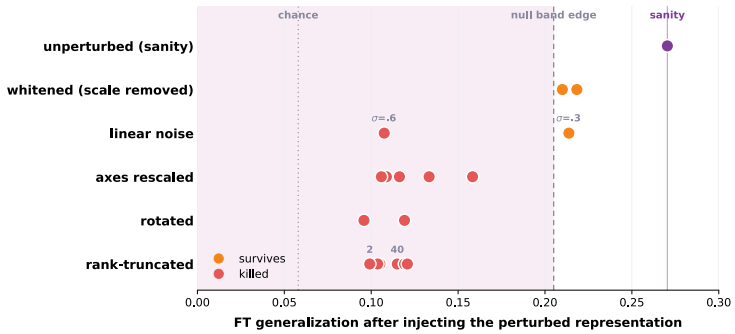


Figure 16: **Perturbing the teacher representation.** Each point shows the recipient's FT generalization after distillation from a perturbed teacher representation, using the same pipeline; perturbations are grouped by family. Whitening retains more of the transfer gain than rescaling individual directions, rotating, or truncating rank.

Table 5: **Distillation stage ablations.** FT generalization for variants of the two-stage pipeline on two bad-tier recipients using the same initial distillation protocol. The full-pipeline row is the matched grid of Fig. 4b (mean of four FT seeds); the other rows are single runs under the same 24-segment protocol.

pipeline variant	recipient A	recipient B
plain FT (no intervention)	0.09	0.07
matched additional PT, no distillation	0.10	—
regularized FT only, no initial distillation	0.09	0.10
initial distillation, then plain FT	$0.20 \rightarrow 0.13$	0.10
distillation + fixed-target regularization	0.15	0.12
distillation + periodic regularization	0.16	0.17
distillation + teacher target (full pipeline)	**0.28**	**0.27**

regularization toward the post-distillation state gives 0.12–0.15, and periodic regularization gives 0.16–0.17. Retaining the distillation loss toward the teacher representation gives the highest scores for both recipients (0.28 and 0.27), whereas matched additional PT without distillation gives 0.10. These comparisons show that both initial distillation and the choice of target during FT contribute to the observed gains.

C.4.2 Distillation across all seeds

Fig. 17 distills the representation of each of 274 teachers into the same bad-tier recipient, whose baseline FT generalization is 0.07. All teachers are distilled under the same fixed protocol, a short initial distillation stage followed by FT with the distillation loss. Every teacher improves the recipient's score, and the gain correlates with the teacher's regularized FT generalization ($r = 0.70$ on the linear scale of Fig. 17; 0.65 against the logarithm of the score, as plotted in Fig. 4c).The highest-scoring teachers raise the recipient above the null band, with a maximum score of 0.32 under this short protocol; the full two-stage pipeline in Sec. 4 uses longer initial distillation and a tuned second stage.

Fig. 18 compares transfer across teachers and recipients. Panel (a) distills each of 29 teacher representations into three bad-tier recipients using a fixed short protocol (50k-step initial distillation, 16 regularized segments). Scores report the gain over each recipient's baseline FT generalization. A subset of teachers improves all three recipients, with gains up to +0.19. Teacher rankings are similar across recipients (pairwise Spearman $\rho = 0.43$–0.61), but transfer gains have little association with the teacher's plain FT generalization ($r = -0.21$). In particular, the best-performing seed under plain FT yields a low transfer gain; the teacher's regularized FT generalization is a better predictor of transfer. Panel (b) uses the full two-stage pipeline to distill from one good-tier teacher

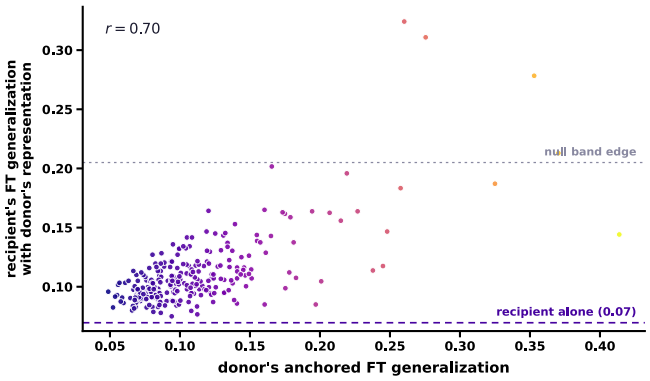


Figure 17: **Distillation across all seeds.** Each teacher's representation is distilled into the same bad-tier recipient using a fixed protocol. The recipient's FT generalization correlates with the teacher's regularized FT generalization ($r = 0.70$; color: teacher score). The highest-performing teachers raise the recipient above the null band.

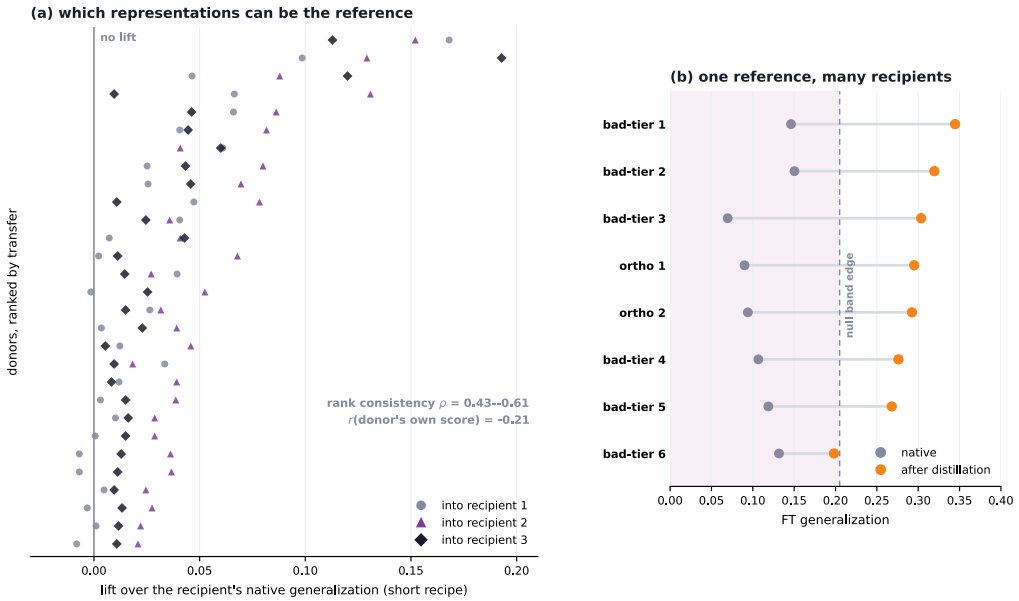


Figure 18: **Teachers and recipients.** (a) Each of 29 teachers is distilled into three recipients using a fixed short protocol; scores show the gain over each recipient's baseline FT generalization, and markers identify recipients. Teacher rankings are similar across recipients, but transfer gains have little association with the teacher's plain FT generalization. (b) The full two-stage pipeline distills one good-tier teacher into eight recipients; grey shows baseline FT generalization and orange shows FT generalization after distillation. Seven of eight recipients exceed the null band.

into eight recipients: six bad-tier seeds and both ortho models. Seven improve from their baselines (0.07–0.15) to 0.27–0.35, above the null band, and the eighth reaches the band's upper edge (0.20).

C.5 The ortho control (Sec. 4.3)

C.5.1 Generalization and reuse under different PT policies

Fig. 19 compares the ortho control from Sec. 4.3 with the shared and weak PT policies. Under the same FT protocol, the shared model, which has a joint representation, shows across-generalization (0.40) but low reuse (0.11). The ortho model, which has separate representations, shows low across-

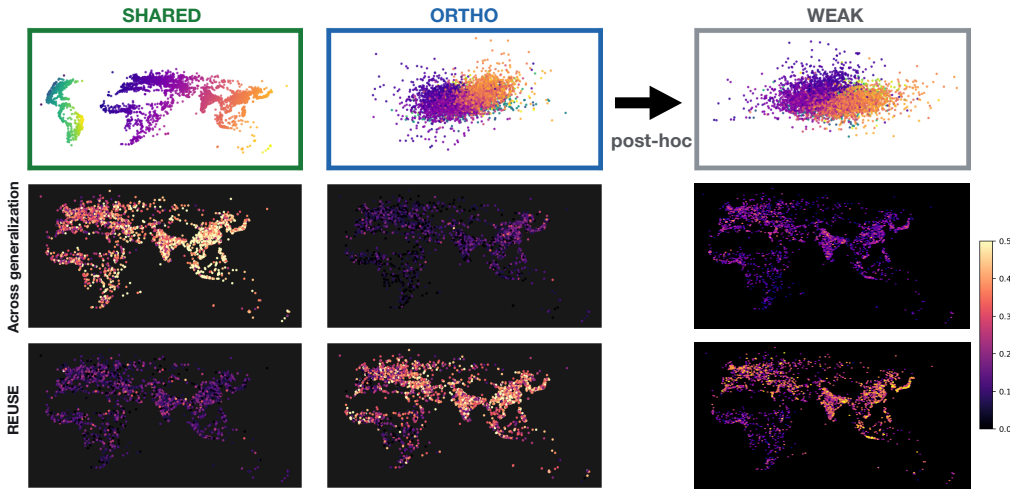


Figure 19: **PT policies, across-generalization, and reuse.** The top row shows final layer-5 maps for one model per policy. Shared forms a joint representation; ortho forms separate representations; weak adds cross-group training after ortho PT, producing less distinct but still separate representations. The lower rows show per-entity FT accuracy on the across-generalization and reuse tests. The shared model scores higher on generalization and lower on reuse than the ortho model. Weak retains some reuse but remains low on across-generalization, consistent with the broader results in Table 6.

generalization (0.09) but higher reuse (0.31). The weak model adds cross-group training after ortho PT; its representations become less distinct but remain separate. It also shows low across-generalization (0.11; 0 of 25 runs exceed the null band). Most weak runs retain reuse (14 of 25); the illustrated model scores 0.11 on this test. Adding cross-group data after ortho PT therefore preserves reuse in most runs but does not produce across-generalization.

The two tests separate the policies in opposite directions: ortho FT generalization remains in the null band (0.05–0.20 across seeds and FT protocols), while shared runs span the range of outcomes in Sec. 3. In five matched seed pairs evaluated with a shorter transfer protocol, shared models have low reuse (0.04–0.09), and the corresponding ortho models score higher in every pair (sign test, $p = 0.031$). App. C.5.3 discusses how within-group data leave the cross-group relation underconstrained. App. C.5.2 repeats both tests for all three policies across PT bases and FT seeds. Across every pretrained geometry we tested, across-generalization and reuse never co-occur: the geometry FT starts from selects which form of generalization FT can achieve.

C.5.2 Replications of the two transfer tests across seeds

Table 6 summarizes replications of both tests by PT policy. No ortho or weak run exceeds the null band on across-generalization (0 of 30 and 0 of 25; the exact 95% upper bounds on the underlying rates are 10% and 11%). No shared run exceeds it on reuse (0 of 25, bound 11%). Shared FT generalization varies across seeds as described in Sec. 3, while most ortho runs (15 of 25) and weak runs (14 of 25) exceed the null band on reuse. Adding cross-group data after ortho PT therefore preserves reuse in the majority of runs but does not yield across-generalization. Replaying general data during FT at ratios up to 0.5 does not change these outcomes. This separation is a property of the PT policy, not of a particular run.

C.5.3 Identifiability of the cross-group relation

The ortho corpus constrains within-group structure without directly specifying cross-group relations, leaving the groups' relative alignment underconstrained. Some PT tasks, including Manhattan distance, and the FT task, the *signed* angle, are nevertheless not invariant to independently rotating or reflecting one group; these tasks restrict the admissible transformations, and a formal identifiability statement would specify the ambiguity that remains. The argument characterizes what the available data leave undetermined rather than establishing impossibility. Empirically, all 30 ortho replications

Table 6: **Across-generalization and reuse across seeds.** Across replications of both tests by PT policy, no ortho or weak run exceeds the null band (0.20) on across-generalization, and no shared run exceeds it on reuse.

policy	test	runs	range	above the null band
shared	across-generalization	278	0.07–0.50	62 (the lottery, Sec. 3)
shared	reuse	25	0.07–0.12	0
ortho	across-generalization	30	0.05–0.20	0
ortho	reuse	25	0.13–0.46	15
weak	across-generalization	25	0.06–0.19	0
weak	reuse	25	0.11–0.39	14

All rows use the full-length protocol, with five bases × five FT seeds per cell (ortho across-generalization includes five earlier runs). Reuse is RR|RR accuracy after FT on group-A instances only. An earlier, shorter transfer protocol (eight shared/ortho runs, three weak) gave comparable results for shared and ortho, with weak reuse of 0.04–0.05. The null band is calibrated on across-generalization and is applied unchanged to reuse.

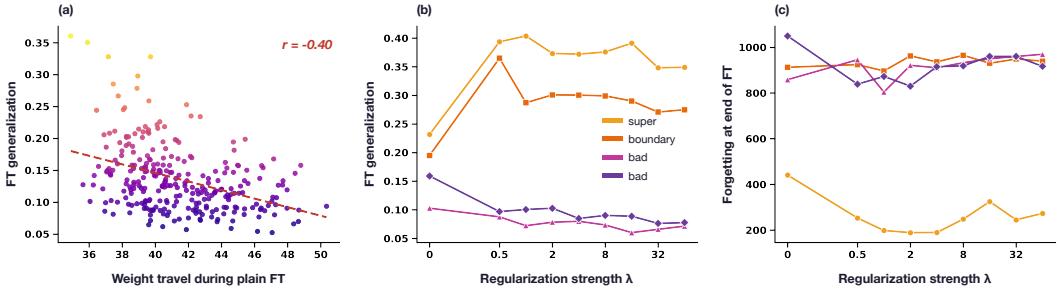


Figure 20: **Parameter change and fixed-target regularization.** (a) Across 276 seeds, parameter change during plain FT is negatively correlated with FT generalization ($r = -0.40$). (b) Four seeds that end plain FT ($\lambda = 0$) at 0.10–0.23; by tier, which is assigned by the peak reached under plain FT, they are one super, one boundary, and two bad seeds. (c) Forgetting over the same sweep, measured as the held-out error on the PT distance task at the end of FT (coordinate units). Without replay data, regularization reduces this error for the super seed; the other three retain high error.

remain at or below the null band; the input-independent marginal-answer optimum is about 0.10. The weak policy adds cross-group data after PT, providing information about relative alignment, but none of its runs exceeds the null band (0 of 25, App. C.5.1).

C.6 Representation regularization (Sec. 5)

C.6.1 Reshaping and forgetting under fixed-target regularization

Under plain FT, total parameter change is negatively correlated with FT generalization ($r = -0.40$, Fig. 20a). The difference between tiers is concentrated in the embedding through layer 5; layer 6 and the output head show similar changes across tiers (App. C.6.2). To test the effect of limiting these changes, we select four seeds that end plain FT in the boundary or bad tier (one super, one boundary, and two bad seeds by tier; 0.10–0.23) and regularize their layer-5 representations toward their pretrained states at increasing strength λ (Fig. 20b). The super and boundary seeds improve to 0.30–0.40, whereas the two bad seeds approach chance. For the super seed, forgetting also decreases: the held-out error on the PT distance task at the end of FT, measured in the world's coordinate units, falls from 435 to 183 at a tuned strength, while the other three seeds' forgetting is unchanged (Fig. 20c). Thus, fixed-target regularization can improve FT generalization and reduce forgetting without replay data, but its effect depends on the pretrained representation; the regularized results across all seeds in Sec. 5 use the periodic variant (App. C.6.3).

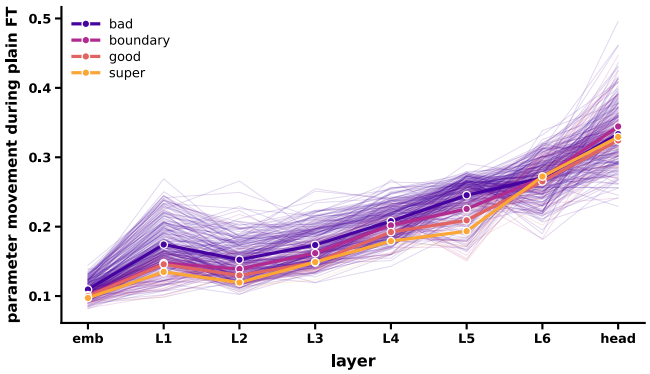


Figure 21: **Per-layer parameter change.** Relative parameter change during plain FT by layer, with thin lines for individual runs and bold lines for tier means. The tiers differ in the embedding through layer 5 and show similar parameter changes at layer 6 and the output head.

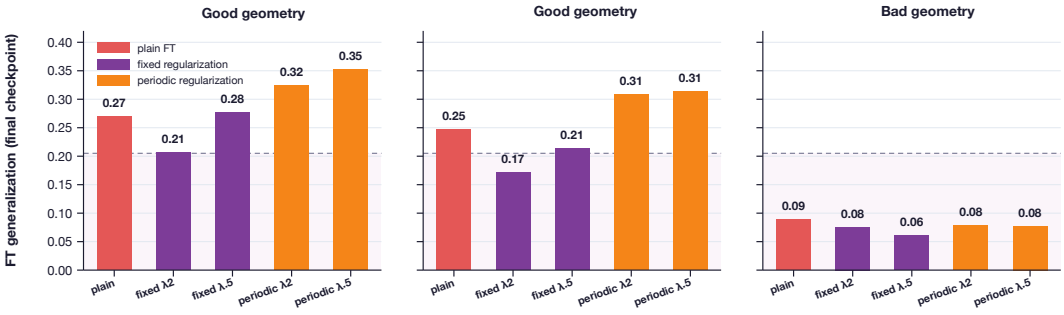


Figure 22: **Plain FT, fixed-target, and periodic regularization.** Final-checkpoint FT generalization under plain FT, fixed-target regularization, and periodic regularization at two strengths. For the good-tier seeds, periodic regularization gives the highest final scores, while strong fixed-target regularization lowers both seeds' tiers. The bad-tier seed remains within its tier under all conditions.

C.6.2 Per-layer parameter change

Fig. 21 reports parameter change during plain FT (Sec. 5) by layer for 276 of the 278 runs, those with saved parameter trajectories. Total parameter change correlates with FT generalization at $r = -0.40$ across these runs and at -0.30 among the runs that fully fit the FT set, so the association is not confined to runs that underfit the FT set. From the embedding through layer 5, runs with high FT generalization show 5–15% less parameter change than typical runs, and bad-tier runs show the largest changes. Layer 6 and the output head show similar parameter changes across tiers.

C.6.3 Plain FT, fixed-target, and periodic regularization

Fig. 22 compares final-checkpoint FT generalization under plain FT, fixed-target regularization, and periodic regularization using the same 48-segment protocol. Fixed-target regularization uses the pretrained representation throughout FT, whereas periodic regularization updates the target from the current model at each segment; we evaluate both at two strengths on two good-tier seeds and one bad-tier seed. For the good-tier seeds, periodic regularization gives the highest final scores (0.31–0.35). Plain FT reaches 0.25–0.27, below its earlier peak, and fixed-target regularization reaches 0.17–0.28; at the stronger fixed-target setting, both good-tier seeds fall below their original tier. The bad-tier seed remains at 0.06–0.09 under all conditions (Sec. 5). These results indicate that periodically updating the target preserves FT generalization better than holding the representation near a fixed pretrained state.

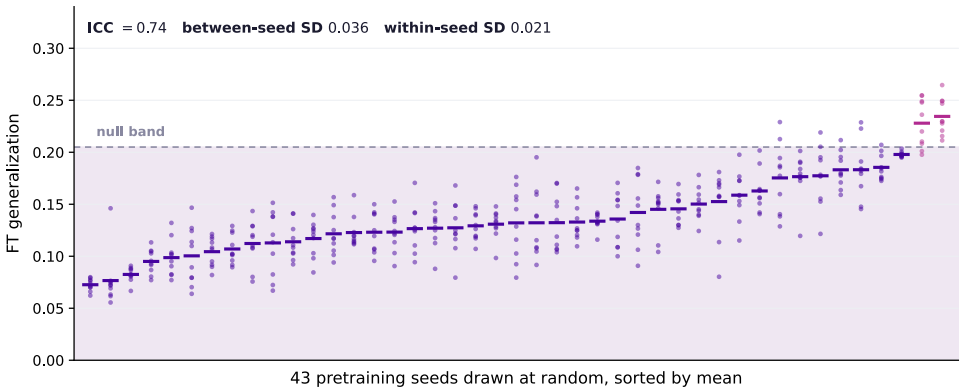


Figure 23: **Variation across randomly sampled PT seeds.** Forty-three PT seeds, each fine-tuned under ten FT seeds (dots), sorted by their mean FT generalization (bar) and colored by the mean's tier. The between-seed standard deviation (0.036) exceeds the within-seed standard deviation (0.021), with an intraclass correlation of 0.74.

C.6.4 Weight-space and linear-probe baselines

We compare two methods for constraining deviation from the pretrained model during FT: L2-SP, a weight-space penalty toward the pretrained parameters (Li et al., 2018), and LP-FT, linear probing followed by FT (Kumar et al., 2022). We evaluate both on seeds spanning the tiers under the standard FT protocol. Bad-tier seeds remain at or below 0.11 FT generalization with either method, whereas representation distillation raises seeds in the same tier to 0.27–0.28 (Table 5). L2-SP at $\lambda = 10^{-3}$ is indistinguishable from plain FT within measurement noise, while a stronger penalty ($\lambda = 10^{-2}$) impairs learning, consistent with the intermediate optimal strength in Fig. 20.

C.7 Statistical notes (Secs. 3–5)

This section reports uncertainty estimates for the paper's main quantitative claims, computed from the runs described above without additional training.

Variation across PT and FT seeds. Treating the 7×14 grid of Table 4 as a one-way random-effects design, the PT seed accounts for 89.9% of the variance in FT generalization (intraclass correlation 0.899; $F(6, 91) = 125$, $p < 10^{-15}$). Because those seven seeds were selected to span the tiers, we repeat the analysis on 43 randomly sampled PT seeds, each fine-tuned under 10 FT seeds (430 runs). In this sample, the intraclass correlation is 0.736 ($F(42, 387) = 28.9$, $p < 10^{-15}$); the between-seed standard deviation is 0.036 against a within-seed (FT) standard deviation of 0.021, and the median seed varies by 0.06 across its ten FT seeds (Fig. 23). In both samples, variation across PT seeds exceeds variation across FT seeds.

Selection by PT loss and answer readout. Across all seeds ($n = 278$), final PT loss correlates with FT generalization at $r = -0.41$ (95% CI -0.50 to -0.31). The lowest-loss decile contains 2 high-performing runs among 27 runs (high-performing = super or good tier; 5.4% base rate; $p = 0.43$). The top answer-readout decile contains 7 such runs among 27 ($p \approx 5 \times 10^{-4}$ against the base rate). The answer-readout decile is thus enriched relative to the base rate; the bootstrap interval for the direct difference between deciles (95% CI -0.04 to 0.41) does not separate the two selectors, and these results complement the causal interventions in Sec. 4.

In-distribution accuracy and FT generalization. In-distribution accuracy under regularized FT correlates with regularized FT generalization at $r = 0.67$ (0.61–0.73), compared with $r = 0.33$ (0.22–0.43) for in-distribution accuracy under plain FT. A test of dependent correlations rejects equality (Steiger $z = 6.2$, $p < 10^{-9}$, $n = 278$). For completeness, plain FT in-distribution accuracy correlates with plain FT generalization at $r = 0.51$ and regularized FT generalization at 0.33; regularized FT in-distribution accuracy correlates with regularized FT generalization at 0.67 and plain FT generalization at 0.33. Correlations are stronger within each protocol than across protocols, and

the regularized pair has the strongest correlation. Under regularized FT, in-distribution accuracy also remains informative because training does not force every run to fit the FT set (Fig. 6).

Exact binomial bounds. For conditions with no observed successes (Table 6), we report exact binomial upper bounds: no ortho run shows across-generalization in 30 replications (rate $< 10\%$ at 95%), no weak run does so in 25 ($< 11\%$), and no shared run shows reuse in 25 ($< 11\%$). The five matched reuse pairs have the same ordering (sign test, $p = 2^{-5} = 0.031$).

C.8 Behavior distillation

Behavior distillation replaces the layer-5 representation objective with $\mathrm{KL}(p_\theta \,\|\, p_{\theta_{\text{good}}})$ over every real token at $\tau = 1$. The corpus, step count, schedule, and FT protocol are identical to those used for representation distillation. Behavior distillation reduces KL divergence from the teacher and increases top-1 agreement, whereas the gain in FT generalization is specific to representation distillation.

	s43	sd816	ortho
KL to teacher, before $\to$ after	$0.060 \to 0.026$	$0.095 \to 0.021$	$0.275 \to 0.020$
top-1 agreement, before $\to$ after	$0.81 \to 0.83$	$0.81 \to 0.84$	$0.76 \to 0.81$
FT generalization, plain FT	0.096	0.075	0.080
FT generalization, behavior distillation	0.073	0.073	0.078
FT generalization, representation distillation	0.280	0.271	0.320
FT generalization, behavior at $5\times$ weight	0.110	0.104	—

The loss is evaluated at $\tau = 1$ over a 73-token vocabulary.

D Visualizations

D.1 Representation gallery across seeds

Figs. 24–30 show each seed's layer-5 representation at ten PT checkpoints, with each panel projected onto a basis fitted at that checkpoint. Runs that differ only in random seed develop visibly different maps early in PT, and these differences persist. The gallery provides a qualitative view of the variation in representations studied through the interventions in Sec. 4. FT generalization depends on structure that neither the projected maps nor static geometric summaries of the representation resolve (App. C.2.2). Full-resolution maps and all PT checkpoints will be released for every seed.