LONG-TAILED ENTITY DISTRIBUTIONS PRESERVE WORLD MODEL PLASTICITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Our world is ever-changing and, while language models can incorporate new information flexibly in-context, they face difficulties doing so *in weights*, in a way that generalizes. We show that the pre-training data distribution is a key lever: long-tailed entity distributions preserve a model's ability to integrate new entities into its world model. We study this in a controllable synthetic world where cities with known geometry underlie multiple relational tasks. We first show that models use the representational geometry of entities when solving tasks, and that how well a new entity is integrated into the world geometry predicts its cross-task generalization ($r^2 = 0.93$). We then find that under uniform pre-training, the world representation forms early but loses plasticity: its ability to integrate new entities via gradient descent decays with continued training, even as pre-training loss decreases. Pre-training with Zipfian entity frequencies counters this, improving new-entity integration $3\times$ on average and $4.9\times$ on the hardest task, and outperforming 15 alternatives including meta-learning. We hypothesize that sparse exposure keeps some entities effectively new (i.e. incompletely learned), providing continued practice integrating them throughout pre-training. At high Zipf α, however, tail learning and world-model quality deteriorate. In all, we establish that long-tailed entity distributions help maintain the plasticity of entity representations and reveal a trade-off between forming a world model and keeping it open to change.

1 INTRODUCTION

Bob: I'm moving to Hoboken.
Alice: Where's that?
Bob: Right next to New York.
Alice: Oh wow, you're moving far from here!

This exchange is unremarkable, but it reveals how naturally humans integrate new entities into their knowledge base. Alice knows only a single relation between Hoboken and a familiar city, but that suffices to place it on an internal map of the world, from which its relation to everything else, including Alice's own location, follows without further instruction. Language models show similar abilities when new information is provided in-context. However, when a new entity must instead be learned *in weights*, the acquired knowledge often fails to propagate beyond the form in which it was trained (Onoe et al., 2023; Berglund et al., 2024; Cohen et al., 2024; Zhong et al., 2023).

Existing approaches treat this as a post-training problem. Knowledge editing methods target individual facts (De Cao et al., 2021; Mitchell et al., 2022a;b; Meng et al., 2022; 2023), yet edits often fail to propagate to their implications (Cohen et al., 2024; Zhong et al., 2023), and where a fact is localized does not reliably indicate where it is best edited (Hase et al., 2023). Gradient-based fine-tuning is more general but data-inefficient: new facts are learned slowly, often only in the form they are presented, and can corrupt existing knowledge (Allen-Zhu & Li, 2023; Ovadia et al., 2024; Gekhman et al., 2024; Zucchet et al., 2025; Park et al., 2025; O'Neill, 2026). Both families treat updating as something done to an already trained model.

This work. We instead ask whether the problem lies further upstream: to be updated well, a model may need to be *trained* to be updatable, with this ability determined during pre-training by the data itself. The pre-training distribution is known to determine what later training can extract and learn (Allen-Zhu & Li, 2023; Zucchet et al., 2025), and long-tailed distributions in particular change what

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.

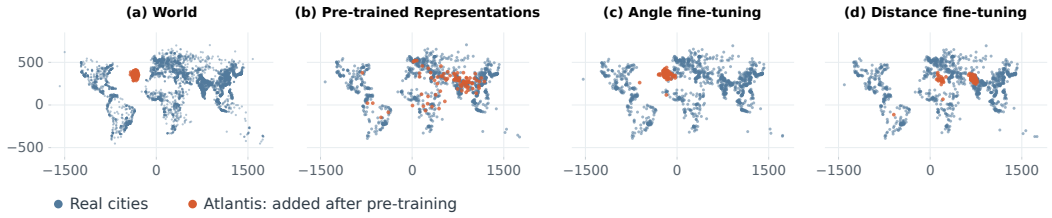


Figure 1: (a) Coordinates of 5,075 real cities (blue) in the world and 100 Atlantis cities (orange), introduced after pre-training is complete. (b–d) Linearly decoded coordinates from model representations of cities. (b) After pre-training with a uniform entity distribution. (c,d) Fine-tuning on Atlantis entities using Angle or Distance tasks. Angle produces a more geographically aligned Atlantis cluster.

and how models learn (Chan et al., 2022; Wang et al., 2026; Huang et al., 2026). We study this through *representations*: integrating an entity means placing it correctly relative to familiar entities in a shared representation space, as Hoboken is placed on Alice's map. Studying this question in natural data is difficult, however: it is often unclear how far a new fact *should* propagate (Hase et al., 2024), and the representation it should be integrated into is unknown. We therefore turn to a synthetic setting (Park, 2026) in which the world and the representation it should induce are known, and the data distribution can be controlled exactly.

We study this question in the synthetic city world of Park (2026) (Fig. 1), where integration can be read directly from representations. We find that standard pre-training forms a world representation that soon hardens against new entities, and that a long-tailed, Zipfian entity distribution preserves its plasticity, up to a limit beyond which the tail is neglected entirely. **Overall, our contributions are:**

1. **Entity representations are causal and predict generalization (Sec. 3, Fig. 2, 3)**: Editing city representations along the world geometry changes model outputs as expected from the geometric transformation and fine-tuned new-entity representations are directly usable by the original pre-trained model. Furthermore, representational integration of new entities predicts their cross-task generalization ($r^2 = 0.93$), while performance on the fine-tuning task does not ($r^2 = 0.15$).
2. **Plasticity decays during pre-training even as the world model improves (Sec. 3, Fig. 4)**: Under uniform pre-training, the world model forms early, but subsequent training makes new entities harder to integrate. An accurate world model and the ability to edit it are distinct properties.
3. **Zipfian entity sampling preserves plasticity (Sec. 4, 5, Fig. 5, 7, 10)**: We show that sampling entities from a Zipfian distribution throughout pre-training largely prevents this loss of plasticity. With $\alpha = 1.25$, new-entity reconstruction error drops $3\times$ on average across fine-tuning tasks and $4.9\times$ for distance. This data intervention outperforms 15 alternative interventions, including new-entity MAML (Finn et al., 2017). These findings generalize to relocating existing entities and to fine-tuning tasks held out from pre-training. We understand this as the long tail keeping entities effectively new: although every entity appears early, rare ones are seen so sparsely that the model keeps integrating them throughout pre-training. Consistent with this, two other ways of keeping entities effectively new, introducing them late (staging) or keeping them rare (scarcity), each recover most of the benefit.
4. **A trade-off between world-model formation and plasticity (Sec. 4, Fig. 5a)**: Near-uniform sampling makes new entities hard to integrate; excessive skew ($\alpha \gtrsim 1.5$) sharply worsens real-city reconstruction. Intermediate skew best balances world-model formation and later integration.

2 Setup

We adopt the synthetic world modeling setting of Park (2026), illustrated in Fig. 1. The world contains 5,075 *real* cities, each associated with a random ID (e.g. `c_0304`) and underlying 2D coordinates (e.g. $x = -1224, y = 378$). Autoregressive Transformer models are pre-trained on seven geometric tasks operating on city IDs, such as distance, angle, and perimeter. We list all tasks and their details in App. B.1. For example, one training sequence could be "`dist(c_1226,c_3022)=485`", which is tokenized at the character level and corresponds to "the distance between San Francisco and New York is 485 units". After pre-training, the world is expanded by an additional 100 *Atlantis* cities —

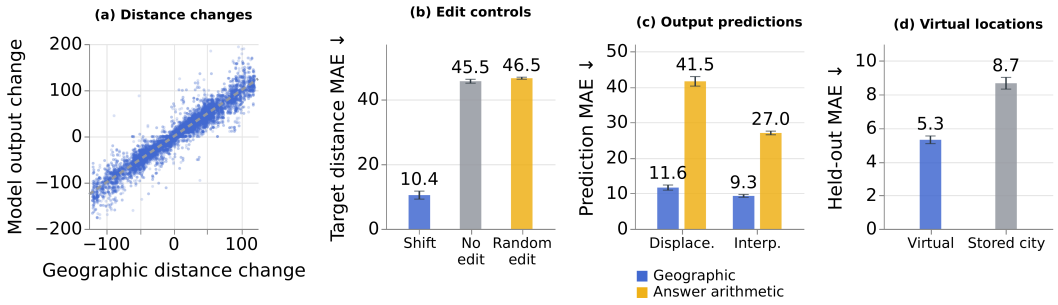


Figure 2: **Geometric interventions affect distance computation.** (a) Individual model-output and geographic distance changes; the dashed line indicates equality. (b) Displacement versus unchanged and random-edit controls. (c) Geographic predictions versus endpoint-answer arithmetic. (d) Continuous-location versus stored-city fits on held-out anchors.

i.e. located in the fictional island of Atlantis. The model is fine-tuned on these Atlantis cities using a single task and then evaluated on whether the new knowledge transfers to other tasks.

This setting separates the underlying world (cities at coordinates) from the tasks that generate observations about it. Park (2026) finds that multi-task training produces representations of cities aligned with the world's geometry, without direct coordinate supervision. We call these *world representations* when city coordinates can be linearly decoded, and test whether models use them for task computation in Sec. 3.

We use **representational integration** to describe whether fine-tuned Atlantis cities are incorporated into the learned world model, and **cross-task generalization** to describe the model's ability to use the acquired knowledge across tasks. Park (2026) finds that Atlantis integration and generalization both depend on the fine-tuning task. They point out that distance fine-tuning results in particularly poor integration and generalization.

Experimental details We train six-layer Qwen2 models (Yang et al., 2024) from scratch on all seven tasks and fine-tune on Atlantis using one task, with three seeds. We measure representational integration by Euclidean coordinate reconstruction error: we fit a linear probe on real-city representations and evaluate it on Atlantis cities, whose coordinates never enter the probe fit. To form these representations, we concatenate the layer 5 activations from the tokens in the prompt corresponding to the city's final digit and subsequent comma. We measure cross-task generalization using the normalized improvement (NI) metric from Park (2026), averaging across non-fine-tuning tasks, where NI $= 0$ (baseline) and NI $= 1$ (oracle) refer to the performance of models pre-trained without and with Atlantis. Activation patching (Zhang & Nanda, 2024) intervenes on layer 4 activations at matching source/target token positions; implementation details appear in App. B.

3 World Representations Form Early, but Plasticity Decays

In this section, we first investigate whether the representational geometry of cities contributes to how models compute tasks, whether newly acquired entity representations can be used by those computations, and how pre-training shapes the ability to acquire such representations.

Representational geometry and task computations To understand whether pre-trained models actually solve tasks using the geometric structure we can decode from city representations, we test whether geometric transformations of city representations induce corresponding changes in model predictions. Let h denote a city's representation (layer 4 activations, as described in Sec. 2) and z denote its world coordinates. We test the following two transformations. We include additional details in App. C.

First, we sample three distinct cities (with IDs `A`, `B`, `C`) and consider a *displacement* of `C` to the hypothetical location $z' = z_C + \lambda(z_B - z_A)$. For a recipient city `X` and anchor city `Y`, we run the same prompt "`dist(X, Y)`" twice, replacing h_X with h_C in one run and $h_C + \lambda(h_B - h_A)$ in the other.

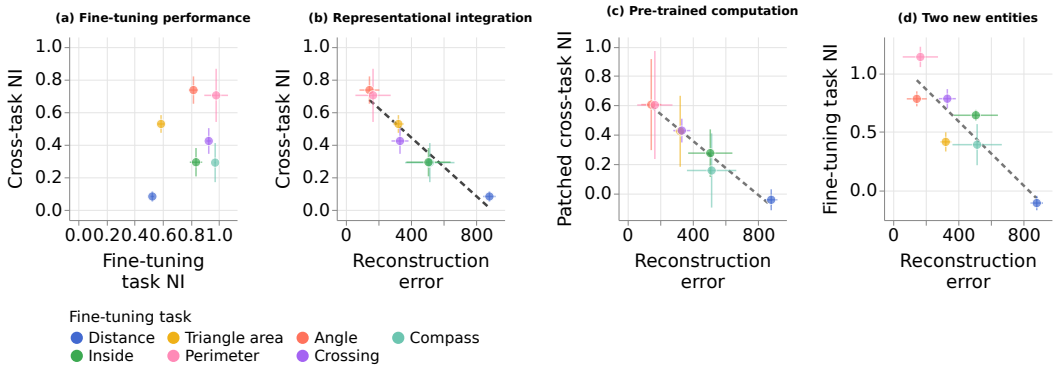


Figure 3: **New-entity integration is associated with generalization and computational reuse.** (a) Performance on the Atlantis fine-tuning task versus mean performance on the other six tasks. (b) Atlantis coordinate reconstruction error versus cross-task performance. (c) Reconstruction error versus cross-task performance after patching layer-4 Atlantis activations from the fine-tuned model into the original pre-trained model. (d) Reconstruction error versus performance on the fine-tuning task with two Atlantis cities in the input.

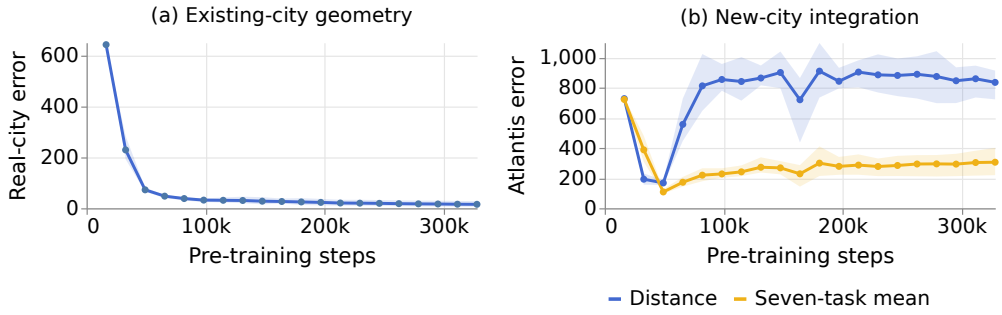


Figure 4: **World geometry improves while adaptability declines.** Reconstruction error for (a) real cities during pre-training and (b) Atlantis after distance fine-tuning or averaging all seven fine-tuning tasks at each checkpoint. Lines show three-seed means; ribbons show the seed range.

Second, we consider an *interpolation* between A and B, with target location $z' = (1-\lambda)z_A + \lambda z_B$. We similarly replace h_X with h_A or $(1-\lambda)h_A + \lambda h_B$ in separate runs of the same prompt.

Displacing city representations changes the model's distance answers in line with the corresponding geographic changes (Fig. 2a). Relative to patching h_C, these changes match the geographic changes with R^2 of 0.905 (evaluated against the identity line). The transformed states also yield lower error relative to the displaced target than unchanged states or equal-norm random edits (Fig. 2b). For both displacement and interpolation, geographic predictions explain the outputs better than applying the same arithmetic directly to the model's endpoint distance answers (Fig. 2c). To test whether the outputs instead reflect an existing city, we fit a continuous location and select a stored city using generated distances to eight anchors. The continuous location better predicts outputs on eight held-out anchors (Fig. 2d).

These interventions support a computational role for city geometry in distance prediction.

New-entity integration and computational reuse Having established that pre-trained computations respond to geometric transformations of entity representations, we ask whether fine-tuning makes new entities accessible to those computations. In Fig. 3(a), models generally improve on the task used to introduce Atlantis cities, but this improvement does not reliably predict cross-task generalization ($r^2 = 0.15$). In contrast, lower Atlantis reconstruction error is strongly associated with better cross-task generalization (Fig. 3(b); $r^2 = 0.93$). These observations suggest that fine-tuning

produces solutions with different degrees of reuse: some incorporate new entities into representations shared across tasks, while others remain specialized to the fine-tuning setting.

We directly test compatibility with existing computations by patching layer-4 Atlantis representations from each fine-tuned model, extracted under its fine-tuning-task prefix, into its original pre-trained model. We then evaluate the other six tasks. In Fig. 3(c), representations from models with better Atlantis integration yield better performance under this intervention ($r^2 = 0.96$). Thus, successful fine-tuning can produce representations usable by task computations learned before the model encountered Atlantis cities. This supports computational reuse without identifying geometric alignment as the unique mediator of generalization.

A solution specialized to the fine-tuning examples may also fail when the task remains unchanged. In Fig. 3(d), we evaluate the fine-tuning task on inputs containing two Atlantis cities, a combination under-represented during fine-tuning. Performance again correlates with representational integration ($r^2 = 0.78$). Integration is therefore associated with generalization across both tasks and entity combinations.

World representations and their adaptability are distinct These results identify a desirable fine-tuning outcome: new entity representations that existing computations can use. We next examine how the ability to achieve this outcome changes during pre-training. Under uniform pre-training, representations of existing cities become aligned with the world early in training (Fig. 4(a)). However, fine-tuning from successive checkpoints reveals worsening Atlantis reconstruction after distance fine-tuning as pre-training continues (Fig. 4(b)). Acquiring an accurate representation of the existing world therefore does not ensure that the model remains able to extend it. These observations motivate investigating pre-training procedures that preserve subsequent new-entity integration.

4 Long-Tailed Entity Distributions Preserve Plasticity

We ask whether the pre-training distribution can preserve later integration. MAML explicitly optimizes performance after adaptation (Finn et al., 2017), with updates that can favor within-task gradient agreement (Nichol et al., 2018). We instead vary entity frequency during ordinary pre-training: frequent entities may support shared task structure, while continued learning of rare entities may preserve the ability to extend it. Zipfian sampling provides this variation without an explicit meta-learning objective.

We sample cities with weights $p_i \propto r_i^{-\alpha}$ for randomly assigned ranks r_i, shared across tasks; $\alpha = 0$ gives uniform sampling. Atlantis reconstruction error decreases as α rises to approximately 1.25–1.40, then worsens along with real-city geometry (Fig. 5(a)). Moderate skew balances world-model formation with plasticity; excessive skew leaves the tail underlearned.

More frequent entities have lower reconstruction and training-corpus errors (Fig. 5(b,c)), but this advantage does not consistently extend to held-out queries (App. E.3). This suggests that fitting familiar entities and queries can coexist with poor task generalization, especially under strong skew.

In Fig. 5(d), changing from $\alpha = 0$ to $\alpha = 1.25$ lowers mean Atlantis reconstruction error for every fine-tuning task. Distance fine-tuning improves from 879 to 179 (a $4.9\times$ reduction); error averaged over the other six fine-tuning tasks falls from 331 to 125 (a $2.6\times$ reduction). In Fig. 5(e), distance fine-tuning also improves cross-task NI from 0.084 to 0.517. The behavioral gains are less uniform across the remaining tasks. We believe this results from already sufficient Atlantis integration in the uniform ($\alpha = 0$) setting. Decoded locations illustrate this improvement after distance fine-tuning (Fig. 6).

4.1 Comparisons against other methods

We compare Zipf sampling with 15 alternatives (Fig. 7), including MAML (Finn et al., 2017), warmup–stable–decay (Hu et al., 2024), shrink-and-perturb (Ash & Adams, 2020), and replay (Robins, 1995), alongside regularization and representation supervision. Our task-specific MAML episodes adapt to fresh entities, excluding Atlantis. Full recipes and per-task results appear in App. F.

Across seven fine-tuning tasks, error falls from 409 under uniform pre-training to 133 with Zipf sampling, compared with 252 for the strongest alternative (the insertion-sharpness penalty) and 365

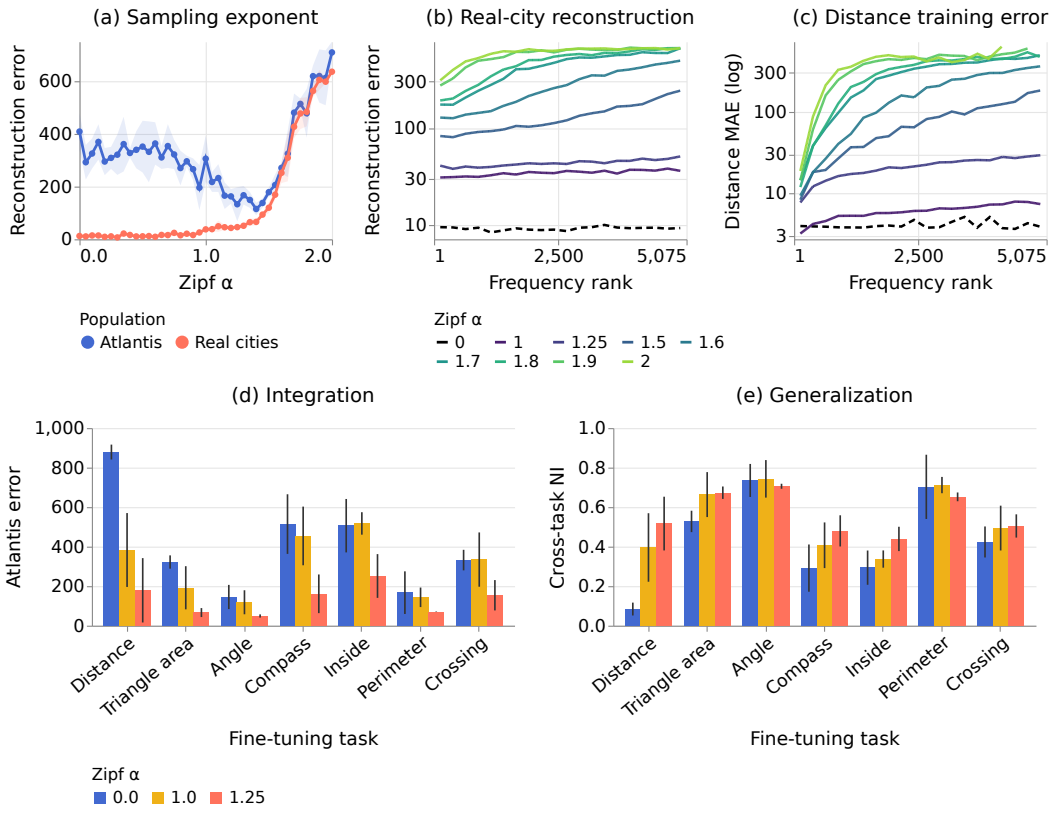


Figure 5: **Entity frequency shapes existing-world learning and subsequent integration.** (a) Atlantis and held-out real-city reconstruction errors after fine-tuning, averaged across seven fine-tuning tasks. (b,c) Log-scaled measurements by real-city frequency rank (left: most frequent) in the pre-trained model: coordinate reconstruction error and distance mean absolute error on up to four sampled training-corpus rows per city. Rank-measurement details appear in App. E.3. (d,e) Atlantis reconstruction error and cross-task NI after each fine-tuning task, comparing $\alpha \in \{0, 1, 1.25\}$.

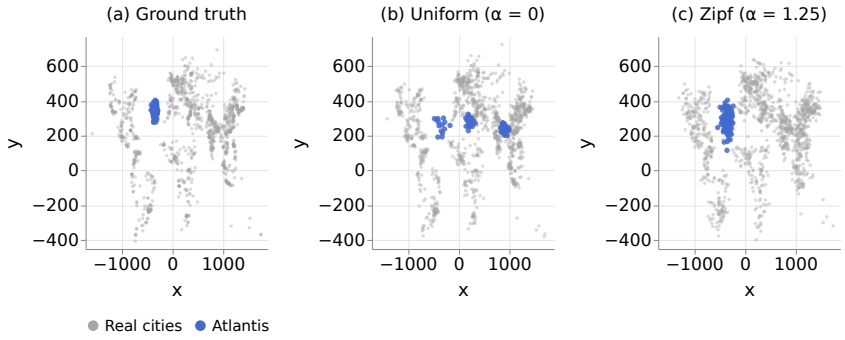


Figure 6: **Decoded city locations after distance fine-tuning.** Each model uses the seed with lowest Atlantis reconstruction error. Real cities are gray; Atlantis cities are blue.

for new-entity MAML. For distance fine-tuning, Zipf sampling yields 179, compared with 374 for the strongest alternative (input noise). Zipf sampling therefore outperforms the tested alternatives under these recipes and budgets.

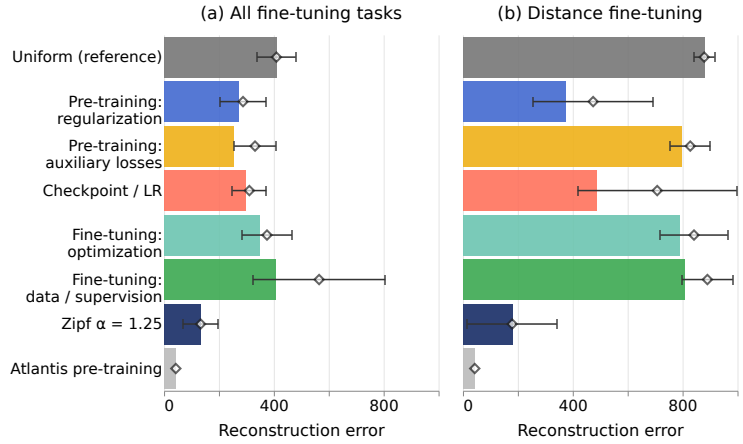


Figure 7: **Entity sampling versus alternative interventions.** Atlantis reconstruction error averaged across seven fine-tuning tasks (left) or after distance fine-tuning (right). Bars select each family's best method; diamonds and whiskers pool methods and seeds (mean ± SD), rather than giving uncertainty for the selected method. Recipes, reference populations, and aggregation details appear in App. F.

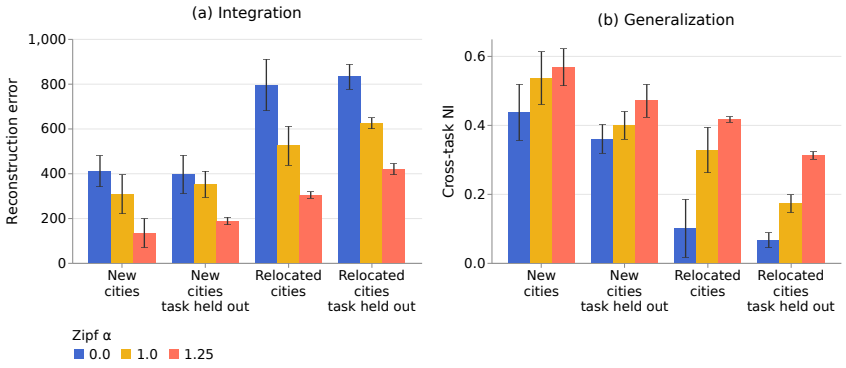


Figure 8: **Adaptation to new and relocated cities, including held-out tasks.** (a) Reconstruction error and (b) cross-task NI for $\alpha \in \{0, 1, 1.25\}$. Task averages precede three-seed means ± SD. Reconstruction covers 87 eligible new cities or all 100 relocated cities; NI averages the six other tasks. Protocols and per-task results appear in App. G.

4.2 Effectiveness in other training settings

Zipfian pre-training also makes it easier to update existing entities (Fig. 8). After pre-training, we relocate 100 real cities to the Atlantis locations, retaining their IDs, and fine-tune on the updated world. Averaged across seven fine-tuning tasks and three seeds, $\alpha = 1.25$ lowers relocated-city reconstruction error from 795 to 303 (a 2.6× reduction relative to uniform pre-training) and raises cross-task NI from 0.100 to 0.417.

We also test adaptation through a previously unseen task by holding out the eventual fine-tuning task from pre-training, then adding 50,000 real-city examples of that task during fine-tuning. With $\alpha = 1.25$, reconstruction error falls from 395 to 187 for new cities and from 831 to 419 for relocated cities, approximately halving error in both settings. Mean cross-task NI also increases, from 0.360 to 0.471 and from 0.066 to 0.312, respectively. Thus, the benefit extends to revising existing knowledge and adapting through tasks absent from pre-training. Protocols and per-task results appear in App. G.

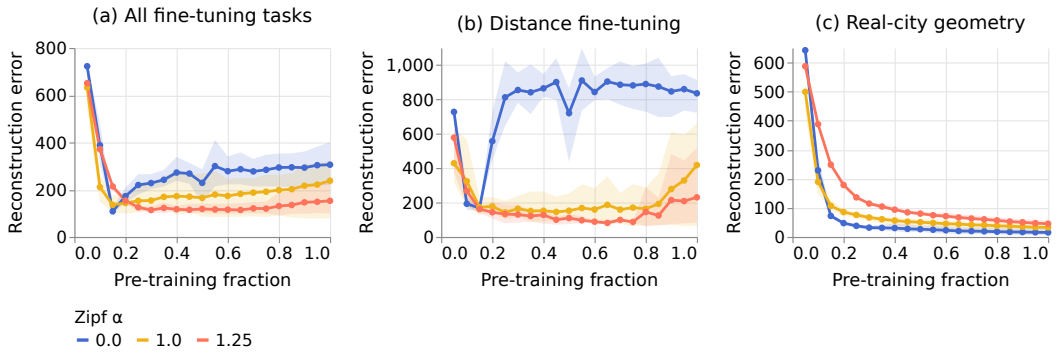


Figure 9: **Gradual world learning accompanies better adaptability.** Atlantis reconstruction after fine-tuning from successive checkpoints, averaged across tasks (a) or for distance alone (b); real-city reconstruction during pre-training (c). Lines show three-seed means and ribbons the seed range. Panel (c) includes all 4,545 probe-eligible cities and measures aggregate learning, not individual acquisition times.

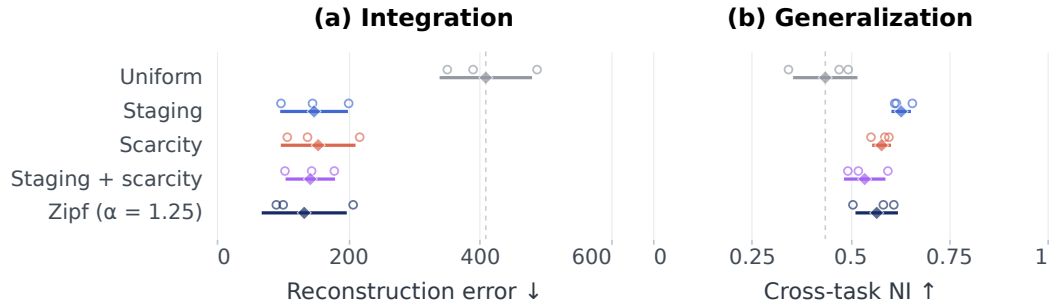


Figure 10: **Staging and scarcity improve integration.** (a) Atlantis reconstruction and (b) cross-task NI, averaged across seven fine-tuning tasks within seed. Diamonds, bars, and circles show three-seed means, ± SD, and individual seeds; dashed lines mark uniform means. Zipf uses a different fine-tuning replay distribution. Recipes and per-task results appear in App. H.

5 Keeping Entities Effectively New

Gradual world learning accompanies adaptability Zipfian pre-training ($\alpha = 1.25$) preserves better later integration than uniform sampling while learning the existing world more gradually (Fig. 9). Real-city reconstruction error falls from 179 at 20% of training to 46 at completion, compared with 48 to 16 under uniform sampling. Final errors also concentrate among less frequent cities (Fig. 5(b,c)). We hypothesize that continued learning of incompletely integrated entities provides practice for later integration, and this continued practice promotes maintaining plasticity in the learned world model.

Introducing new entities during training All cities first appear by 0.65% of training at $\alpha = 1.25$, versus 0.032% under uniform sampling. First exposure is therefore delayed but still early. We test whether extending entity introductions throughout training (*staging*) can also preserve later integration.

We randomly order the real cities, make 60% available initially, and release four additional 10% cohorts at 20%, 40%, 60%, and 80% of pre-training. A training row becomes available only when all its cities have been released. Staging lowers mean Atlantis reconstruction error across seven fine-tuning tasks from 409 to 148 and raises cross-task NI from 0.437 to 0.629 (Fig. 10). After distance fine-tuning, staging yields error 197, whereas random-row release with the same available-pool-size schedule yields 852, close to the uniform baseline's 879 (Fig. 11(a)).

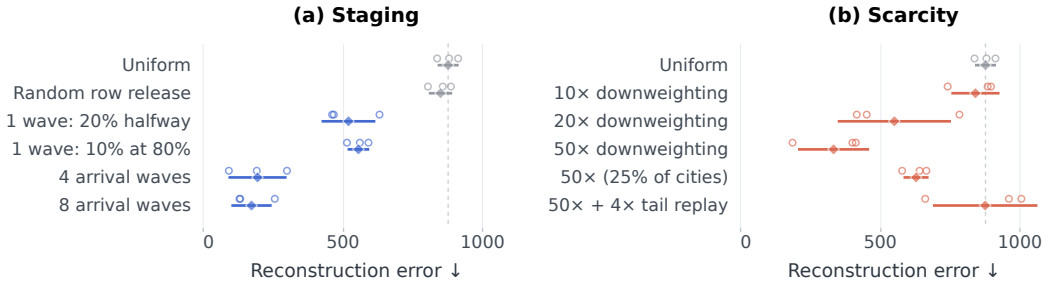


Figure 11: **Entity staging helps; tail replay removes the scarcity benefit.** Distance-fine-tuned Atlantis reconstruction under staging (a) and scarcity (b) controls. Row release matches pool sizes, not entity histories; tail replay keeps training-row budget and fine-tuning replay fixed. Diamonds, bars, and circles show means, ± SD, and three seeds. Dashed lines mark uniform means.

Scarcity of entities during training We can also measure the effect of having rare entities in our training distribution (i.e. *scarcity*), which is another characteristic of Zipfian distributions and could result in seeing under-integrated entities throughout training. Here, we assign cities to either a rare or frequent group (with probability $1/2$). Rare cities receive relative sampling weight 0.02, versus 1 for frequent cities. This separates persistent rarity from explicit introductions. Our scarcity distribution lowers task-averaged Atlantis error to 154 and raises cross-task NI to 0.580 (Fig. 10).

On the other hand, adding three copies of rare-entity rows to pre-training reverses this benefit after distance fine-tuning: error rises from 335 to 878 (Fig. 11(b)). Thus, making rare entities more familiar removes the benefit in this control. We describe other variations of staging schedules and scarcity in App. H. In all, these controls support patterns of exposure to training entities that keep entities incompletely learned and promote practice integrating these entities throughout pre-training.

6 Discussion

Our evidence is limited to small Transformers and settings with synthetic world geometry. It is still an open question whether long-tailed data preserves plasticity at scale in natural language settings. Although models appear to use information provided in-context in flexible ways, it is not always convenient to restate such knowledge in every prompt or to even know the relevant knowledge. We study whether weight updates make new facts reusable, such that they do not need to be explicitly stated. Our findings suggest treating future adaptability as a criterion for designing pre-training distributions. The staging and scarcity results motivate testing whether continued practice integrating entities can preserve this adaptability in natural-language models, while maintaining the coverage needed to learn useful world representations.

7 Related Work

Knowledge integration and world representations New-entity learning and knowledge editing often fail to propagate beyond trained facts (Onoe et al., 2023; Cohen et al., 2024). Spatial structure is linearly decodable in language models (Gurnee & Tegmark, 2023), while interventions in Othello models demonstrate a computational role for internal world states (Li et al., 2023). Building on the city world of Park (2026), we measure geometric integration, test computational reuse, and study how pre-training affects both.

Plasticity, meta-learning, and data distributions Warm-starting and continual learning can impair later learning (Ash & Adams, 2020; Dohare et al., 2024). MAML explicitly optimizes post-adaptation performance (Finn et al., 2017); its updates can favor within-task gradient agreement (Nichol et al., 2018). Our MAML recipe targets within-task adaptation; our evaluation also tests cross-task transfer. Chan et al. (2022) show that rare classes and bursty exposure encourage in-context learning. We study integration through subsequent weight updates. Further related work appears in App. A.

AI USE STATEMENT

In this work, we used generative AI tools for "propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods".

We have not used generative AI tools for "generate synthetic data sets, support qualitative and thematic data analysis, interpret results" and "help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, assist with translation, clean and reformat dataset" are not applicable to this work.

Additionally, we used generative AI tools for "create or modify scientific figures or images, suggest experimental parameters, create or edit software code, creation of artifacts, draft parts of a research paper, summarize or analyse existing literature, discover research topics or identify gaps, brainstorming, sourcing/searching for information, edit a research paper to improve readability, identify relevant literature". We have reviewed all AI-assisted work. For example, although we use AI tools for conducting many tasks in this project, our scientific process is human-driven and the narrative in this paper is carefully curated. We have only included experiments and results in our paper that we have thought about considerably. We have checked the code and experimental details to ensure these are aligned with our goals. We use additional software to ensure the artifacts and figures in this paper are generated directly from the code we have audited. The primary method and investigations in our work were not proposed by AI tools and we are not aware of these appearing in any other work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

App. B specifies data generation, optimization, probes, evaluation, and aggregation. Intervention recipes and staging/scarcity controls appear in Apps. F and H. We plan to release the complete codebase and data.

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.

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.

Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/1910.08475.

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.

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.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2205.05055.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 2024. doi: 10.1162/tacl_a_00644. URL https://aclanthology.org/2024.tacl-1.16/.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models, 2021. URL https://arxiv.org/abs/2104.08164.

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: 768–774, 2024. doi: 10.1038/s41586-024-07711-7. URL https://www.nature.com/articles/s41586-024-07711-7.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135, 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

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.

Wes Gurnee and Max Tegmark. Language models represent space and time. *arXiv preprint arXiv:2310.02207*, 2023. URL https://arxiv.org/abs/2310.02207.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models, 2023. URL https://arxiv.org/abs/2301.04213.

Peter Hase, Thomas Hofweber, Xiang Zhou, Elias Stengel-Eskin, and Mohit Bansal. Fundamental problems with model editing: How should rational belief revision work in LLMs?, 2024. URL https://arxiv.org/abs/2406.19354.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2404.06395.

Jing Huang, Daniel Wurgaft, Rachit Bansal, Laura Ruis, Naomi Saphra, David Alvarez-Melis, Andrew Kyle Lampinen, Christopher Potts, and Ekdeep Singh Lubana. Why larger models learn more: Effects of capacity, interference, and rare-task retention, 2026. URL https://arxiv.org/abs/2605.29548.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20617–20642, 2024. URL https://proceedings.mlr.press/v235/huh24a.html.

M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics—Simulation and Computation*, 18(3):1059–1076, 1989. doi: 10.1080/03610918908812806. URL https://doi.org/10.1080/03610918908812806.

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.

Jaeyeon Kim, Sehyun Kwon, Joo Young Choi, Jongho Park, Jaewoong Cho, Jason D. Lee, and Ernest K. Ryu. Task diversity shortens the ICL plateau, 2025. URL https://arxiv.org/abs/2410.05448.

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.

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.

Andrew K. Lampinen, Arslan Chaudhry, Stephanie C. Y. Chan, Cody Wild, Diane Wan, Alex Ku, Jörg Bornschein, Razvan Pascanu, Murray Shanahan, and James L. McClelland. On the generalization of language models from in-context learning and finetuning: a controlled study, 2025. URL https://arxiv.org/abs/2505.00661.

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 *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.13382.

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

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.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1711.05101.

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.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets, 2024. URL https://arxiv.org/abs/2310.06824.

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.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022. doi: 10.52202/068431-1262. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer, 2023. URL https://arxiv.org/abs/2210.07229.

Eric J Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. *arXiv preprint arXiv:2303.13506*, 2023.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale, 2022a. URL https://arxiv.org/abs/2110.11309.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale, 2022b. URL https://arxiv.org/abs/2206.06520.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, 2023. URL https://arxiv.org/abs/2309.00941.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0607f4c705595b911a4f3e7a127b44e0-Abstract.html.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 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.

Maya Okawa, Ekdeep Singh Lubana, Robert P. Dick, and Hidenori Tanaka. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task, 2024.

Charles O'Neill. Can a language model learn facts continually in its weights?, 2026. URL https://arxiv.org/abs/2607.11020.

Yasumasa Onoe, Michael Zhang, Shankar Padmanabhan, Greg Durrett, and Eunsol Choi. Can LMs learn new entities from descriptions? Challenges in propagating injected knowledge. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5469–5485, 2023.

OpenDataSoft / GeoNames. Geonames – all cities with a population > 1000. https://public.opendatasoft.com/explore/dataset/geonames-all-cities-with-a-population-1000, 2025. Accessed: 2025.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. Fine-tuning or retrieval? comparing knowledge injection in llms, 2024. URL https://arxiv.org/abs/2312.05934.

Core Francisco Park. Convergent World Representations and Divergent Tasks. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/pdf/2602.00533.

Core Francisco Park, Zechen Zhang, and Hidenori Tanaka. *New News*: System-2 fine-tuning for robust integration of new knowledge, 2025. URL https://arxiv.org/abs/2505.01812.

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.

Gautam Reddy. The mechanistic basis of data dependence and abrupt learning in an in-context classification task, 2023. URL https://arxiv.org/abs/2312.03002.

Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2): 123–146, 1995. doi: 10.1080/09540099550039318. URL https://doi.org/10.1080/09540099550039318.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(56):1929–1958, 2014. URL https://jmlr.org/papers/v15/srivastava14a.html.

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

Keyon Vafa, Peter G. Chang, Ashesh Rambachan, and Sendhil Mullainathan. What has a foundation model found? using inductive bias to probe for world models, 2025. URL https://arxiv.org/abs/2507.06952.

Jacob VanderPlas, Brian E. Granger, Jeffrey Heer, Dominik Moritz, Kanit Wongsuphasawat, Arvind Satyanarayan, Eitan Lees, Ilia Timofeev, Ben Welsh, and Scott Sievert. Altair: Interactive statistical visualizations for Python. *Journal of Open Source Software*, 3(32):1057, 2018. doi: 10.21105/joss.01057. URL https://joss.theoj.org/papers/10.21105/joss.01057.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, pp. 1096–1103, 2008. doi: 10.1145/1390156.1390294. URL https://www.cs.toronto.edu/~larocheh/publications/icml-2008-denoising-autoencoders.pdf.

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.

Zixuan Wang, Xingyu Dang, Jason D. Lee, and Kaifeng Lyu. The power of power law: Asymmetry enables compositional reasoning, 2026. URL https://arxiv.org/abs/2604.22951.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

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.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model?, 2025. URL https://arxiv.org/abs/2504.13837.

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.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.16042.

Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15686–15702, 2023. doi: 10.18653/v1/2023.emnlp-main.971. URL https://aclanthology.org/2023.emnlp-main.971/.

Nicolas Zucchet, Jörg Bornschein, Stephanie Chan, Andrew Lampinen, Razvan Pascanu, and Soham De. How do language models learn facts? dynamics, curricula and hallucinations, 2025. URL https://arxiv.org/abs/2503.21676.

Adam Zweiger, Jyothish Pari, Han Guo, Ekin Akyürek, Yoon Kim, and Pulkit Agrawal. Self-adapting language models, 2025. URL https://arxiv.org/abs/2506.10943.

A Additional related work

We group additional related work by its connection to learning and updating entity representations.

Catastrophic forgetting, plasticity, and continual learning

- McCloskey & Cohen (1989) show that connectionist networks trained on tasks sequentially suffer catastrophic interference with earlier learning.
- Kirkpatrick et al. (2017) mitigate forgetting with elastic weight consolidation, which slows learning on weights important to earlier tasks.
- Li et al. (2018) regularize fine-tuned weights toward the pre-trained solution (L2-SP) to preserve transferred knowledge.
- Kotha et al. (2024) explain forgetting in language models partly as fine-tuning skewing the model's implicit inference of which task it is solving.
- Luo et al. (2025) measure forgetting of general knowledge and reasoning during continual instruction tuning of LLMs.
- Achille et al. (2019) show that deep networks have critical learning periods, where deficits early in training cannot be reversed by later training.

Acquiring new knowledge through fine-tuning

- Gekhman et al. (2024) find that LLMs learn facts unknown to the base model slowly during fine-tuning, and that learning them increases hallucination.
- Berglund et al. (2024) show that models fine-tuned on "A is B" fail to generalize to "B is A" (the reversal curse).
- Treutlein et al. (2024) show that LLMs can infer and verbalize latent structure from facts scattered across training documents.
- Wang et al. (2025) explain several instances of out-of-context reasoning as LoRA fine-tuning adding a steering vector toward a concept the model already represents.
- Allen-Zhu & Li (2023) show that knowledge stored during pre-training is extractable by fine-tuning only if the pre-training data present it with sufficient diversity.
- Allen-Zhu & Li (2024) show that language models struggle to manipulate stored knowledge, e.g. for comparison or inverse search.
- Park et al. (2025) find a gap between fine-tuning and in-context learning for integrating news, and narrow it with self-generated training data.
- Lampinen et al. (2025) find that in-context learning generalizes more flexibly than fine-tuning in controlled settings, and that augmenting fine-tuning data with in-context inferences helps.
- O'Neill (2026) tracks facts written sequentially into LLM weights, finding that bare-statement training yields recitation rather than usable knowledge, and that later writes leave earlier facts stored but unreachable.
- Zweiger et al. (2025) train models with reinforcement learning to generate their own fine-tuning data, so that subsequent weight updates improve performance.

How pre-training shapes fine-tuning

- Bommasani et al. (2021) describe the paradigm of adapting large pre-trained foundation models to downstream tasks.
- Jain et al. (2023) show in procedurally defined tasks that fine-tuning often learns a thin wrapper over pre-trained capabilities.
- Prakash et al. (2024) show that fine-tuning enhances the entity-tracking circuit already present in the pre-trained model.
- Ward et al. (2025) show that reasoning fine-tuning repurposes a direction already present in the base model's activations.
- Kumar et al. (2022) show that full fine-tuning can distort pre-trained features and underperform out of distribution.

- Neyshabur et al. (2020) study what transfer learning transfers, separating feature reuse from low-level statistics of the data.
- Liu et al. (2022) show that models with the same pre-training loss can differ in downstream performance due to implicit bias.
- Aghajanyan et al. (2020) find that pre-trained language models have low intrinsic dimension for fine-tuning.
- Zhang et al. (2025) find in controlled settings that reinforcement learning post-training yields capability gains only when pre-training leaves headroom.
- Yue et al. (2025) question whether reinforcement learning with verifiable rewards expands reasoning capacity beyond the base model.

World representations Huh et al. (2024) argue that representations learned across models and modalities may converge toward shared world structure. Our continuous-geometry interventions complement this representational perspective by testing effects on predictions.

- Nanda et al. (2023) show that Othello-playing models represent the board linearly, in terms of the current player's and opponent's pieces.
- Marks & Tegmark (2024) find linear, causally relevant representations of truth in LLMs.
- Vafa et al. (2024) show that generative models can pass standard world-model diagnostics while their implied world models are incoherent.
- Vafa et al. (2025) probe world models through inductive bias, finding that models trained on orbital trajectories do not apply Newtonian mechanics when adapting to new tasks.
- Nishi et al. (2024) show in a synthetic setting that knowledge editing shatters structured entity representations.
- Kumar et al. (2025) argue that networks with the same behavior can have fractured, entangled internal representations.

Training distributions, multitask learning, and synthetic settings

- Caruana (1997) show that multitask learning improves generalization through shared representations.
- Aghajanyan et al. (2021) show that massive multitask pre-finetuning improves subsequent fine-tuning.
- Michaud et al. (2023) model neural scaling as learning discrete quanta in order of their Zipfian use frequency.
- Wang et al. (2026) show that power-law rather than uniform training distributions improve compositional reasoning, as frequent skill compositions serve as stepping stones for learning rare ones.
- Zucchet et al. (2025) study factual recall in a synthetic setting, finding that imbalanced fact distributions shorten learning plateaus and that fine-tuning on new knowledge quickly corrupts existing parametric memories.
- Huang et al. (2026) show that smaller models devote capacity to frequent tasks and fail to learn rare ones, and that larger models retain rare-task features through reduced gradient interference.
- Reddy (2023) show how data distributional properties drive abrupt emergence of in-context learning.
- Kim et al. (2025) show that training on diverse tasks shortens loss plateaus in in-context learning.
- Okawa et al. (2024) study the emergence of compositional abilities in a synthetic setting.

B EXPERIMENTAL DETAILS

We use the city-world framework of Park (2026). Unless noted otherwise, experiments use the following protocol; alternative recipes are specified in App. F.

B.1 WORLD AND GEOMETRIC TASKS

The world contains 5,075 GeoNames cities with populations above 100,000 (OpenDataSoft / GeoNames, 2025). Coordinates are $x = \mathrm{round}(10\,\mathrm{longitude})$ and $y = \mathrm{round}(10\,\mathrm{latitude})$; tasks use Euclidean, not geodesic, geometry. Atlantis adds 100 cities with independent longitude and latitude drawn from Gaussians centered at $(-35°, 35°)$, each with standard deviation $3°$, then converted identically. Unique random IDs from 0–9,999 have four digits (e.g. `c_0304`). Coordinates and IDs are fixed across seeds.

Examples have the character-tokenized form `task(arguments)=answer`, without spaces or coordinate inputs (Table 1). Numeric answers are integer-rounded, ties to even, without leading zeros. Commas separate arguments; `inside` instead separates the query city from its polygon with a semicolon, and `cross` separates its two segments with a semicolon.

Task	Cities	Answer for coordinates $z_i = (x_i, y_i)$	Output
Distance (`dist`)	2	$\|z_2 - z_1\|_2$.	Coordinate units
Triangle area (`triarea`)	3	$\frac{1}{2}\|\det(z_2 - z_1, z_3 - z_1)\|$.	Squared coordinate units
Angle (`angle`)	3	The angle at the middle city, between $z_1 - z_2$ and $z_3 - z_2$.	Degrees, 0–180
Compass (`compass`)	2	Direction from the first city to the second, using eight $45°$ sectors centered on the cardinal and intercardinal directions.	N, NE, E, SE, S, SW, W, NW
Inside (`inside`)	4–9	Whether the first city lies in the convex hull of the remaining cities; the boundary counts as inside.	TRUE/FALSE
Perimeter (`perimeter`)	3–8	$\sum_{i=1}^{n}\|z_{i+1} - z_i\|_2$ for n input cities, with $z_{n+1} = z_1$. Vertices retain their input order; this is not the convex-hull perimeter.	Coordinate units
Crossing (`cross`)	4	Whether segments (z_1, z_2) and (z_3, z_4) intersect, including touching or collinear overlap.	TRUE/FALSE

Table 1: Geometric tasks used for pre-training, fine-tuning, and evaluation. Numeric outputs are integer-rounded. City IDs are distinct within each input; distinct IDs may share rounded coordinates.

Uniform sampling rejects repeated city IDs, coincident compass endpoints, and zero-length angle arms. Inside and crossing have equal TRUE/FALSE quotas. Variable-arity proposals sample 3–8 cities, but processing by arity before filling quotas can change the retained arity distribution. Atlantis examples sample a tuple from the full world, replace a random argument with a uniformly sampled Atlantis ID, and reject duplicate IDs. They contain at least one Atlantis city; other arguments may also be Atlantis cities.

B.2 DATA AND OPTIMIZATION

Pre-training uses one million rows per task (seven million total), allowing repeated queries. Each task has 512 validation examples and two shared evaluation sets of 2,000 queries: real-only and Atlantis-containing. Pre-training excludes exact validation/evaluation prompts; Atlantis fine-tuning and real-city elicitation exclude their corresponding evaluation prompts. Equivalent queries, such as reversed distance pairs, may remain. The two-Atlantis evaluation below additionally excludes symmetric equivalents.

Randomly initialized Qwen2-style Transformers (Yang et al., 2024) use six blocks, width 128, four attention and key/value heads, MLP width 512, and separate embeddings and output weights (approximately 1.60M parameters), implemented in Hugging Face Transformers (Wolf et al., 2020). The vocabulary contains 94 printable non-space ASCII characters plus BOS, EOS, and PAD. Rows include BOS/EOS and pad to at most 256 tokens. Baseline attention dropout is zero; embedding and linear weights use initialization standard deviation 0.1.

Both stages minimize next-token cross-entropy over prompts, answers, and EOS, masking padding only. AdamW (Loshchilov & Hutter, 2019) uses $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, one parameter group, float32, and no gradient clipping. Each stage resets the optimizer, warms up for 50 updates,

then decays the learning rate linearly (Table 2). Rows are shuffled without replacement and reshuffled when fewer than a full batch remain.

Setting	Pre-training	Fine-tuning
Optimizer	AdamW	AdamW, reset state
Peak learning rate	3×10^{-4}	10^{-5}
Weight decay	0.01	0.01
Batch size	128 rows	128 rows
Warmup	50 updates	50 updates
Schedule after warmup	Linear decay	Linear decay
Training budget	42 million row presentations	30 nominal epochs
Optimizer updates	328,125	28,545 for the default mixture

Table 2: Default optimization settings. Fine-tuning updates equal $\lfloor 30N/128 \rfloor$ for mixture size N; interventions that add rows therefore change the number of updates unless explicitly compute-matched.

Single-task fine-tuning uses 100,000 Atlantis rows, 20,000 real-city replay rows, and 256 new real-city elicitation rows per task (121,792 total). Atlantis facts occur only in the selected task; replay and elicitation cover real cities. All parameters are updated from the final pre-training checkpoint unless specified otherwise. Trajectories repeat this procedure at 5%, 10%, ..., 100% of pre-training, resetting the fine-tuning optimizer and schedule each time.

Zipf sampling A seed-specific random permutation assigns shared city ranks $r_i \in \{1, \ldots, 5075\}$ across tasks. Arguments are sampled with weights $p_i = r_i^{-\alpha} / \sum_j r_j^{-\alpha}$, conditional on distinct IDs. Geometric-validity and class-balance constraints can alter observed frequencies. We sweep $\alpha = 0, 0.05, \ldots, 2$ at fixed row budget; above 1.5, an exact conditional sampler replaces inefficient rejection sampling. Validation, evaluation, Atlantis fine-tuning, and the main Zipf comparisons' real-city replay remain uniform.

B.3 Representation and behavioral measurements

Coordinate reconstruction Layer numbers count completed transformer blocks. A BOS-prefixed distance prefix, `dist(c_1234,`, supplies layer-5 states at the final ID digit and comma; concatenation yields $h_i \in \mathbb{R}^{256}$. Each checkpoint gets a separate ordinary-least-squares probe with intercept, $\hat{z}_i = W h_i + b$, fitted on 3,250 real cities and tested on 1,250 others with a fixed split. IDs beginning with zero are excluded from probing, leaving 87 Atlantis cities, but remain in training and behavioral evaluation. Atlantis coordinates never enter the probe fit.

Reconstruction error is $|S|^{-1} \sum_{i \in S} \|\hat{z}_i - z_i\|_2$, in scaled coordinate units. Post-fine-tuning probes are refitted to that checkpoint's real-city states. For relocation, only unchanged cities fit the probe (up to 3,250, after reserving 1,250 for testing); all 100 moved cities, including leading-zero IDs, are scored against updated coordinates. The pre-training trajectories and frequency-rank analysis use a different probe population, specified in App. E.3.

Generation and task scores Greedy decoding starts after the equals sign and stops at EOS or 24 generated tokens. Distance, triangle area, angle, and perimeter use mean absolute error; compass, inside, and crossing use exact accuracy. Invalid categorical outputs are incorrect. Unparseable numeric answers incur the error of predicting zero and remain in the mean; parse rates are tracked separately.

Normalized improvement For each task t, let E_t denote an evaluated model's mean absolute error and A_t its accuracy. Let superscripts B and O denote the fixed baseline and oracle references. We

use the log-error and linear-accuracy transformations

$$\mathrm{NI}_t = \begin{cases} \dfrac{\log(E_t^B/E_t)}{\log(E_t^B/E_t^O)}, & \text{numerical tasks,} \\ \dfrac{A_t - A_t^B}{A_t^O - A_t^B}, & \text{categorical tasks.} \end{cases} \tag{1}$$

References are seed-0 uniform models pre-trained without Atlantis (B) or on all 5,175 cities (O), with equal corpus sizes and budgets, evaluated without fine-tuning. They are shared across methods and seeds. Unlike Park (2026)'s real-city upper reference, NI $= 1$ denotes oracle performance on Atlantis. NI is unclipped, and zero need not equal a method's own pre-fine-tuning score. Cross-task NI averages the six non-fine-tuning tasks. Relocation and held-out-task tests retain these references; the two-Atlantis test recomputes them on its own queries.

B.4 Additional evaluation protocols

Activation interventions Geometric interventions patch the final ID digit and comma after block 4 (App. C). Computational reuse instead patches all seven positions of the city ID and delimiter from the fine-tuned model into its original pre-trained model, evaluating the other six tasks. Donors use the fine-tuning-task prefix; delimiters are semicolons for inside and commas otherwise.

Computational reuse uses 1,024 real-city prompts per recipient task and seed. We replace the first city's activation span with an eligible Atlantis donor's and score against answers recomputed from donor coordinates; the input ID remains unchanged. Donor prefixes end before later arguments. Prompts are shared across fine-tuning tasks within each seed. Controls use real-city donors, unpatched prompts, literal ID replacement, and recipient-task donor prefixes. The main figure uses Atlantis donors, fine-tuning-task prefixes, and the fixed NI references.

Two-Atlantis composition The composition test uses 1,024 queries per task with exactly two Atlantis cities and real-city remaining arguments. We exclude geometrically equivalent tuples, including symmetric reorderings, found in any relevant fine-tuning corpus across seeds or in oracle pre-training. NI references use these same queries. This tests unseen combinations, although fine-tuning itself can contain two-Atlantis examples.

Relocated cities and held-out tasks Each seed relocates 100 real cities to the fixed Atlantis coordinates, retaining IDs and using the same cohort across Zipf exponents. Fine-tuning uses 100,000 updated facts; replay and elicitation exclude moved cities. Held-out-task pre-training omits the eventual fine-tuning task but retains the 42-million-presentation budget. Adaptation adds 50,000 real-city rows of the new task to 100,000 target-city rows, 20,000 replay rows, and 256 elicitation rows per pre-training task. In relocation, real-city rows use unchanged cities. Evaluation uses 2,000 target-containing queries for each of the six pre-training tasks.

B.5 Replication and aggregation

Training comparisons use seeds 0–2, varying initialization, corpus generation, training order, fine-tuning data, and Zipf ranks while fixing the world, probe split, and ordinary evaluations. Task averages are computed within seed before averaging seeds. Error bars use sample SD; trajectory ribbons use seed ranges where specified. Geometry interventions use seeds 1 and 2; the Atlantis oracle uses one model. Family-level pooling is defined below. Experiments and analyses are cached Misen tasks, with Polars aggregation and Altair rendering (VanderPlas et al., 2018).

C Geometric interventions and computational reuse

Intervention protocol For Fig. 2, donor states come from distance prefixes ending before the anchor appears. Edits replace the final ID-digit and comma activations after block 4. We restrict these analyses to models that pass an ordinary city-replacement control at this site. Displacement uses 32 donor pairs with three distinct base cities each and $\lambda \in \{-0.5, -0.25, 0.25, 0.5\}$; interpolation uses $\lambda \in \{0.25, 0.5, 0.75\}$. Errors average donor groups within each model before averaging across models; replication details appear in App. B.5.

Geometric responses and edit controls Distance changes are measured relative to an unmodified h_C patch. In Fig. 2(a), $R^2 = 1 - \sum_j (y_j - x_j)^2 / \sum_j (y_j - \bar{y})^2$, where x_j and y_j are geographic and model-output changes. It is computed separately for each model and then averaged; no line is fitted. All validly parsed points contribute, including those outside the displayed axes. Panel (b) compares displacement with unchanged states and two equal-norm random edits, averaging the latter.

Alternative behavioral explanations For Fig. 2(c), endpoint-answer arithmetic predicts $y_C + \lambda(y_B - y_A)$ for displacement and $(1 - \lambda)y_A + \lambda y_B$ for interpolation. Diagnostic anchors separate geometric and arithmetic predictions by at least 10 true-coordinate units, without fitting either predictor to edited answers. For panel (d), a continuous point and the best of 5,075 stored cities fit the same eight generated distances, then predict distances to eight held-out anchors. The fitted point need not equal the prescribed target.

C.1 Transfer between individual tasks

Fig. 12 disaggregates the task averages in Fig. 3; the activation-patching protocol appears in App. B.4.

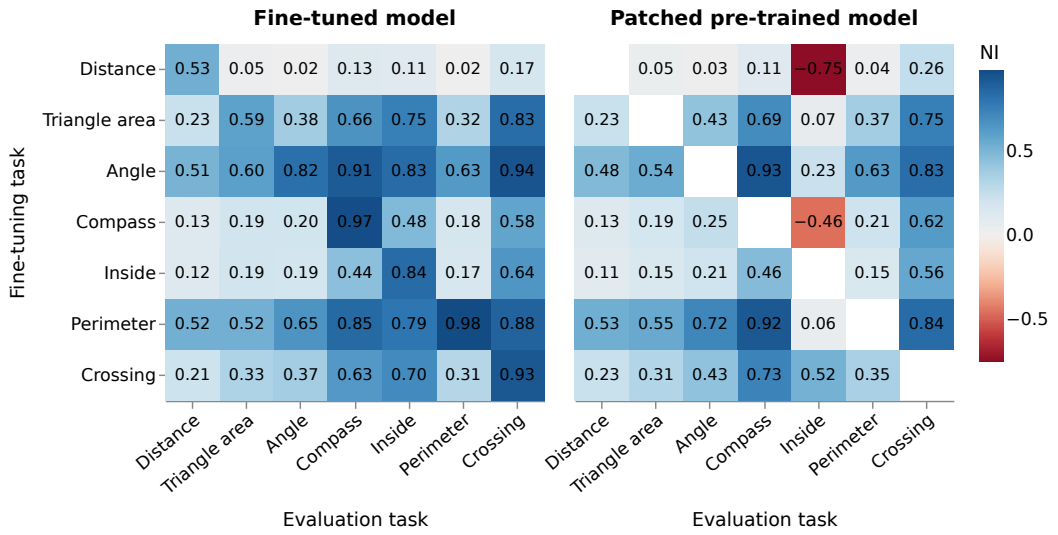


Figure 12: Fine-tuning-task by evaluation-task NI for Fig. 3. Left: the fine-tuned model, including its own-task diagonal. Right: Atlantis activations from that model patched into its original pre-trained model; own-task cells are unevaluated. Cells show mean NI using fixed references; negative NI is retained.

D ADAPTABILITY BY FINE-TUNING TASK

The following curves disaggregate Figs. 4 and 9. Each checkpoint undergoes seven separate single-task fine-tunes.

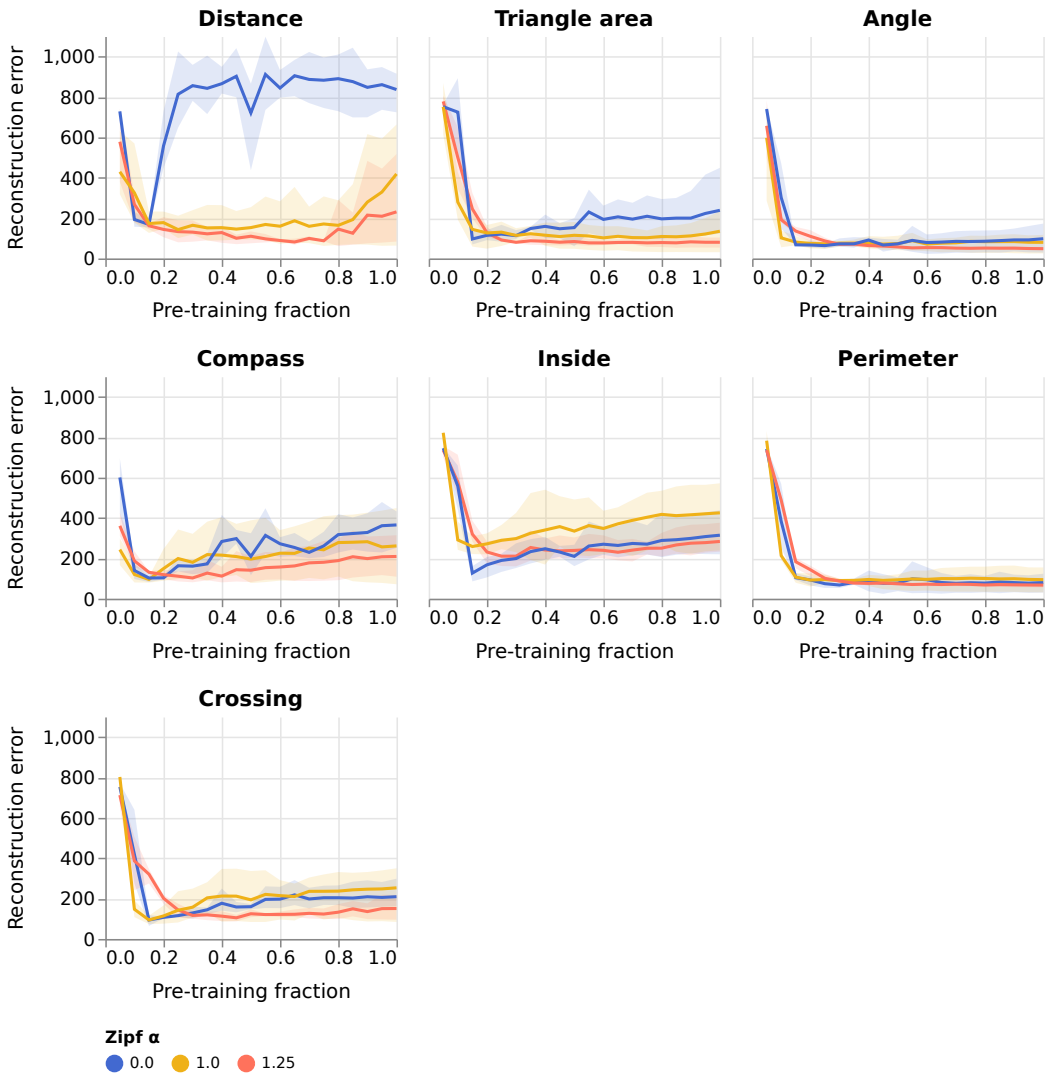


Figure 13: Atlantis reconstruction by fine-tuning task and pre-training checkpoint. Lines show three-seed means; ribbons show seed ranges. The blue ($\alpha = 0$) curves disaggregate the uniform-model average in Fig. 4; all three curves disaggregate Fig. 9.

E ZIPF SAMPLING BY TASK AND ENTITY POPULATION

We separate the task-averaged exponent sweep into its seven fine-tuning tasks and report reconstruction and behavior separately.

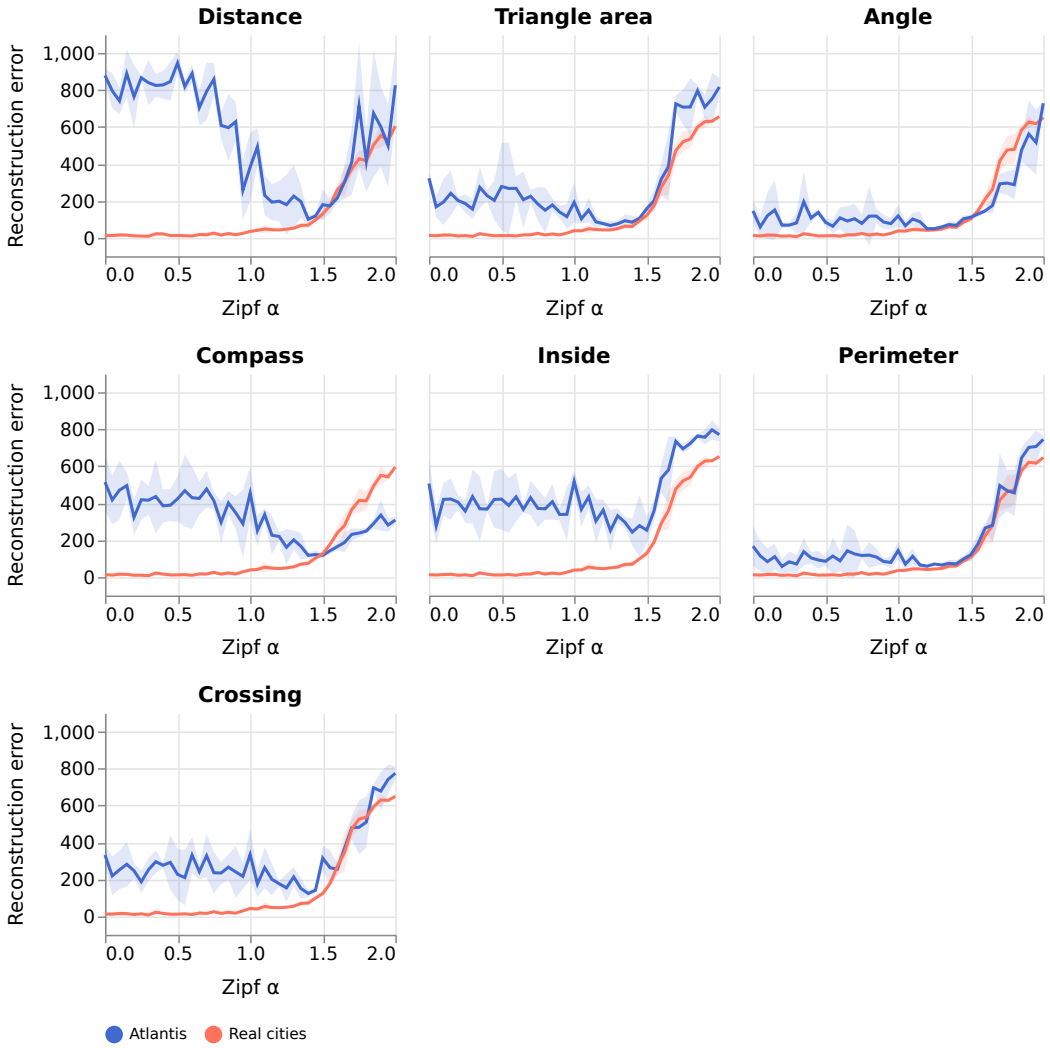


Figure 14: Post-fine-tuning reconstruction across Zipf exponents, separately for Atlantis and held-out real cities. Each panel is one fine-tuning task; lines and ribbons show three-seed means ± sample SD. This disaggregates the post-fine-tuning curves in Fig. 5.

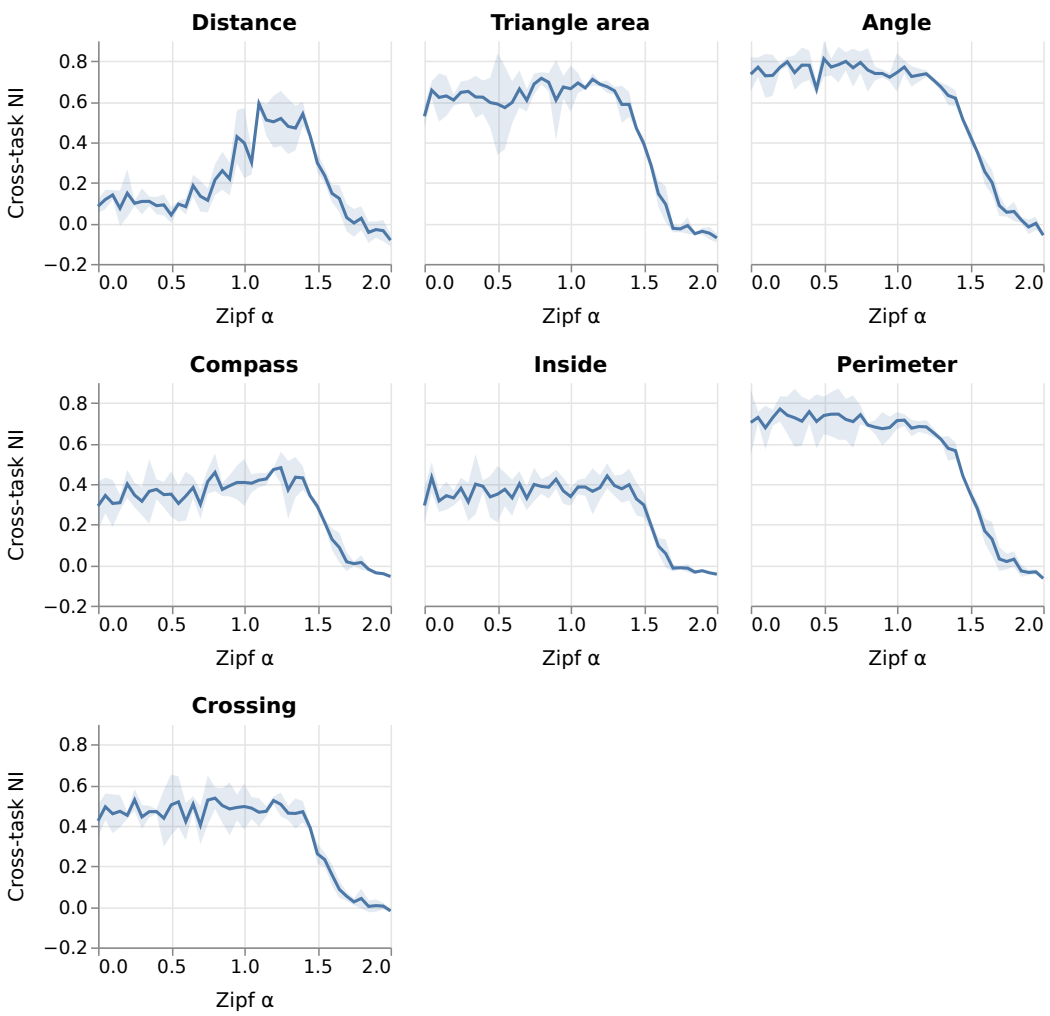


Figure 15: Cross-task NI across Zipf exponents for each fine-tuning task. Lines and ribbons show three-seed means ± sample SD. NI averages the six other tasks; axes retain negative values and full uncertainty intervals.

E.1 REAL-CITY RETENTION

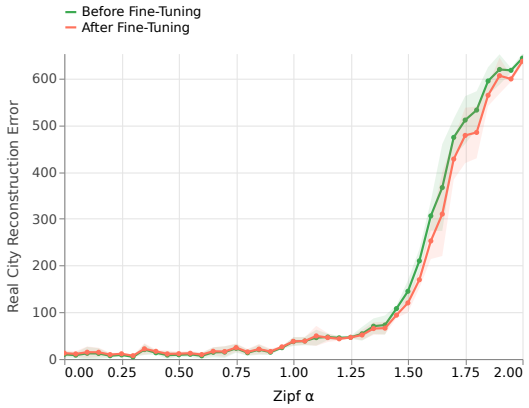


Figure 16: Held-out real-city reconstruction before and after fine-tuning across Zipf exponents. Task averages precede three-seed means ± sample SD; both curves share a y-axis.

E.2 REAL-CITY TASK PERFORMANCE AND FREQUENCY RANKS

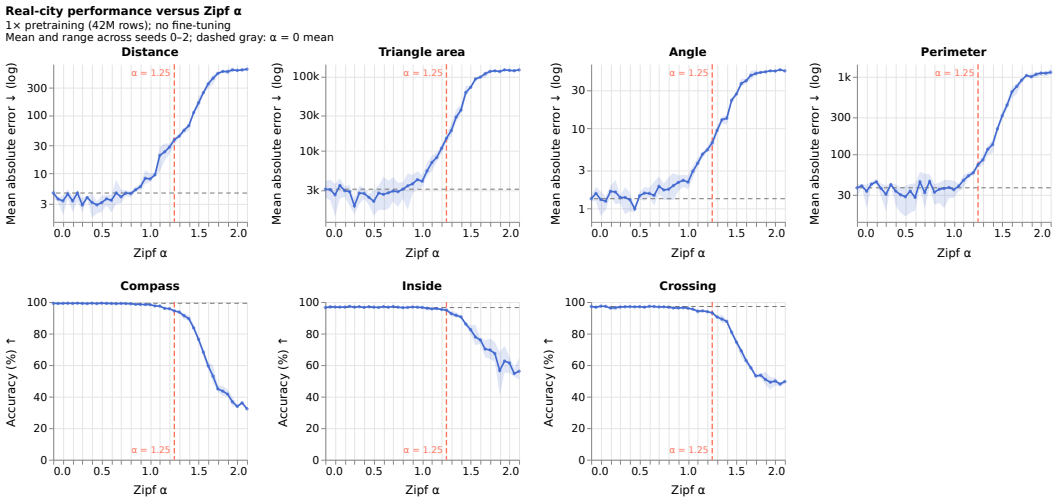


Figure 17: Real-city performance before fine-tuning at 42 million training-row presentations, on 2,000 shared queries per task. Lines show three-seed means; ribbons show seed ranges. Numeric-task MAE uses logarithmic axes; categorical tasks use accuracy. Dashed references mark uniform performance and $\alpha = 1.25$.

E.3 PERFORMANCE BY ENTITY FREQUENCY RANK

The frequency-rank panels of Fig. 5 use final checkpoints (328,125 updates; 42 million presentations), $\alpha \in \{0, 1, 1.25, 1.5, 1.6, 1.7, 1.8, 1.9, 2\}$, and seeds 0–2. We rank all 5,075 cities by actual query presentations containing them, across tasks and argument positions, breaking ties by ID and retaining zero-exposure cities. The 20 equal-width bins contain 253–254 cities before filtering. City scores are averaged within seed, then seeds weighted equally. Uniform-model ranks reflect realized count variation; ranks do not equate absolute exposure across exponents.

Frequency-rank reconstruction and pre-training trajectories use a separate layer-5 OLS probe per checkpoint, fitted on 3,636 real cities and evaluated on all 4,545 eligible cities, including 909 held out. Splits are fixed. Leading-zero IDs are excluded from probing but retain their original frequency ranks.

Distance error samples up to four training-row occurrences per focal city, uniformly without replacement. Each query scores only its focal city; repeated prompts retain multiplicity, although generation is deduplicated. Unparseable answers incur zero-prediction error; no extreme errors are removed. This estimates training-corpus error, not held-out generalization. Cities absent from distance training are unscored, and bins require at least 20 scored cities in every seed.

Held-out task performance Figs. 18 and 19 evaluate the same checkpoints on 2,000 uniformly sampled held-out real-city queries per task, shared across all models. Each query contributes its score to every city it involves. We average query scores within each city, city scores within each frequency-rank bin, and then the three seeds equally. Cities without held-out queries are unscored; bins require at least 20 scored cities in every seed. Ranks retain the ordering among all 5,075 cities, including cities unseen in pre-training. Unlike the training-corpus measurements in Fig. 5(c), these queries come from the held-out evaluation set.

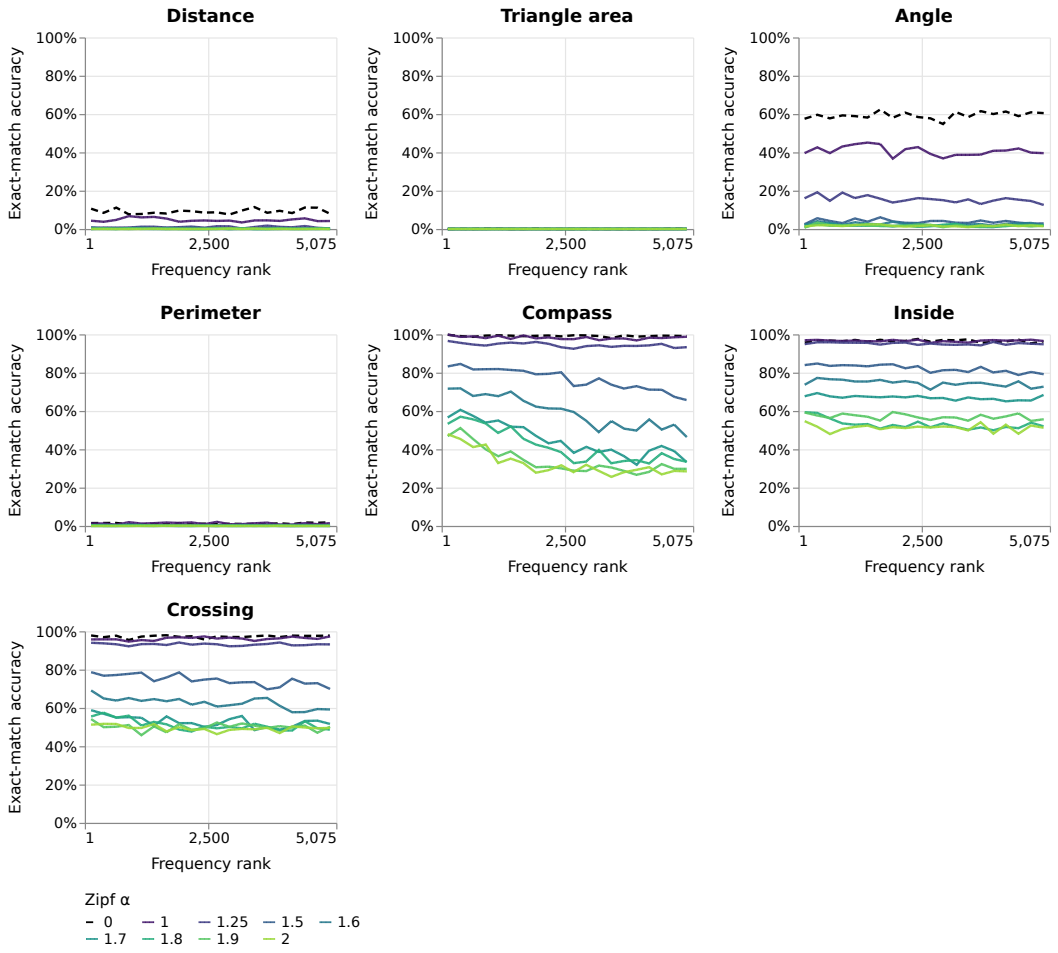


Figure 18: **Held-out task accuracy by entity frequency rank.** Exact-match accuracy for all seven tasks after pre-training, for $\alpha \in \{0, 1, 1.25, 1.5, 1.6, 1.7, 1.8, 1.9, 2\}$. Rank 1 is most frequent. Lines average city scores within 20 rank bins and then average three seeds equally; higher is better. The dashed black line is uniform pre-training ($\alpha = 0$).

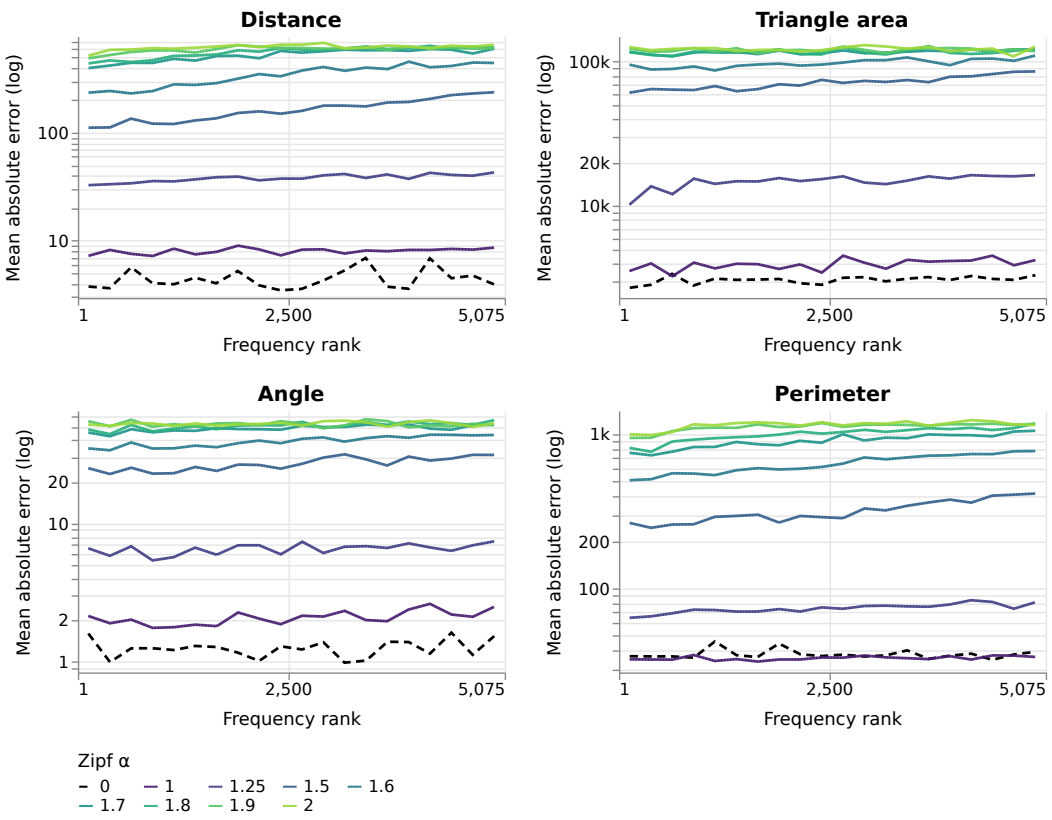


Figure 19: **Held-out numerical error by entity frequency rank.** Mean absolute error for the four numerical tasks on the same held-out queries, with the same models and averaging as Fig. 18. Each task uses its own logarithmic y-axis in its answer units; lower is better. Unparseable numerical answers incur the error of predicting zero, following the canonical evaluator. No observations or extreme errors are discarded.

F Alternative intervention recipes

We compare 15 alternatives in five families, plus distance-derived coordinate supervision for distance only. Recipes otherwise retain the uniform baseline, using seeds 0–2. Task averages precede seed averages. Each family bar selects its best method mean; diamonds and whiskers pool method–seed values after task averaging. The Atlantis-pre-training reference is one checkpoint without adaptation. Uniform and Zipf references use three seeds. All interventions score the same 87 probe-eligible Atlantis cities with the layer-5 coordinate probe. Family whiskers are sample standard deviations across pooled method–seed values, not uncertainty intervals for the selected best method.

Pre-training regularization Separate recipes use attention dropout (Srivastava et al., 2014) at rate 0.1, AdamW weight decay (Loshchilov & Hutter, 2019) at 0.1, or denoising-style character corruption (Vincent et al., 2008) at probability 0.05 with clean prediction targets.

Pre-training auxiliary objectives Our *task-conditioned mobility* objective differentiates through one virtual SGD step (learning rate 0.01) on one task involving selected entity-like units identified from recurring surface forms. Its outer loss favors their displacement relative to background units via the difference of log displacements. It runs every 1,000 pre-training steps with weight 0.3, rewarding movement rather than post-adaptation accuracy.

New-entity MAML adapts the objective of Finn et al. (2017), differentiating through three SGD steps at learning rate 0.001 and minimizing answer/EOS loss on disjoint query facts. Each episode samples 32 fresh entity/location bindings with four support and four query facts per entity, using

the eventual fine-tuning task's geometry and arity. Bindings are resampled and exclude Atlantis. Each task gets a separate initialization, 327 meta-updates, and outer-gradient norm capped at one, in addition to ordinary training. The objective optimizes within-task adaptation, not cross-task transfer or coordinate reconstruction.

Our *insertion-sharpness* penalty adds $3 \times 10^{-4} \max(0, z^\top H z)$ every 500 steps: H is the parameter Hessian of prediction loss on synthetic new-entity distance facts, and z has independent random signs. The quadratic form is a Hutchinson trace estimate (Hutchinson, 1989); we clip it before applying the penalty. Events use eight of 64 fixed synthetic entities, distinct from Atlantis, with 32 facts each. *Coordinate supervision* trains a linear head every four steps, with weight 0.3, on 60% of real cities; Atlantis coordinates remain hidden.

Checkpoint and learning-rate schedules Fine-tuning starts at 80% of pre-training: either a linear-decay checkpoint with learning rate 3×10^{-5}, or a warmup–stable–decay checkpoint (Hu et al., 2024) immediately before its final 20% decay, with learning rate 3×10^{-4}. Both checkpoint and fine-tuning rate change.

Fine-tuning optimization Separate recipes use learning rate 10^{-4} instead of 10^{-5}; shrink-and-perturb (Ash & Adams, 2020), $\theta \leftarrow 0.9\theta + 0.03\theta_{\text{init}}$, with fresh initialization θ_{init} and learning rate 10^{-4}; or update only MLPs in zero-indexed blocks 0–2. The latter differs from the blocks 1–3 diagnostic in App. I.

Fine-tuning data and supervision Replay (Robins, 1995) increases from 20,000 to 100,000 real-city rows, or uses 100,000 real-city prompts with greedy model-generated answers (a pseudorehearsal-style variant retaining real prompts). Our subspace penalty (weight 0.3) suppresses Atlantis components outside the rank-16 real-city activation subspace. Distance-only coordinate supervision (weight 1) penalizes normalized squared error through the frozen pre-training probe, using coordinates trilaterated from training distances and known real-city locations rather than Atlantis labels.

These are selected settings, not equal-compute family rankings: meta-updates and Hessian penalties add pre-training computation, MAML uses task-specific initializations, and extra replay increases fine-tuning steps at fixed epochs.

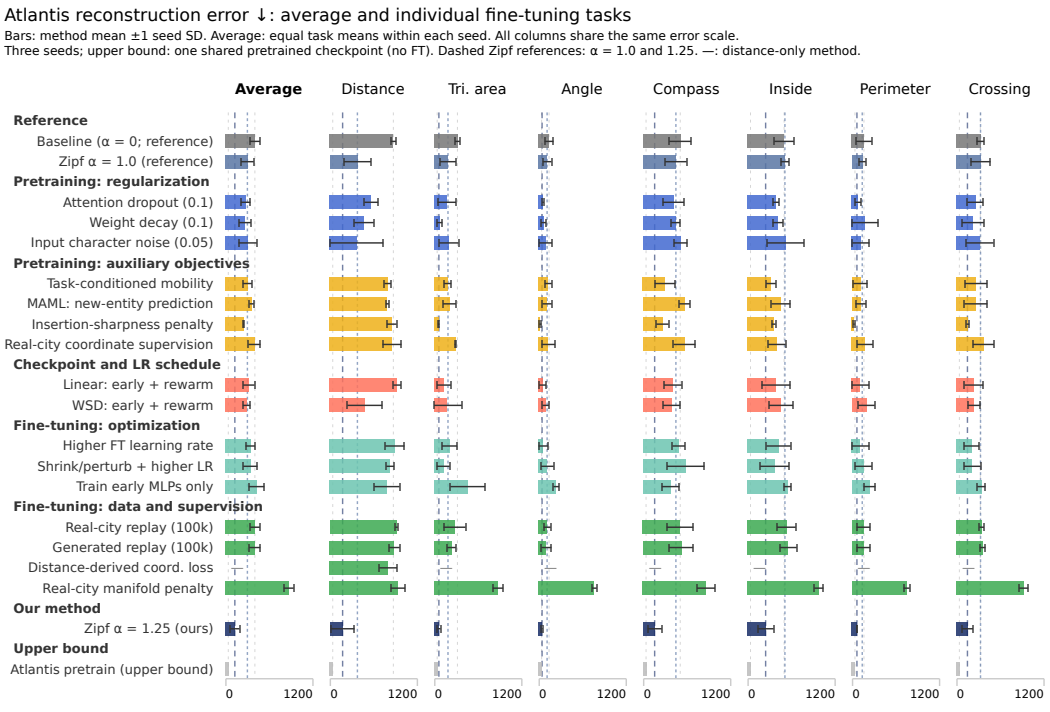


Figure 20: Individual recipes for Fig. 7: seven-task average and per-task reconstruction, with three-seed means ± sample SD. Task averaging precedes seed aggregation. Atlantis pre-training uses one checkpoint; distance-derived supervision appears only for distance. Dashed Zipf $\alpha = 1$ and 1.25 references are shown; $\alpha = 1$ is excluded from main-family statistics.

G ADAPTATION RESULTS BY FINE-TUNING TASK

These figures disaggregate all four settings in Fig. 8. Each bar averages seeds only, retaining the fine-tuning task.

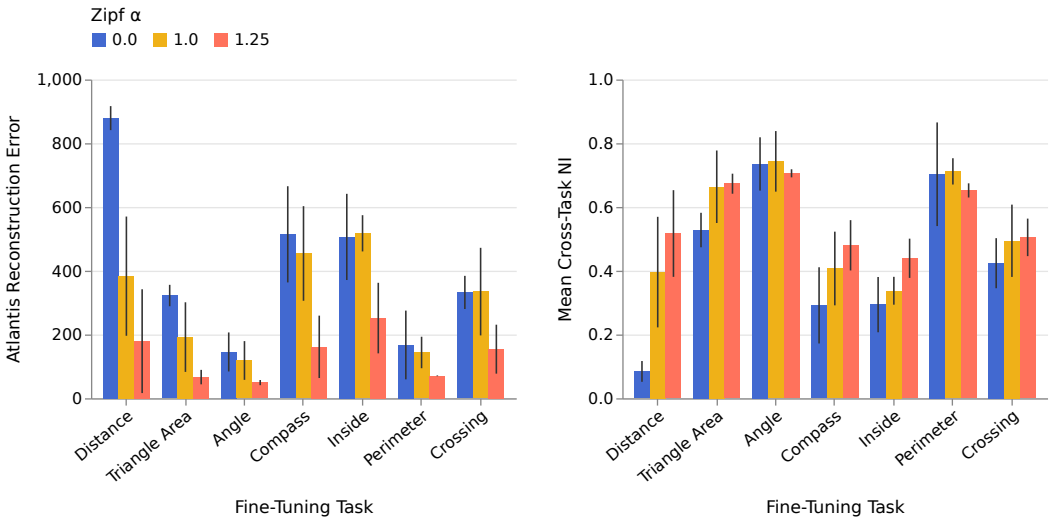


Figure 21: New-city reconstruction (left) and cross-task NI (right) by fine-tuning task for $\alpha \in \{0, 1, 1.25\}$. Bars and whiskers show three-seed means ± sample SD.

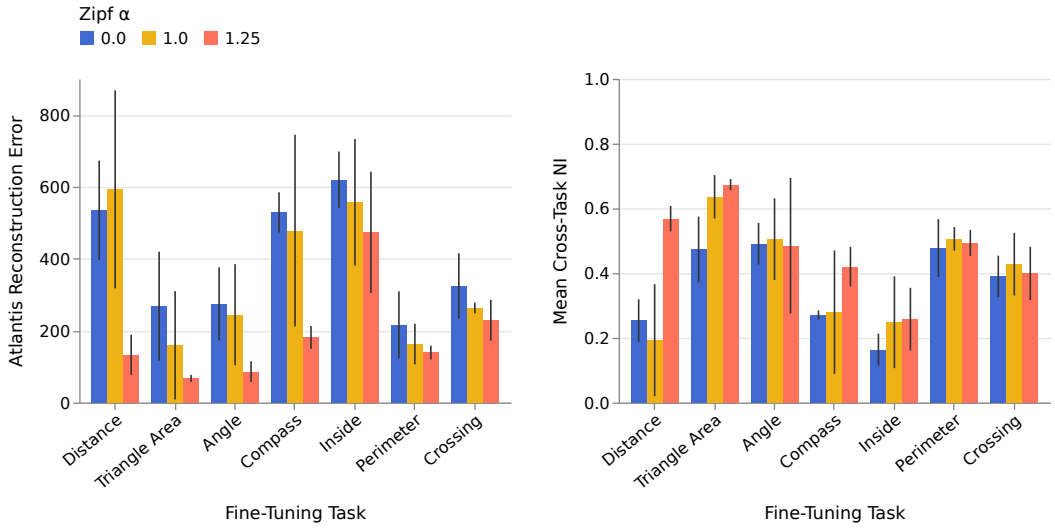


Figure 22: New cities with the fine-tuning task held out from pre-training. Reconstruction covers 87 Atlantis cities; cross-task NI averages the six pre-trained tasks. Fine-tuning includes 50,000 real-city examples of the new task (App. B.4). Bars and whiskers show three-seed means ± sample SD for $\alpha \in \{0, 1, 1.25\}$.

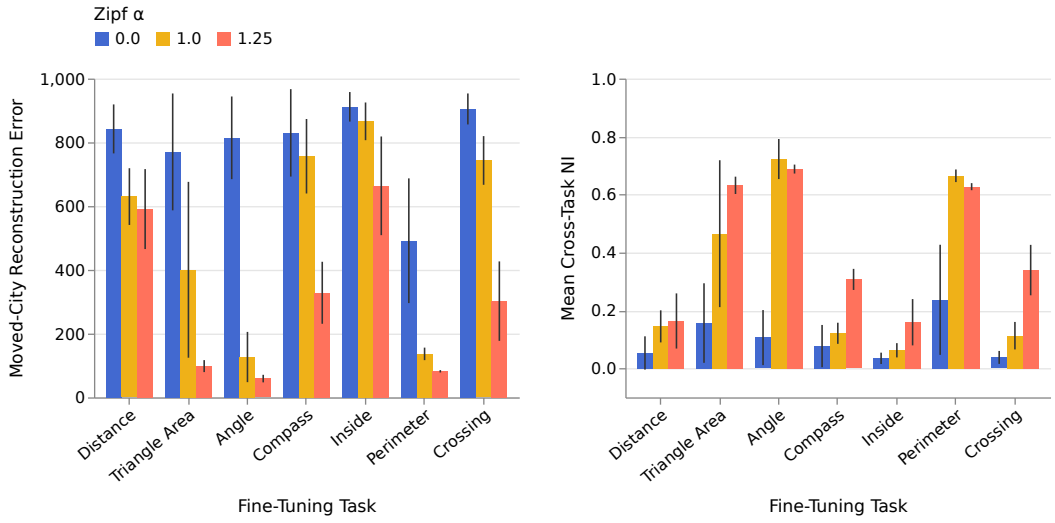


Figure 23: Relocated-city reconstruction and cross-task NI by fine-tuning task. Reconstruction covers all 100 moved cities; NI uses the six other tasks and fixed Atlantis references. Bars and whiskers show three-seed means ± sample SD for $\alpha \in \{0, 1, 1.25\}$.

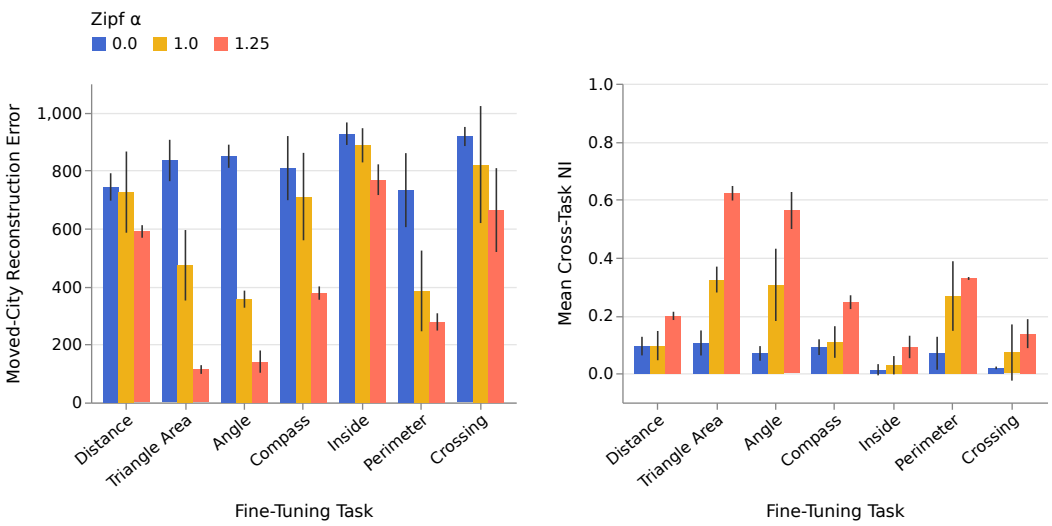


Figure 24: Relocated cities with the fine-tuning task held out from pre-training. Reconstruction covers all 100 moved cities; cross-task NI averages the six pre-trained tasks. Fine-tuning adds 50,000 unchanged-city examples of the new task. Bars and whiskers show three-seed means ± sample SD for $\alpha \in \{0, 1, 1.25\}$.

H STAGING AND SCARCITY BY FINE-TUNING TASK

Staging releases 60% of cities initially and four 10% cohorts at 20%, 40%, 60%, and 80% of training. Scarcity gives half the cities relative sampling weight 0.02; the combined recipe applies both. Zipf uses uniform fine-tuning replay, whereas scarcity replays its own corpus, limiting a causal decomposition of the Zipf effect.

The distance-only controls in Fig. 11 match row-release pool sizes, not entity exposure histories. Eight-wave staging starts with 60% of cities and releases 5% at each 10% increment of training, from 10% through 80%. Single-wave controls start with 80% of cities and release 20% halfway through training, or start with 90% and release 10% at 80%. Scarcity controls change the rare-city weight to 0.1 or 0.05, or assign cities to the rare group with probability 0.25 instead of 0.5. Tail replay adds three copies of rare-entity rows at a fixed budget of 42 million presentations; its fine-tuning replay source is unchanged.

Across seven fine-tuning tasks, staging raises cross-task NI from 0.437 to 0.629 and scarcity raises it to 0.580. The distance-only sensitivity analysis gives reconstruction errors of 197 and 175 for four- and eight-wave staging, versus 522–558 for single-wave introductions. Errors under $10\times$, $20\times$, and $50\times$ scarcity are 843, 553, and 335; applying $50\times$ downweighting to only a quarter of entities yields 630. These are three-seed means. Staging variants also change cohort size and timing, while scarcity variants use their own fine-tuning replay corpora, so the comparisons do not isolate wave count or pre-training scarcity strength alone.

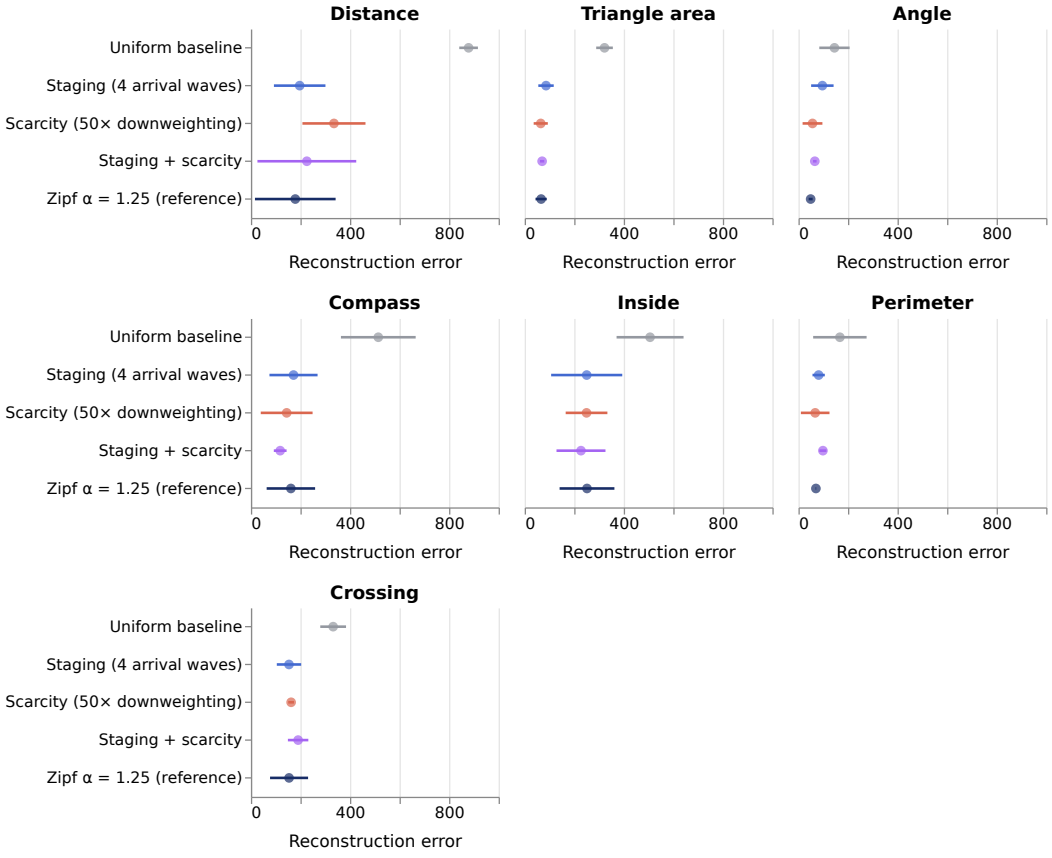


Figure 25: Atlantis reconstruction by fine-tuning task for the recipes averaged in Fig. 10. Points and bars show three-seed means ± sample SD; all tasks use the same 87 eligible Atlantis cities.

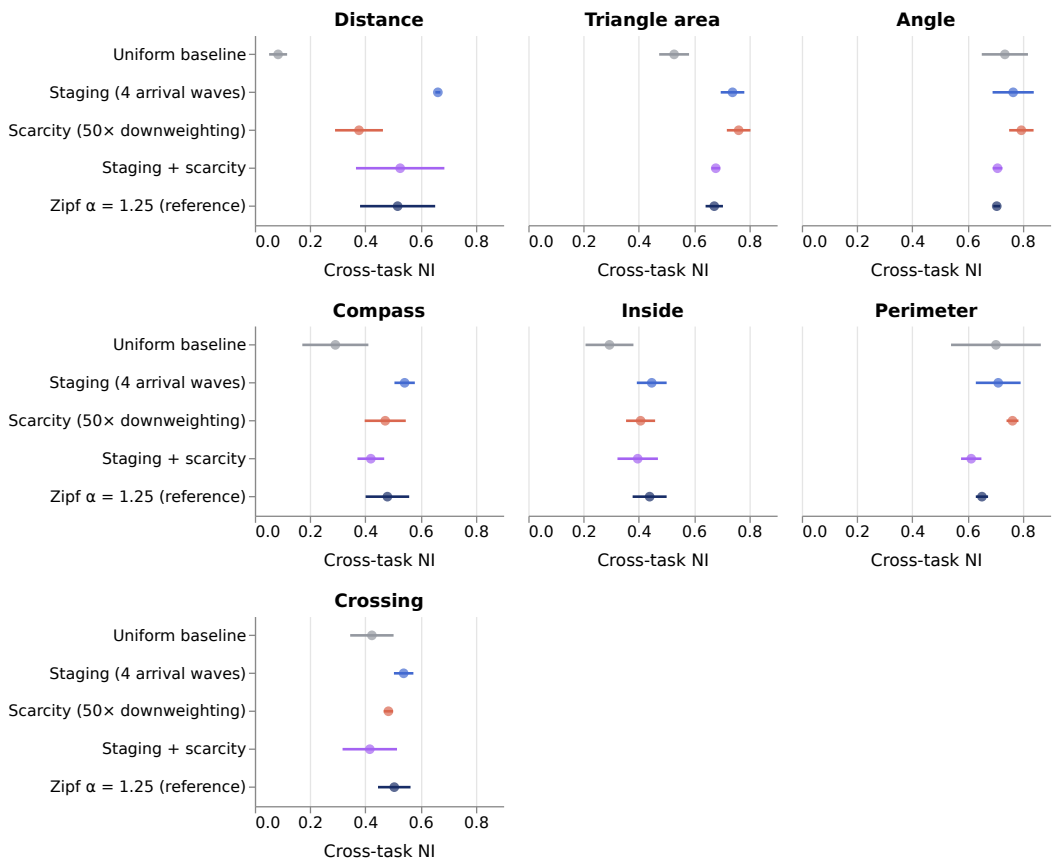


Figure 26: Cross-task NI by fine-tuning task for Fig. 10. Points and bars show three-seed means ± sample SD, retaining values outside $[0, 1]$. NI averages the six other tasks using fixed baseline/oracle references.

I FINE-TUNING ONLY SELECTED MLPS

We repeat the checkpoint experiment while updating only the MLP gate, up, and down projections in zero-indexed blocks 1–3. This window was selected before restricted-fine-tuning results using pre/post-MLP coordinate probes on final $\alpha \in \{0, 1, 1.4\}$ checkpoints across three seeds: these blocks showed the largest mean reconstruction-error reductions. Probes used the standard 3,250/1,250 real-city fit/test split and digit/comma readout. The same window is retained for $\alpha = 1.25$. This measures linear decodability, not causal necessity, and does not establish the same boundary at earlier checkpoints.

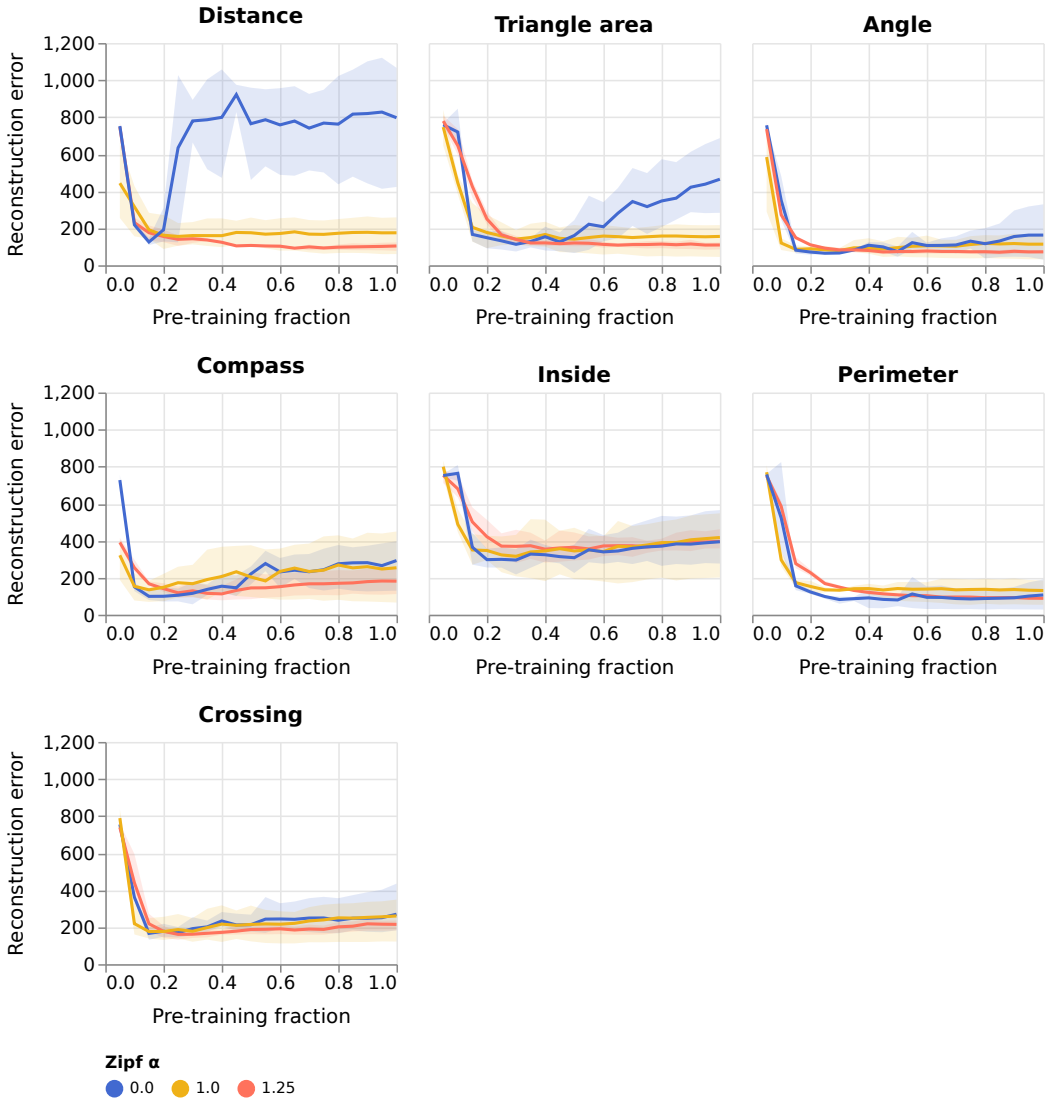


Figure 27: Atlantis reconstruction after fine-tuning only MLPs in zero-indexed blocks 1–3, disaggregated by fine-tuning task. Lines show three-seed means; ribbons show seed ranges. Data, replay, optimization, and the layer-5 probe match the full-parameter experiment in Fig. 13.