Interpretability of Language Models via Task Spaces

Lucas Weber*
University Pompeu Fabra
lucasweber000@gmail.com

Jaap Jumelet
ILLC, University of Amsterdam
jumeletjaap@gmail.com

Elia Bruni†
Osnabrück University
elia.bruni@gmail.com

Dieuwke Hupkes†
Meta
dieuwkehupkes@meta.com

Abstract

The usual way to interpret language models (LMs) is to test their performance on different benchmarks and subsequently infer their internal processes. In this paper, we present an alternative approach, concentrating on the *quality* of LM processing, with a focus on their language abilities. To this end, we construct 'linguistic task spaces' – representations of an LM's language conceptualisation – that shed light on the connections LMs draw between language phenomena. Task spaces are based on the interactions of the learning signals from different linguistic phenomena, which we assess via a method we call 'similarity probing'. To disentangle the learning signals of linguistic phenomena, we further introduce a method called 'fine-tuning via gradient differentials' (FTGD). We apply our methods to language models of three different scales and find that larger models generalise better to overarching general concepts for linguistic tasks, making better use of their shared structure. Further, the distributedness of linguistic processing increases with pre-training through increased parameter sharing between related linguistic tasks. The overall generalisation patterns are mostly stable throughout training and not marked by incisive stages, potentially explaining the lack of successful curriculum strategies for LMs.

1 Introduction

Recently, language models (LMs) have reached a level of sophistication in language production where their output is often indistinguishable from human-generated language (Liang et al., 2022). However, the complexity inherent in language production means that effective models are also inherently complex, making them challenging to interpret.

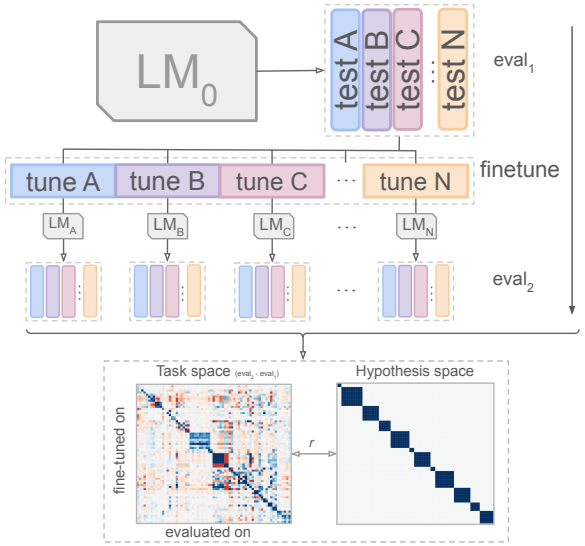


Figure 1: The process of similarity probing to obtain a task space based on transfers: 1. Evaluate the untuned LM on all tasks ($eval_1$); 2. Tune one LM for each task; 3. Re-evaluate the LMs on all tasks ($eval_2$). Calculate all transfers ($eval_2$ - $eval_1$) and compare the resulting transfer task space to a hypothesized set of transfers (Hypothesis space).

Commonly, linguistic interpretability involves assessing an LM's ability through simple evaluation tasks like grammatical acceptability judgments of various language constructions (e.g. Linzen et al., 2016; Marvin and Linzen, 2018). While these methods inform us about a model's *performance*, they do not provide insights into the *quality* of the model's solutions. This is especially the case when error analysis is not possible due to high model performance. However, it is the quality of processing that is interesting from the viewpoint of the interpretability researcher, the cognitive scientist or linguist. Here, we introduce a method to interpret the language processing of LMs holistically. We show how linguistic knowledge in LMs interconnects. We build upon the framework of Weber et al. (2021) that proposes to consider linguistic

*Now affiliated with **Fraunhofer IIS**; corresponding author

†Shared senior authorship

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 4522–4538

phenomena as 'tasks' an LM has to optimise and allows us to analyse the interactions of those tasks, similar to ideas from multi-task learning (MTL). For example, consider the following sentences:

(1) John did *not* see **anything**.

(2) *If* John sees **anything**, he will be surprised.

In both sentences, a downward-entailing environment (negation vs. conditional) allows for the negative polarity item (NPI) *anything* to be used. Understanding whether it is acceptable to produce an NPI in either sentence can be considered a different task. LMs can use different rules to solve these tasks: an LM might learn the co-occurrence statistics of certain trigger words (e.g. '*not*' vs. '*if*') with NPIs. On the other hand, it might generalise to a more abstract linguistical conceptualisation and understand that both – negation and conditionals – create downward-entailing environments permitting NPIs. With either rule, the model resolves acceptability judgements correctly, while the *quality* of both solutions is decisively different. Hence, assessing the generalisation of linguistic tasks reveals how LMs conceptualise language.

Similar to 'task spaces' in MTL (more details in § 2.1), we can represent an LM's generalisation behaviour in a *linguistic task space*, a multi-dimensional space relating linguistic tasks according to their similarity. To construct linguistic task spaces, we introduce *similarity probing*, a method to estimate linguistic similarity. This method involves selectively fine-tuning LMs on specific linguistic tasks and assessing the impact of the fine-tuning on other tasks (see Figure 1), as well as the alignment of tasks in gradient space. We extricate single linguistic tasks from their entanglement in natural language via a method we call *fine-tuning via gradient differentials (FTGD)*. FTGD selectively updates a small, relevant subspace of parameters with 'gradient differentials'.

The contributions of this paper can be summarised as follows:

1. Propose linguistic task spaces as an interpretability method for deeper model understanding and as a tool for linguistic theory testing.
2. Introduce *FTGD*, a technique to disentangle linguistic tasks from their language context and selectively fine-tune them in LMs.
3. Introduce *similarity probing*, an efficient method for generating large linguistic task spaces.
4. Analyze the development of language conceptualisation of LMs throughout pre-training by constructing language task spaces at various stages of LM pre-training.

2 Background and related work

In this section, we summarise related work and additional background on task spaces in MTL (§ 2.1), the use of LMs for linguistic theorizing (§ 2.2) and fine-tuning machine learning models in low-dimensional subspaces (§ 2.3).

2.1 Task-similarity spaces in MTL

In MTL, the transfer between different tasks is thought to be determined by their 'similarity' (Ben-David and Borbely, 2008). Constructing similarity spaces and task taxonomies to determine which tasks should be trained together has been a prominent goal in the literature. One of the earliest examples of constructing task-similarity spaces can be found in Thrun and O'Sullivan (1996). More recently, Zamir et al. (2019); Standley et al. (2020) constructed task taxonomies for computer-vision tasks based on the transferability of task-specific representations. Similarly, Achille et al. (2019) create 'task embeddings' for visual classification tasks by comparing their task structure through Fisher Information Matrices. From a theoretical perspective, Lee et al. (2021) investigate task similarity using synthetic tasks in a controlled setting, finding that their similarity measure can predict learning outcomes.

2.2 Linguistic spaces

Their ability to consistently construct acceptable language made LMs interesting, explicit linguistic theories (Baroni, 2022). However, similar to humans (Watson, 1913; Titchener, 1912; Nisbett and Wilson, 1977), LMs cannot introspectively report their internal processes. Consequently, there has been growing interest in developing methods to gain theoretical insights by analysing internal processes of LMs in what has been described as 'synthetic linguistics' (Chowdhury and Zamparelli, 2019). A collection of interpretability work assumes implicit linguistic similarity spaces in LMs revealed in their generalisation behaviour: Weber et al. (2021) demonstrate how language models generalize across linguistically related construc-

tions, suggesting an implicit task hierarchy within broad tasks like language modeling.

Chowdhury and Zamparelli (2019) observe that pre-trained models more easily learn grammatical than ungrammatical structures, showing how LMs generalise across consistent linguistic structure. Prasad et al. (2019) and Sinclair et al. (2022) use priming experiments to determine the relationship between different linguistic tasks and recover their hierarchical organization. Pérez-Mayos et al. (2021) fine-tune LMs on various downstream tasks and evaluate the effects on their syntactic understanding. Müller-Eberstein et al. (2023) probe for linguistic subspaces in language models using information theoretic probes. To our knowledge, we conduct the first attempt to *explicitly* construct a linguistic similarity space. Ultimately, our linguistic spaces are akin to knowledge representations in conceptual spaces (Gardenfors, 2004, 2014), popular in the cognitive sciences.

2.3 Fine-tuning in low-dimensional subspaces

Recently, the idea that tasks can be fine-tuned in low-dimensional subspaces of overparameterised models has gained popularity. A range of previous work shows how tasks can be effectively trained using projections into low-dimensional subspaces (Li et al., 2018; Aghajanyan et al., 2020; Gressmann et al., 2020; Hu et al., 2022; Li et al., 2022; Zhang et al., 2023). Other research demonstrates the reducibility of tasks to small subnetworks by learning discrete maskings for task-irrelevant parameters or activations (Frankle and Carbin, 2018; Mallya et al., 2018; Sanh et al., 2020; Zhang et al., 2021; Zhao et al., 2020; Csordás et al., 2020; Guo et al., 2020; Chintam et al., 2023), without explicitly projecting representations into lower-dimensional space.

3 Methods

In this paper, we connect work on task spaces in MTL (Zamir et al., 2019; Standley et al., 2020; Achille et al., 2019) with the idea of linguistic similarity spaces (Weber et al., 2021) in a method we call **similarity probing**. Similarity probing consists of three steps: First, we evaluate our untuned model on a wide range of linguistic phenomena (i.e. a 'linguistic task'). Then, we selectively tune a separate LM on each linguistic task. Finally, we evaluate each model again on all linguistic tasks and assess the tuning's impact in terms of performance transfers and compare different properties of the gradient updates (more details follow in § 3.2). Fine-tuning a linguistic phenomenon is not straightforward. For that reason, we start by introducing a modified training procedure we will call 'fine-tuning via gradient differentials' (§ 3.1).

3.1 Fine-tuning via gradient differentials (FTGD)

The major problem when fine-tuning an LM on a linguistic phenomenon is what we here call 'linguistic entanglement'.

Linguistic entanglement Within language data, linguistic tasks are necessarily interwoven (Weber et al., 2021). For example, certain tasks are present in every sentence (e.g. `subject verb agreement [SVA]`). This presents a challenge if we want to use natural language data to selectively fine-tune a separate task `A` since any potential data point to train task `A` necessarily also contains information on `SVA`. The learning signals of `A` and of `SVA` are overlapping and can not be unambiguously attributed to either task.

On the other hand, natural language data is also rich in spurious correlations between different task distributions. For example, two tasks `A` and `B` might occur in similar contexts or frequently share vocabulary in their realisations. The similarity in learning signal between `A` and `B` in such cases may solely be due to these spurious correlations instead of any conceptual similarity.

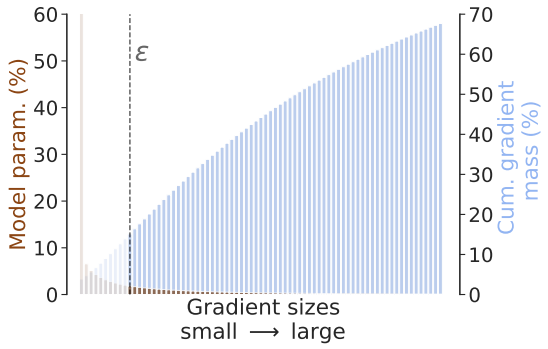
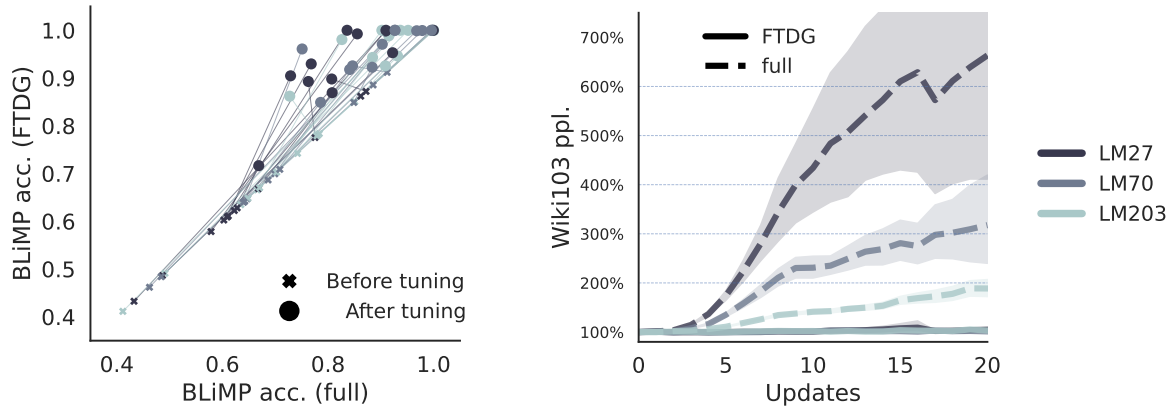


Figure 2: Parameter bins ranging from small to large gradients. Only a small amount of parameters carry the largest portion of gradient mass. Our cut-off (ϵ) maintains a large portion of gradient mass while reducing the amount of trained parameters significantly.

Disentangling linguistic tasks Our method builds on the assumption that gradients in LM training are a linear combination of an ensemble of 'sub-



(a) (b)

Figure 3: (a) BLiMP accuracy per phenomenon before and after fine-tuning using full gradients or our gradient difference method. FTGD is either as effective or more effective in improving benchmark performance on all phenomena. (b) The relative increase in perplexity (ppl) on the *wiki103* validation set during the fine-tuning process of models trained for 20 epochs. FTGD barely affects perplexity, while full gradients are highly disruptive.

gradients', representing different linguistic tasks. Following this assumption, we can isolate a specific linguistic task by subtracting two gradients that only differ in that task's specific subgradient, effectively eliminating all other subgradients. To obtain minimally different gradients, we make use of 'minimal pair' sentences. A minimal pair consists of two almost identical sentences, only distinguished by a minimal difference that renders one of them ungrammatical with respect to a specific linguistic task. An example of a minimal pair for NPI licensing through negation is

(3) John did *not* see **anything**.

(4) * John did see **anything**.

where (4) differs minimally from (3) to render it unacceptable. Using minimal pairs as training data, we proceed in the following way: At every update, we calculate the gradients $g^+(\theta)$ for the grammatical examples and $g^-(\theta)$ for their ungrammatical counterparts, with θ being our model parameters. We then calculate the gradient differentials $g^\Delta = g^+ - g^-$.

Estimating similarity in the resulting high-dimensional gradient space is challenging (see e.g. Beyer et al., 1999; Zimek et al., 2012). Therefore, we additionally reduce the gradients to a small parameter subspace by dropping parameters for which the differential $g^\Delta(\theta)$ does not differ from 0 by a margin of $\epsilon = 10^{-3}$:

$$\theta_0 = \{\theta : |g^+(\theta) - g^-(\theta)| > \epsilon\}$$

In other words, we select those parameters θ_0 where the gradients for positive and negative examples are sufficiently different. We then fine-tune the model by using only $g^\Delta(\theta_0)$. With this approach, we reduce the number of trainable parameters to an average of 5% of the full parameters while maintaining 81% of the gradient mass (see Figure 2).

3.2 Similarity Probing

We determine the similarity between tasks `A` and `B` by examining various aspects of how LMs learn them. Following the MTL literature, we concentrate on *transfer learning* (Zamir et al., 2019; Standley et al., 2020) and the *gradient alignment* (Yu et al., 2020) between `A` and `B`.

Transfer probing We determine the transfer between `A` and `B` by fine-tuning a language model on `A` and measuring the performance on a benchmark test of `B` before and after the fine-tuning (see Figure 1). Fine-tuning on task `A` may have three potential influences on task `B`. We interpret them as follows:

1. The performance of `B` increases: `A` and `B` are related and have high similarity.
2. The performance of `B` decreases: `A` and `B` are related and have low similarity.
3. The performance of `B` is unchanged: `A` and `B` are unrelated.

We normalise[1] all transfers to mitigate any floor and ceiling effects. Assessing similarity in this way

[1]For negative transfers, we normalise by the *maximally possible accuracy decrease* on the benchmark (i.e. the pre-fine-

has the advantage of being technically easier to apply and does not require a machine with a large memory.

Gradient Probing We can also directly relate tasks in the parameter space: we compare the *overlap* of different task subspaces and the *alignment* of gradients g^{Δ} within those subspaces. Following Yu et al. (2020), we predict:

1. A and B have great subspace overlap and g^{Δ} is aligned: A and B are similar and will benefit each other.
2. A and B have a great subspace overlap, but g^{Δ} is not aligned: A and B are dissimilar and will interfere.
3. A and B have a small or no subspace overlap: A and B are unrelated and will not interact.

We determine the **overlap** between subspaces of tasks A and B by calculating their Jaccard-similarity J_{sim}

$$J_{sim}(\theta_0^A, \theta_0^B) = \frac{|\theta_0^A \cap \theta_0^B|}{|\theta_0^A \cup \theta_0^B|}$$

and use cosine similarity (CS) to measure gradient alignment between tasks. The gradient-based method enables us to get a more detailed insight into the interaction between tasks.

4 Experiments

In the empirical analysis of our method, we pre-train three different generative LMs up to various stages. Then, we test the FTGD on a trained-out checkpoint to ensure that it works as intended. Subsequently, we apply it to all intermediate checkpoints to interpret the change in the LMs' language conceptualisation throughout the training process.

4.1 Training details

Data We pre-train our LMs on the standard split of a common English Wikipedia corpus (; Merity et al., 2017). For probing their linguistic ability, we use the BLiMP corpus (Warstadt et al., 2020). BLiMP consists of minimal pairs for 13 higher-level linguistic *phenomena* in the English language, which can be subdivided into 67 distinct realisations called *paradigms*. Each paradigm contains 1000 individual minimal pairs, sizing the whole corpus at 67,000 data points. For our experiments, we consider every paradigm as a separate linguistic task. We use 85% of each paradigm's data for probe training and retain 15% for evaluation.

Models and pre-training We employ decoder-based generative transformer language models based on the fairseq library (Ott et al., 2019). We consider three models of different sizes with ∼27M, ∼70M and ∼203M trainable parameters[2] respectively, with all other hyperparameters kept constant across models[3]. After 20 epochs of pre-training, we reach average final perplexities of 65.21, 38.32 and 27.61 on the *wiki103* validation set of training across 5 runs.

FTGD We adapt the setup for fine-tuning in the probing phase to avoid potential confounds: we switch to plain stochastic gradient descent (SGD) to preempt interference of Adam's momentum terms with the probing process. Additionally, we change the batch size to 850 to contain the entire training data for the probed paradigm to minimize the influence of individual data point variations on the learning signal. We fine-tune on the probed paradigm until the model's performance on the validation set converges[4] or we reach a maximum of 20 updates.

4.2 Results

We first show how FTGD compares to full-gradient fine-tuning (§ 4.2.1), continue with an analysis of the similarity probing method (§ 4.2.2) and end with the analysis of the development of similarity space throughout the LM pertaining process (§ 4.2.3).

4.2.1 FTGD

The desideratum of our method is that it improves a specific linguistic task in isolation. This requires effectiveness and selectivity: it improves the model's performance on a specific linguistic task while not interfering with unrelated capacities. We assess effectiveness by comparing FTGD with regular fine-tuning using the full gradients g^{+} and observe that the difference method achieves equivalent or higher fine-tuning performance (see Figure 3a). To assess selectivity, we compare how much both fine-tuning methods interfere with the LM's general ability to

tuning accuracy), and for positive transfers, we normalise by the *maximally possible accuracy increase* on the benchmark (i.e., 1 - the pre-fine-tuning accuracy).

[2]Hyperparameters for (LM27, LM70 and LM203): layers = (3, 6, 12), hidden- and embedding-size = (256, 512, 1024), attention-heads = (4, 8, 16), ffn size = (1024, 2048, 4096)

[3]Hyperparameters: batch size = 16, dropout = 0.1, learning rate = 0.0001; Optimiser: Adam (Kingma and Ba, 2015)

[4]The stopping criterion is defined as current performance being lower or equal to the average of the last five steps.

generate language. We measure this ability via the LM's perplexity on the *wiki103* validation set and find FTGD to be much less disruptive than full gradient fine-tuning (see Figure 3b). Both results show how FTGD is indeed effective and selective.

4.2.2 Linguistic task spaces

After applying our similarity probing method, we obtain linguistic task spaces containing similarity values between all possible pairings of BLiMP paradigms. The heatmap in Figure 1 visualises a transfer space for a LM203 model. From here, we can take two perspectives on the analysis of linguistic task spaces: we can use them to test linguistic hypotheses or to interpret the linguistic conceptualisation of the LM. While we will concentrate on the latter use for the remainder of the paper, we will touch upon the possibility of doing linguistic hypothesis testing in the discussion.

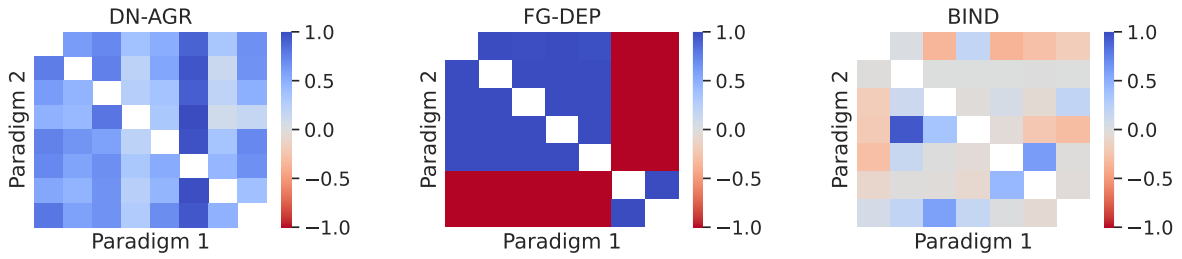
Comparing similarity measures As laid out in Section 3.2, we can construct task spaces based on different similarity measures (via performance transfers i.e. '*transfer probing*' or properties of their gradients i.e. '*gradient probing*'). Additionally, we consider multiple options to construct gradient task spaces: the overlap between gradient subspaces J_{sim} (i.e. the degree to which tasks are learned with the same parameters), the alignment of the gradients in those subspaces (using cosine similarity [CS]) or the gradient alignment weighted by the degree of subspace overlap ($J_{sim} \times CS$). In the upper part of Table 1, we compare how predictable the resulting gradient task spaces (GTS) are for the respective TTS. Subspace overlap alone (GTS$_{J_{sim}}$) yields comparatively low correlations, as sharing parameters is necessary but not sufficient for transfer to occur. Accordingly, we find the similarity of gradients within the overlapping subspaces (GTS$_{CS}$) to be much more predictive of transfers between tasks. If we now weigh the alignment by the degree of overlap (GTS$_{J_{sim} \times CS}$), we expect the GTS to become even more predictive of transfers (i.e. high alignment with larger parameter sharing should lead to higher transfer than high alignment with little parameter sharing). Surprisingly, GTS$_{J_{sim} \times CS}$ does not lead to improved prediction of transfers leaving gradient alignment as the best predictor of transfers.

[5]For WD, we report the absolute value of the correlation, as we relate a distance with a similarity measure.

Task space	Hypothesis	LM27	LM70	LM203
TTS	GTS$_{J_{sim}}$	.41$_{\pm.01}$	.41$_{\pm.01}$	.39$_{\pm.01}$
	GTS$_{CS}$	.70$_{\pm.02}$	.72$_{\pm.01}$	.69$_{\pm.02}$
	GTS$_{J_{sim} \times CS}$	.51$_{\pm.01}$	.50$_{\pm.01}$	.46$_{\pm.01}$
TTS	V. overlap	.17$_{\pm.03}$	.15$_{\pm.03}$	.16$_{\pm.01}$
	WD[5]	.13$_{\pm.02}$	.17$_{\pm.01}$	.13$_{\pm.01}$
	by phen.	.27$_{\pm.01}$	.29$_{\pm.01}$	.33$_{\pm.02}$
GTS$_{CS}$	V. overlap	.20$_{\pm.01}$	.19$_{\pm.01}$	.18$_{\pm.01}$
	WD	.20$_{\pm.01}$	.25$_{\pm.00}$	.27$_{\pm.01}$
	by phen.	.40$_{\pm.00}$	.43$_{\pm.00}$	.44$_{\pm.00}$

Table 1: Correlations between task spaces and different hypothesis spaces. The first set of rows shows the correlations of the transfer task space (TTS) with gradient task spaces (GTS). GTSs are based on various similarity metrics (Jaccard Similarity [J_{sim}]; cosine similarity [CS]; the product of J_{sim} and CS [$J_{sim} \times CS$]). GTS$_{CS}$ is the most predictive of transfers between linguistic tasks. The second and third sets of rows show the correlations of TTS and GTS$_{CS}$ with low-level controls (the shared vocabulary of different tasks [V. overlap] and the Wasserstein distance between tasks [WD]) as well as with the clustering-by-phenomena hypothesis space. Generalisation within phenomena is stronger than across low-level controls.

Global transfer patterns We can get a global idea of the type of features the LMs generalise across, by comparing the task space with a 'hypothesis space', a synthetic space representing a hypothesis that we *expect* a model to generalise across. To see whether the task spaces capture meaningful higher-level features within linguistic phenomena, or rather are due to low-level, spurious features, we generate three different types of hypothesis spaces. First, we test the clustering of paradigms into their higher-level BLiMP phenomena as it is shown in Figure 1. A high correlation of task spaces with this hypothesis space means that LMs indeed find the higher-level structural features that all paradigms from the same phenomenon have in common. We compare this against low-level, spurious controls in the form of vocabulary-level similarities between tasks (normalised vocabulary overlap [V. overlap] and the Wasserstein distance [WD] between vocabulary distributions; additional information in Appendix A.1.1). We observe that the phenomenon structure is much more predictive of the generalisation patterns than the low-level controls (see the bottom portion of Table 1), con-



(a) (b) (c)

Figure 4: Different similarity patterns within phenomena for LM203 after 20 epochs of pre-training. We find high similarity for all different paradigms in `determiner noun agreement` (a); high similarity but interfering subclusters for `filler-gap dependencies` (b); and no similarity for different `binding` paradigms (c). The exact identities of the individual rows and columns can be found in Table 3 in Appendix A.5.

	A-AGR	ARG-S	BIND	CON-R	DN-AGR	ELLIP	FG-DEP	IRR-F	ISL-E	NPI-L	QUANT	SV-AGR
LM27	0.07 ±0.19	0.02 ±0.18	0.03 ±0.19	-0.07 ±0.33	0.24 ±0.12	0.39 ±0.06	0.03 ±1.0	-0.42 ±0.41	0.13 ±0.52	0.2 ±0.44	0.19 ±0.38	0.05 ±0.32
LM70	0.08 ±0.18	0.04 ±0.4	0.03 ±0.12	-0.07 ±0.58	0.33 ±0.07	0.6 ±0.05	0.04 ±0.99	-0.62 ±0.35	0.13 ±0.43	0.18 ±0.37	0.41 ±0.39	0.18 ±0.35
LM203	0.11 ±0.36	0.07 ±0.25	0.03 ±0.26	-0.01 ±0.44	0.56 ±0.23	0.6 ±0.02	0.05 ±1.0	-0.97 ±0.05	0.21 ±0.47	0.2 ±0.36	0.48 ±0.36	0.33 ±0.12

(a)

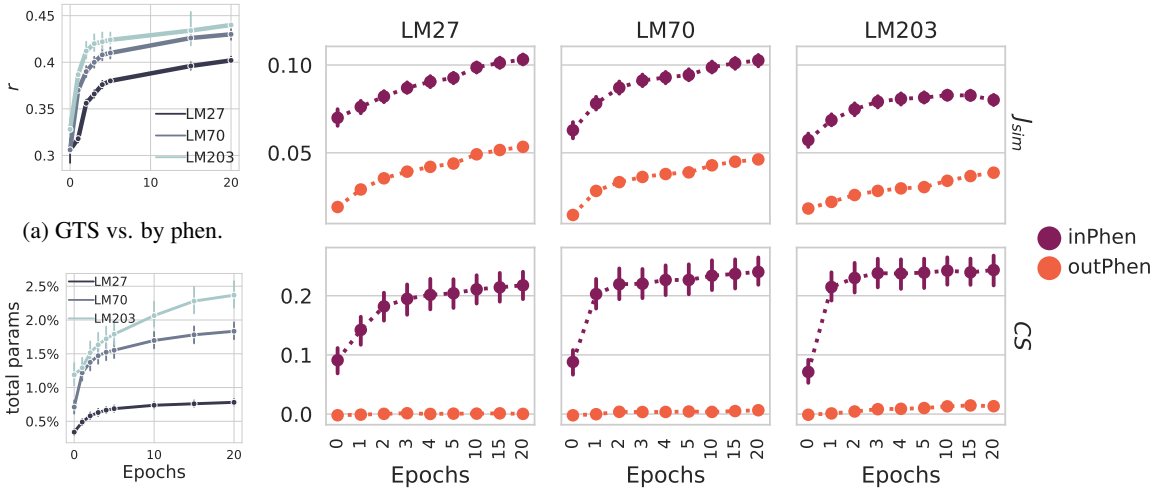
Figure 5: The degree of within-phenomena transfer for different models pre-trained for 20 epochs. A high value indicates that the model strongly generalises the phenomenon. A mapping of abbreviations to full names of phenomena can be found in Appendix A.3

firming that the models generalise related tasks beyond their shared vocabulary and that we can capture this generalisation in our task spaces. The generalisation within phenomena further increases with increasing model size as shown by higher correlations for larger models.

Individual phenomena In the previous paragraph, we found that LMs tend to generalise according to the higher-level linguistic structure globally. We can get a more differentiated picture by looking at the within-phenomena transfers for individual phenomena. As laid out in § 3.1, there are three main transfer patterns that we can expect within a phenomenon: First, the paradigms within a phenomenon have high similarity values, as we can see in the phenomenon `determiner noun agreement` [**DN-AGR**]) in Figure 4a). In this case, the model has discovered the overarching phenomena. Beyond **DN-AGR**, we observe a similar pattern for **ELLIP**, **QUANT** and **SV-AGR**, showing in the high average within-phenomenon similarity values with relatively low standard deviations in Figure 5 (see Appendix A.3 for a full table of abbreviations). Second, the model discovers the similarity between paradigms, but cannot reconcile them, as we see it in `filler-gap dependencies` [**FG-DEP**] (see Figure 4b). Those phenomena have low similarity values but very high standard deviations or negative similarity. Models find subclusters of paradigms to transfer across, but cannot reconcile the different subclusters with each other. As a consequence, the subclusters are highly interfering with each other. In irregular forms [**IRR-F**], the 'subclusters' consist of single paradigms that test different usages of lexical items with irregular morphology (e.g. as a verb vs. as an adjective). Our models do not resolve this ambiguity, leading to high interferences between the tasks. Third, for some phenomena, we do not observe any interactions between their paradigms (see, e.g. `binding` [**BIND**] in Figure 4c). The LM finds idiosyncratic solutions to all the paradigms and does not discover the more general phenomenon. With increasing size, models tend more towards the first pattern, solving paradigms more by generalising to the higher-level phenomenon.

4.2.3 Analysis of the training process

We have established that linguistic task spaces inform us about LMs language conceptualisation. But how do they change throughout training? Performance on the BLiMP benchmark increases starkly in early training (for learning curves, see Appendix A.2.1). What can we learn from task spaces beyond that observation?



(a) GTS vs. by phen.

(b) Average subspace size

(c) Average subspace overlap and gradient alignment

Figure 6: (a) Correlation of GTS with clustered-by-phenomena space throughout training; (b) The development of subspace size throughout language model training. (c) (Top) The average J_{sim} of task-subspaces either within the same phenomenon or outside the phenomenon; (bottom) the average inner product of Δg of the overlapping subspaces. Larger models align related paradigms much faster and to a higher degree than smaller models.

Development of task spaces We construct similarity spaces for nine model checkpoints throughout pre-training (see Appendix A.5 for visualisations of all similarity spaces). Overall, we find that similarity spaces are remarkably stable: similarity patterns are present from very early in training (within the first epochs; for details, see Appendix A.4). At the same time, the generalisation within phenomena continuously increases, showing a continuous reinforcement of the existing generalisation pattern without major structure shifts in the pattern (see Figure 6a).

The continuity of generalisation patterns is surprising and contrasts with human learning, which is marked by learning stages (Piaget et al., 1952; Gopnik et al., 1999, 2004): as humans deepen their knowledge, new patterns emerge. Language learning in LMs is not marked by such incisive shifts.

Development of subspaces Another interesting angle of analysis is the change of the subspaces θ_0 in which the model is learning specific paradigms. The average size of $|\theta_0|$ continuously grows during pre-training (see Figure 6b), showing that LMs learn linguistic tasks initially more localised but become more distributed throughout training. While the relative subspace overlap J_{sim} increases generally with training, the within-phenomenon overlap is overall higher (see Figure 6c top). Additionally, the alignment of gradients increases selectively for paradigms from the *same* phenomenon (see Figure 6c bottom). This shows how the processing of linguistic tasks starts idiosyncratic (separated and in non-aligned subspaces) and with training, the sharing of structure increases (shared and in aligned subspaces, where appropriate).

4.3 Linguistic hypothesis testing

In addition to testing linguistic task spaces against hypothesis spaces with known structural similarities (such as 'clustering by phenomena' or our vocabulary controls), we can also use them to test *assumed* similarities in linguistic structure. We can construct a hypothesis space that represents contested ideas in linguistic theory and test whether our LM generalises according to the hypothesis by calculating a simple correlation. Our methodology is a step towards model-based theorising or 'synthetic linguistics' (Chowdhury and Zamparelli, 2019). Doing linguistic hypothesis testing, however, is beyond the scope of the current paper.

5 Discussion

In this paper, we construct linguistic task spaces, representing an LM's language conceptualization, which can be used for linguistic hypothesis testing and as a holistic interpretability tool. We introduce FTGD to selectively fine-tune latent, entangled concepts such as linguistic tasks, and 'similarity prob-

ing' to estimate similarities between linguistic tasks through their transfer learning and gradient analysis. We analyse the resulting similarity spaces of LMs throughout pre-training to interpret their learning process.

We find that learning of linguistic tasks begins localized and becomes more distributed with training, with increased parameter sharing among linguistic tasks and gradient alignment especially between linguistically related tasks. Learning theory suggests that a trained model requires fewer dimensions due to increasingly efficient compression rules (*e.g.* Cheng et al., 2023). Opposing the assumed reduction in *intrinsic* dimension, training in our experiments actually *increases* the number of *extrinsic* dimensions on which a task is learned. Intrinsic and extrinsic dimensions might be inversely related in language models, as previously observed by Aghajanyan et al. (2021). A more distributed processing of tasks allows for more overarching structure sharing and generalisation across different subconcepts, which potentially helps to achieve lower *intrinsic* dimensionality. Furthermore, we find that generalization patterns remain surprisingly stable throughout pre-training, without stark shifts to new patterns—a behaviour more typical of human-like learning. This potentially reflects the weakness of classical neural network models to generalise systematically (Hupkes et al., 2020; Lake and Baroni, 2023, see also Lake and Baroni (2018); Ettinger et al. (2018); Bahdanau et al. (2019); Keysers et al. (2019); Yu and Ettinger (2020); Kim and Linzen (2020); Press et al. (2022)). Future generations of LMs employing more human-like learning paradigms (see e.g. Lake and Baroni, 2023) may exhibit stronger shifts in generalization patterns. The observed continuity might explain the lack of successful curriculum learning strategies for language modelling in the past (see e.g. Surkov et al., 2022; Campos, 2021; Weber et al., 2023): in a learning process without notable shifts in generalisation patterns, changes in the data distribution during training are not beneficial.

Future reseach Beyond language, our approach to interpreting LM conceptualisation can be applied to other domains to better understand the current weaknesses of LMs, such as numerical reasoning and cross-lingual concept learning. Furthermore, the potential for explicit linguistic hypothesis testing, though underexplored in this paper, can help bridge the gap between formal linguistic and computational linguistic research. Large, state-of-the-art LLMs may uncover subtle structural similarities that are informative to linguists.

6 Limitations

There are several limitations to the presented methods. First, our fine-tuning and evaluation data are i.i.d. and come from a very narrow distribution: the data are not natural but synthetic, and all data are generated using the same templates. We use this very narrow i.i.d. data to asses the fine-tuning success during probing. However, we cannot be entirely sure whether we succeeded in fine-tuning a specific linguistic task rather than some idiosyncracies of the narrow data distribution. While our FTGD approach might elevate this issue slightly, it does not dispel our doubts completely. The optimal way to guarantee our results would be the evaluation on a set from a separate distribution.

Second, while our approach applies to all types of knowledge domains, it requires *minimal pairs* of tasks within that domain to fine-tune them selectively. Minimal pairs are primarily used in linguistics and are uncommon in other knowledge domains.

Third, as discussed in the previous section, a major weakness of our probing approach lies in the necessary top-down definition of 'anchors' that we use to span the space. We utilise human-defined tasks and relate them to each other. However, a more accurate linguistic space can probably be described by 'anchors' that are defined through the model itself and span the conceptual space with maximal expressivity.

Acknowledgements

We thank the COLT group at UPF for the discussions and their useful feedback; Further, LW thanks the Department of Translation and Language Sciences at the University Pompeu Fabra for funding. Further, this research was possible through computational resources acquired through a grant of the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101019291). This paper reflects the authors' view only, and the ERC is not responsible for any use that may be made of the information it contains. Ultimately, we thank the anonymous reviewers for their time and useful feedback!

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2vec: Task embedding for meta-learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6429–6438. IEEE.

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

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*.

Dzmitry Bahdanau, Harm de Vries, Timothy J O'Donnell, Shikhar Murty, Philippe Beaudoin, Yoshua Bengio, and Aaron Courville. 2019. Closure: Assessing systematic generalization of clevr models. *arXiv preprint arXiv:1912.05783*.

Marco Baroni. 2022. On the proper role of linguistically-oriented deep net analysis in linguistic theorizing. *Algebraic structures in natural language*, pages 1–16.

Shai Ben-David and Reba Schuller Borbely. 2008. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine learning*, 73:273–287.

Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1999. When is "nearest neighbor" meaningful? In *Database Theory—ICDT'99: 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7*, pages 217–235. Springer.

Daniel Campos. 2021. Curriculum learning for language modeling. *ArXiv preprint*, abs/2108.02170.

Emily Cheng, Corentin Kervadec, and Marco Baroni. 2023. Bridging information-theoretic and geometric compression in language models.

Abhijith Chintam, Rahel Beloch, Willem Zuidema, Michael Hanna, and Oskar Van Der Wal. 2023. Identifying and adapting transformer-components responsible for gender bias in an english language model. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 379–394.

Shammur Absar Chowdhury and Roberto Zamparelli. 2019. An LSTM adaptation study of (un)grammaticality. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 204–212, Florence, Italy. Association for Computational Linguistics.

Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. 2020. Are neural nets modular? inspecting functional modularity through differentiable weight masks. *arXiv preprint arXiv:2010.02066*.

Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. 2018. Assessing composition in sentence vector representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1790–1801.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

Peter Gardenfors. 2004. *Conceptual spaces: The geometry of thought*. MIT press.

Peter Gardenfors. 2014. *The geometry of meaning: Semantics based on conceptual spaces*. MIT press.

Alison Gopnik, Clark Glymour, David M Sobel, Laura E Schulz, Tamar Kushnir, and David Danks. 2004. A theory of causal learning in children: causal maps and bayes nets. *Psychological review*, 111(1):3.

Alison Gopnik, Andrew N Meltzoff, and Patricia K Kuhl. 1999. *The scientist in the crib: Minds, brains, and how children learn.* William Morrow & Co.

Frithjof Gressmann, Zach Eaton-Rosen, and Carlo Luschi. 2020. Improving neural network training in low dimensional random bases. *Advances in Neural Information Processing Systems*, 33:12140–12150.

Demi Guo, Alexander M Rush, and Yoon Kim. 2020. Parameter-efficient transfer learning with diff pruning. *arXiv preprint arXiv:2012.07463*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? (extended abstract). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 5065–5069. ijcai.org.

Leonid V Kantorovich. 1960. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2019. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*.

Najoung Kim and Tal Linzen. 2020. Cogs: A compositional generalization challenge based on semantic interpretation. In *2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 9087–9105. Association for Computational Linguistics (ACL).

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Brenden M Lake and Marco Baroni. 2023. Human-like systematic generalization through a meta-learning neural network. *Nature*, pages 1–7.

Sebastian Lee, Sebastian Goldt, and Andrew M. Saxe. 2021. Continual learning in the teacher-student setup: Impact of task similarity. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6109–6119. PMLR.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. 2018. Measuring the intrinsic dimension of objective landscapes. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Tao Li, Lei Tan, Zhehao Huang, Qinghua Tao, Yipeng Liu, and Xiaolin Huang. 2022. Low dimensional trajectory hypothesis is true: Dnns can be trained in tiny subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3411–3420.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *ArXiv preprint*, abs/2211.09110.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Arun Mallya, Dillon Davis, and Svetlana Lazebnik. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European conference on computer vision (ECCV)*, pages 67–82.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Max Müller-Eberstein, Rob Van Der Goot, Barbara Plank, and Ivan Titov. 2023. Subspace chronicles: How linguistic information emerges, shifts and interacts during language model training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13190–13208.

Richard E Nisbett and Timothy D Wilson. 1977. Telling more than we can know: Verbal reports on mental processes. *Psychological review*, 84(3):231.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Laura Pérez-Mayos, Roberto Carlini, Miguel Ballesteros, and Leo Wanner. 2021. On the evolution of syntactic information encoded by BERT's contextualized representations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2243–2258, Online. Association for Computational Linguistics.

Jean Piaget, Margaret Cook, et al. 1952. *The origins of intelligence in children*, volume 8. International Universities Press New York.

Grusha Prasad, Marten van Schijndel, and Tal Linzen. 2019. Using priming to uncover the organization of syntactic representations in neural language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Victor Sanh, Thomas Wolf, and Alexander Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 33:20378–20389.

Arabella Sinclair, Jaap Jumelet, Willem Zuidema, and Raquel Fernández. 2022. Structural persistence in language models: Priming as a window into abstract language representations. *Transactions of the Association for Computational Linguistics*, 10:1031–1050.

Trevor Standley, Amir Roshan Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio

Savarese. 2020. Which tasks should be learned together in multi-task learning? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9120–9132. PMLR.

Maxim Surkov, Vladislav Mosin, and Ivan Yamshchikov. 2022. Do data-based curricula work? In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 119–128, Dublin, Ireland. Association for Computational Linguistics.

Sebastian Thrun and Joseph O'Sullivan. 1996. Discovering structure in multiple learning tasks: The tc algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 489–497. Morgan Kaufmann.

Edward B Titchener. 1912. The schema of introspection. *The American Journal of Psychology*, 23(4):485–508.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

John B Watson. 1913. Psychology as the behaviorist views it. *Psychological review*, 20(2):158.

Lucas Weber, Jaap Jumelet, Elia Bruni, and Dieuwke Hupkes. 2021. Language modelling as a multi-task problem. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2049–2060, Online. Association for Computational Linguistics.

Lucas Weber, Jaap Jumelet, Paul Michel, Elia Bruni, and Dieuwke Hupkes. 2023. Curriculum learning with adam: The devil is in the wrong details. *arXiv preprint arXiv:2308.12202*.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. 2019. Taskonomy: Disentangling task transfer learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 6241–6245. ijcai.org.

Xiongyi Zhang, Jan-Willem van de Meent, and Byron C Wallace. 2021. Disentangling representations of text by masking transformers. *arXiv preprint arXiv:2104.07155*.

Zhong Zhang, Bang Liu, and Junming Shao. 2023. Fine-tuning happens in tiny subspaces: Exploring intrinsic task-specific subspaces of pre-trained language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1701–1713, Toronto, Canada. Association for Computational Linguistics.

Mengjie Zhao, Tao Lin, Fei Mi, Martin Jaggi, and Hinrich Schütze. 2020. Masking as an efficient alternative to finetuning for pretrained language models. *arXiv preprint arXiv:2004.12406*.

Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. 2012. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387.

A Appendix

The supplementary material to this paper contains additional information about the control hypothesis spaces that we employ to verify the meaningfulness of our linguistic spaces (Appendix A.1). Further, we document the development of the LMs' performance on BLiMP in different scenarios (Appendix A.2). Ultimately, we show all heatmaps for all transfer and gradient spaces for all models throughout the whole training process (Appendix A.5).

A.1 Controls

We include control conditions and baselines for our experiments. This appendix section provides additional details.

A.1.1 Vocabulary baselines

We calculate two baselines to estimate the amount of transfer that is due to mere vocabulary overlap between different paradigms:

1. *Normalised vocabulary overlap (NVO)* between the vocabularies V_A and V_B of paradigms A and B – calculated simply as the size of their intersection normalised by the maximum vocabulary overlap between any paradigms X and Y:

$$NVO = \frac{|V_A \cap V_B|}{\max(|V_X \cap V_Y|)}$$

2. *Wasserstein distance* (WD; Kantorovich, 1960) between the vocabularies distributions. These vocabulary controls can be correlated with any task space or hypothesis space. For example, the correlation between these controls and the transfer task spaces § 4.2.2 indicates how much of the transfer between different paradigms can be attributed to the vocabulary overlap between tasks alone.

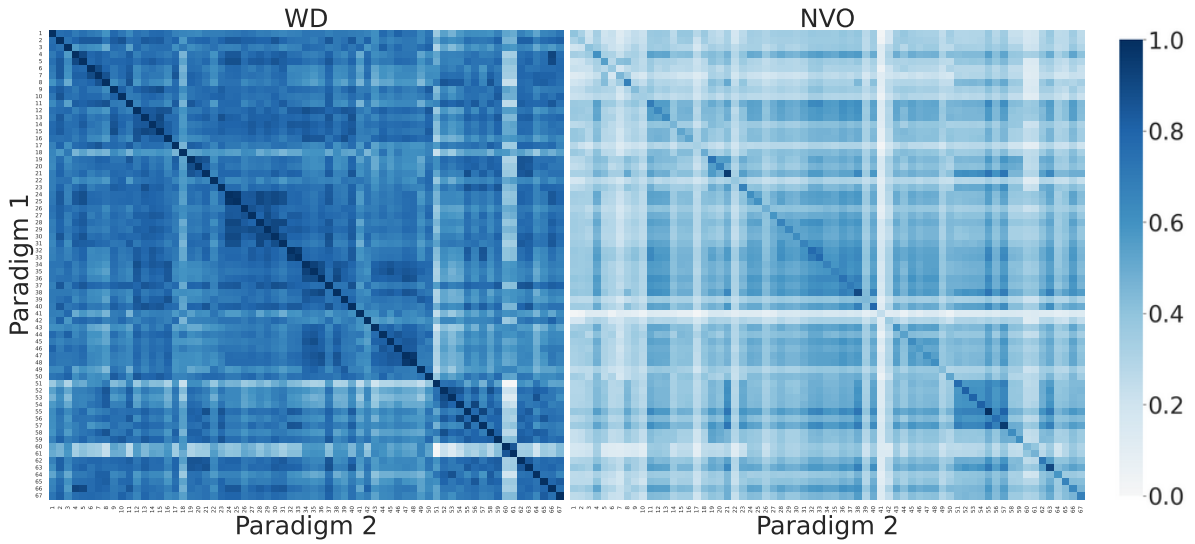


Figure 7: (1 - WD) in the left heatmap and the normalised vocabulary overlap on the right. Labels to individual rows and columns can be found in Table 2

A.2 BLiMP performance

Throughout our experiments, we pre-train and fine-tune our LMs. We here document the performance of the models in different scenarios: first, we show how the models perform on the whole benchmark throughout the pre-training process. Second, we show how different pre-training checkpoints adapt.

A.2.1 BLiMP learning curves

During the pre-training process, we evaluate each saved checkpoint on all paradigms of the BLiMP benchmark and average the results. The following plot shows the respective learning curves for the different models. While none of the models achieve very good performance, the largest model achieves their final performance much faster than the smaller ones.

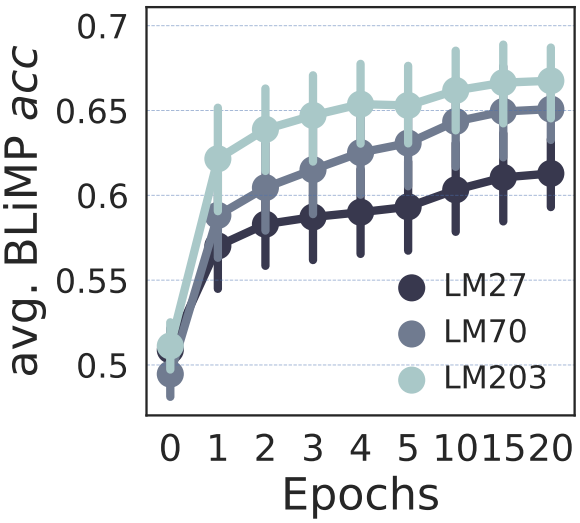


Figure 8: Learning curves of our LMs on the BLiMP benchmark averaged across all paradigms and seeds.

A.2.2 BLiMP probe tuning

The final performance after fine-tuning a specific task changes with the amount of pre-training. The final performance of that model for that specific task is shown in Figure 9. With more pre-training, models adapt better during the fine-tuning. Larger models generally adapt better than smaller models. FTGD works better for models that are pre-trained for longer. This makes sense, as the method requires the difference between minimal pairs to be meaningful (i.e. it requires previous knowledge already contained in the model parameters). The subspace selection will be more accurate as a consequence.

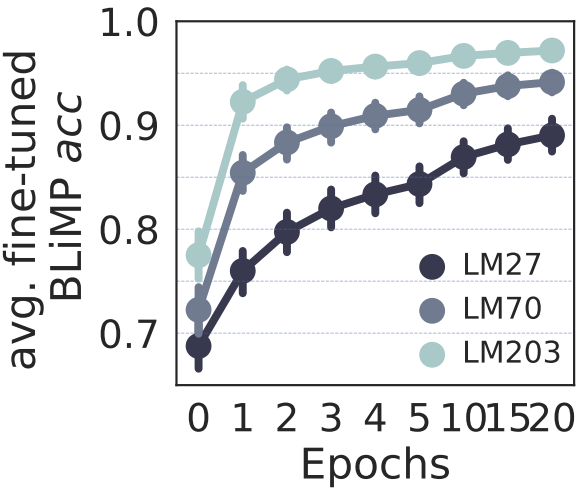


Figure 9: Average final performance after FTGD a linguistic task.

A.3 BLiMP abbreviation map

Abbreviation	Phenomenon	Abbreviation	Phenomenon
A-AGR	Anaphor Agreement	ARG-S	Argument Structure
BIND	Binding	CON-R	Control Raising
DN-AGR	Determiner Noun Agreement	ELLIP	Ellipsis
FG-DEP	Filler Gap Dependency	IRR-F	Irregular Forms
ISL-E	Island Effects	NPI-L	NPI Licensing
QUANT	Quantifiers	SV-AGR	Subject Verb Agreement

Table 2: Mapping of abbreviations to linguistic phenomena.

A.4 Similarity space stability

We here show the correlation of similarity spaces of different epochs with the final similarity space (epoch 20). Generally, similarity spaces are remarkably stable and do not show larger shifts in generalisation patterns.

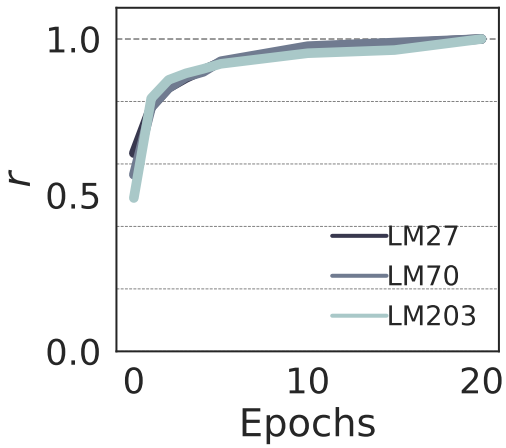


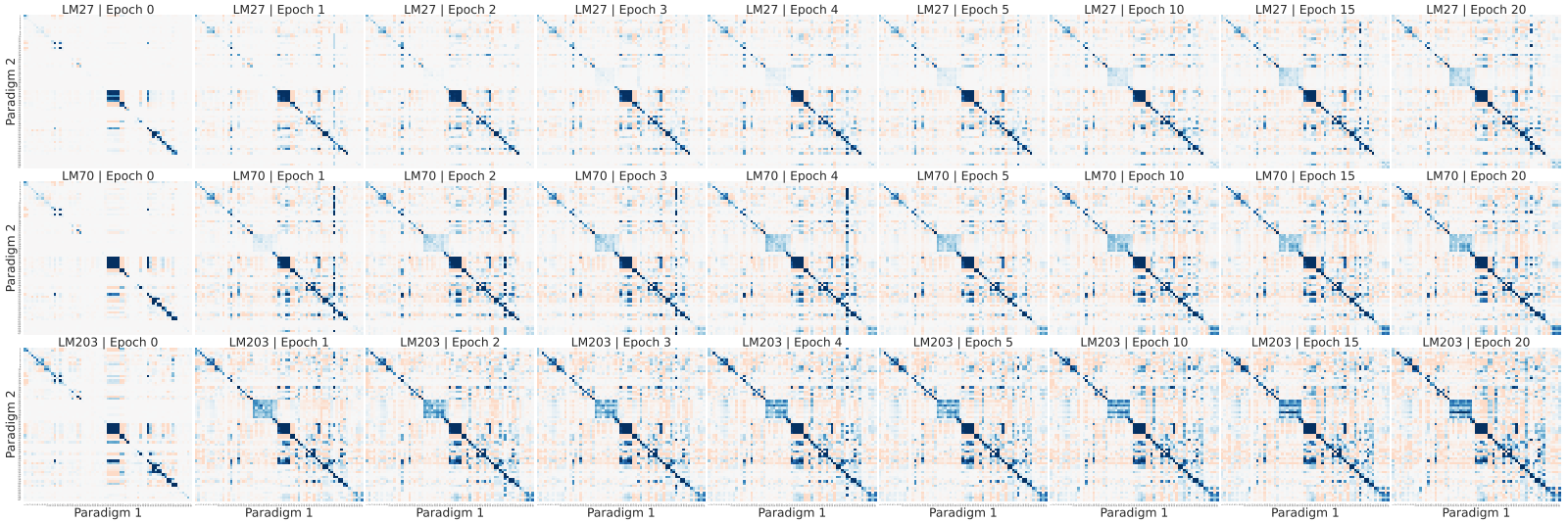
Figure 10: Correlation of gradient similarity spaces with the trained-out gradient space. Correlation is very high after only a few epochs, indicating that the overall pattern of gradient similarities only changes minimally.

A.5 Details similarity spaces

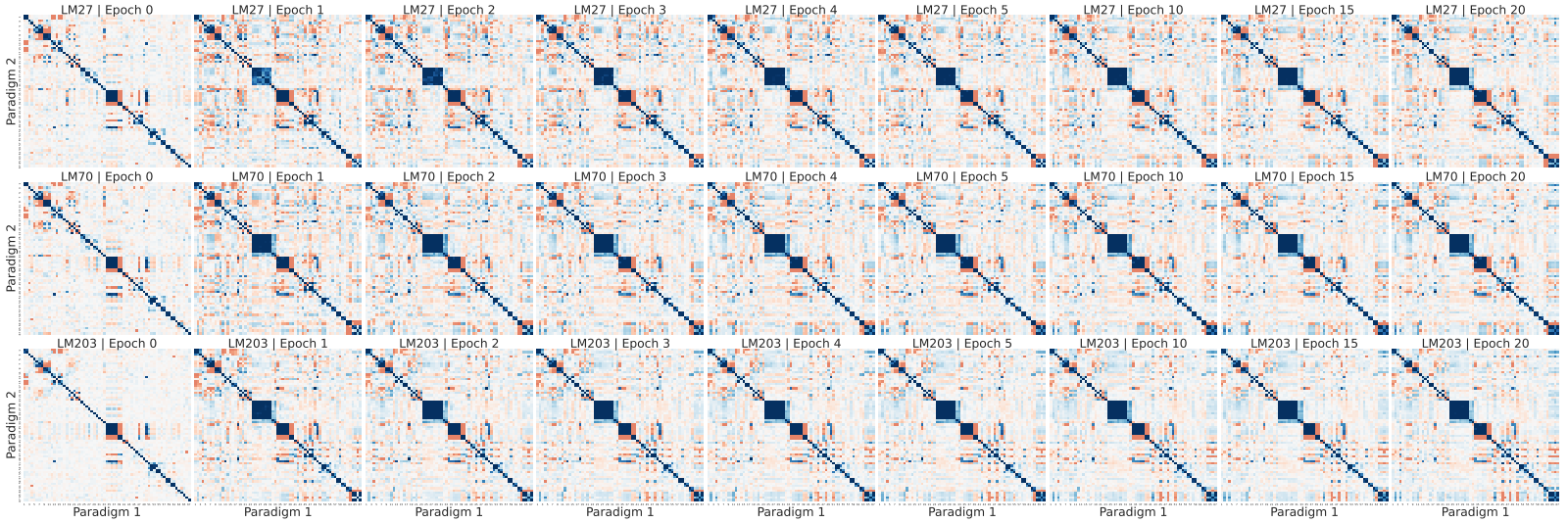
This section contains additional information about similarity spaces that we constructed throughout the paper.

A.5.1 Similarity spaces through training

We construct all similarity spaces throughout the training process. Figure 11 on the following page illustrates the transfer and gradient matrices for each saved model checkpoint. The respective indices for the rows and columns of the heatmaps can be found in Table 3 on the subsequent page.



(a) TTS for all models throughout training.



(b) GTS for all models throughout training.

Figure 11

A.5.2 Paradigm/index map

A complete list of BLiMP phenomena, paradigms and the respective indices for the heatmaps throughout the paper.

Phenomenon	Paradigm	Index
anaphor agreement	anaphor gender agreement	1
	anaphor number agreement	2
argument structure	animate subject passive	3
	animate subject trans	4
	causative	5
	drop argument	6
	inchoative	7
	intransitive	8
	passive 1	9
	passive 2	10
	transitive	11
binding	principle A c command	12
	principle A case 1	13
	principle A case 2	14
	principle A domain 1	15
	principle A domain 2	16
	principle A domain 3	17
	principle A reconstruction	18
control raising	existential there object raising	19
	existential there subject raising	20
	expletive it object raising	21
	tough vs raising 1	22
	tough vs raising 2	23
determiner noun agreement	determiner noun agreement 1	24
	determiner noun agreement 2	25
	determiner noun agreement irregular 1	26
	determiner noun agreement irregular 2	27
	determiner noun agreement with adj 2	28
	determiner noun agreement with adj irregular 1	29
	determiner noun agreement with adj irregular 2	30
	determiner noun agreement with adjective 1	31
ellipsis	ellipsis n bar 1	32
	ellipsis n bar 2	33
filler gap dependency	wh questions object gap	34
	wh questions subject gap	35
	wh questions subject gap long distance	36
	wh vs that no gap	37
	wh vs that no gap long distance	38
	wh vs that with gap	39
	wh vs that with gap long distance	40
irregular forms	irregular past participle adjectives	41
	irregular past participle verbs	42
island effects	adjunct island	43
	complex NP island	44
	coordinate structure constraint complex left branch	45
	coordinate structure constraint object extraction	46
	left branch island echo question	47
	left branch island simple question	48
	sentential subject island	49
	wh island	50
NPI licensing	matrix question NPI licensor present	51
	NPI present 1	52
	NPI present 2	53
	only NPI licensor present	54
	only NPI scope	55
	sentential negation NPI licensor present	56
	sentential negation NPI scope	57
quantifiers	existential there quantifiers 1	58
	existential there quantifiers 2	59
	superlative quantifiers 1	60
	superlative quantifiers 2	61
subject verb agreement	distractor agreement relational noun	62
	distractor agreement relative clause	63
	irregular plural subject-verb agreement 1	64
	irregular plural subject-verb agreement 2	65
	regular plural subject-verb agreement 1	66
	regular plural subject-verb agreement 2	67

Table 3: List of phenomena and paradigms