Structure Inducing Pre-Training

Anonymous Author(s)
Affiliation
Address
email

Abstract

Language model pre-training (LMPT) has been incredibly impactful in natural language processing; however, LMPT methods have not been as successful when generalized to other domains, such as biomedical domains. To understand this disparity, we first ask what properties of natural language make language modeling so successful and whether or not these hold in other domains. Next, we ask to what extent existing pre-training methods are explicitly designed to account for these discrepancies. We find that the question of how existing pre-training methods impose relational structure in their induced, per-sample latent spaces—*i.e.*, what constraints do pre-training methods impose on the distance between the pre-trained embeddings of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$—is both significantly understudied and important for LMPT performance in non-NLP domains. To address this, we introduce a descriptive framework for pre-training that illustrates how relational structure can be induced. We demonstrate the utility of this framework through theoretical and empirical analyses showing that this approach can offer meaningful improvements over existing methods across various domains and tasks.

1 Introduction

Pre-trained language models such as BERT [17] or GPT-III [5] have revolutionized the way we approach natural language processing (NLP) due to their ability to rapidly produce models for novel downstream tasks in a highly effective and data-efficient manner. Such pre-training (PT)/fine-tuning (FT) approaches are also of great interest in machine learning for the sciences (in particular, the biomedical sciences) due to the prevalence of few-shot learning problems over these modalities. However, designing effective PT/FT methods for these domains remains a major challenge. In this work, we attempt to help solve this challenge by introducing a new, descriptive framework for PT methods that offers concrete, theoretically motivated guidance on designing PT objectives to maximize suitability to a given class of FT tasks.

Motivating this framework is the ongoing phenomenon of researchers attempting to adapt language model pre-training methods from NLP to biomedical domains. Researchers have developed numerous methods which employ generative, intra-sample, imputation-based objectives to pre-train models—*e.g.*, autoregressive or mask-based sequence models for protein sequences or other biological sequences [84, 88, 97], biomedical graphs [37] or medical timeseries [70, 125]—despite the fact that such objectives alone typically offer no guidance at the per-sample level (*e.g.*, masked-imputation language modelling for BERT alone does not directly constrain the output of the `[CLS]` whole-sample embedding). While this disparity is fine for NLP, where many downstream tasks of interest can be re-framed as per-token, language modeling tasks (*e.g.*, through prompting [91]), there is little to no reason that such pre-training methods will capture the appropriate per-sample structure in other

Submitted to NeurIPS 2021 AI for Science Workshop.

domains to generalize effectively to per-sample fine-tuning tasks—*e.g.*, protein remote homology prediction in structural biology, drug side effect prediction for molecular graphs, or patient medical outcomes for medical time series.

Our framework, structure-inducing pre-training (SIPT), addresses this problem by introducing a novel pre-training loss that induces the relational structure of a target, pre-training similarity graph G_{PT} into the per-sample latent space. In this way, users can explicitly design different pre-training methods (by changing the graph G_{PT}) to capture the structure of the domain in different ways. Our framework is simultaneously general enough to capture many existing pre-training methods (including methods like BERT [17], supervised multi-task pre-training methods like MT-DNN [63], or contrastive pre-training methods like DeCLUTR [26]), specific enough to permit concrete theoretical analyses relating the structure of graph G_{PT} to downstream FT performance, and flexible enough for us to define new pre-training methods with richer guarantees than existing approaches.

Empirically, we find that building methods using rich graphs G_{PT} via our framework can offer consistent performance improvements over existing pre-training methods across diverse datasets, modalities, and fine-tuning tasks. Our method can also be used successfully on NLP datasets, showing that although the language modeling objective is quite powerful in that setting, we can still find further improvements by injecting rich, per-sample structure through our method.

In the remainder of this work, we do the following. First, in Section 2, we expand on the central argument outlined here: that existing pre-training methods lack a sufficient objective to induce structure in the per-sample latent space. We support this argument through both analysis and a quantitative review of over 90 existing studies on NLP and NLP-derived PT methods. Next, in Section 3, we introduce our new pre-training framework, SIPT. We provide a formal definition of our framework, as well as a theoretical guarantee of how the structure of the graph G_{PT} relates to downstream task performance. Finally, in Section 4, we demonstrate the practical utility of this framework via real-world experiments across three biomedical domains and 10 different fine-tuning tasks. In all cases, we compare both against baseline methods that only leverage intra-sample objectives (*e.g.*, language modelling alone) and against existing methods that additionally leverage a weaker form of per-sample structure (*e.g.*, a supervised per-sample PT objective), finding in all cases that SIPT methods match or outperform these baselines.

2 Motivation

2.1 The importance of per-sample PT tasks in non-NLP domains

Language models, such as BERT [17], and their derived methods, operate over domains that consist of whole samples (*e.g.*, entire text passages), which themselves consist of individual sub-units or tokens (*e.g.*, tokens). These models are commonly trained through some variant of a generative imputation method over these internal sub-units, possibly in concert with auxiliary losses leveraging embeddings of entire samples. For example, BERT is pre-trained using both a masked language modeling objective, which leverages per-token embeddings to predict the identities of artificially masked tokens and a next-sentence prediction (NSP) objective, which classifies if the overall text passage consists of sub-passages that occurred in the presented order in the raw data.

These models typically produce both embeddings of whole samples and of individual tokens, thus generating both a "per-sample" and a "per-token" latent space (See Appendix Definitions 1-2), and can be fine-tuned for both per-sample and per-token tasks. In general, generative, intra-sample imputation objectives (like language modeling) strongly constrain the output structure of the per-token latent space and are further likely to be at least somewhat related to downstream tasks at the per-token level. However, it is also clear that language modeling objectives alone will (in general) neither necessarily constrain the per-sample latent space nor offer insight into how to solve downstream per-sample tasks. For natural language, we can use techniques like prompting to re-frame per-sample tasks as per-token, language-modeling tasks, but approaches of this nature do not exist for non natural language domains (Figure 1).

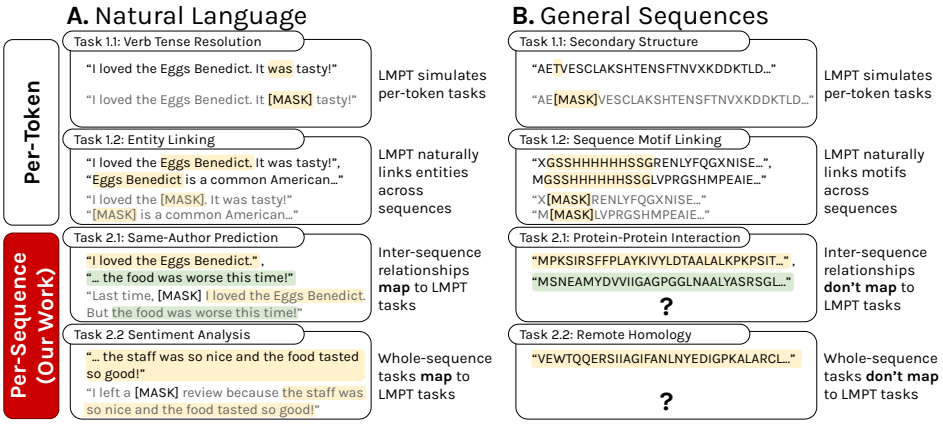

Figure 1: **A**: NLP language modeling objectives are clearly related to both per-token and per-sample downstream tasks. **B**: For other domains, "language modelling" objectives may only offer information at the per-token level.

This analysis suggests that if we want to design PT methods for scientific domains, in particular where per-sample, fine-tuning tasks are of interest, then we need to design new algorithms with per-sample, PT objectives that are related to these eventual downstream tasks. Of course, for PT, we still wish the resulting model to be general across tasks within a domain, so we can re-frame this question as *whether or not we can design PT methods that induce structure in the per-sample latent space that reflects relationships of interest for the target domain*.

2.2 Weaknesses of existing per-sample PT tasks

Researchers have devoted significant efforts to developing per-sample PT objectives. For example, BERT's NSP task is a per-sample binary classification objective. In order to understand if/why these per-sample objectives may be insufficient for non-NLP PT, we conducted a review of over 90 NLP and NLP-derived PT methods. We classified existing per-sample objectives on two independent axes: whether or not they constrain the per-sample latent space in an *implicit* or *explicit* manner, and whether or not they do so only *shallowly* or *deeply*.

Explicit methods constrain structure in such a way that one can directly reason about the relational structure of the per-sample latent space at optimality. For example, under a supervised classification objective such as BERT's NSP task [17], one can explicitly quantify how similar two samples will be in the latent space (under an inner-product similarity metric) based on whether or not they share the same PT task label. In contrast, under a contrastive learning PT objective defined by a noising/augmentation procedure such as DeCLUTR [26], one cannot explicitly reason about how two independent samples will be related in the output latent space.

Shallow methods constrain structure in such a manner that the constraints could be fully satisfied even in a very low-dimensional per-sample latent space. For example, a binary classification supervised pre-training task requires only a single dimension to satisfy at optimality, as samples can merely be mapped such that the sign of their output corresponds to their pre-training label.

With this breakdown, we find that while there has been significant research on how to design effective PT systems, the vast majority of that research has focused on systems that either impose no per-sample pre-training objectives; impose explicit, but shallow objectives; or impose deep, but implicit objectives. There is thus a marked gap in research on per-sample pre-training losses that are at once explicit and deep, or generalizable frameworks explaining how to design such objectives.

Our framework, Structure-Inducing Pre-training (SIPT) is designed precisely to fill this gap. For full details on our review, formal definitions of the terms introduced above, and quantitative analyses of the papers analyzed, please see Appendix Sections B.

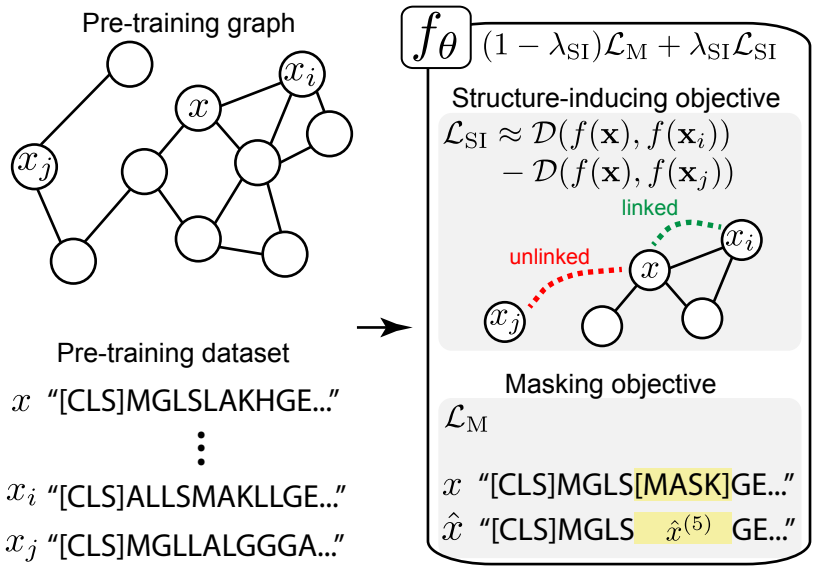

Figure 2: **Our Pre-training (PT) Framework**: We re-cast the PT formulation by taking a pre-training graph G_{PT} as an auxiliary input. G_{PT} is used to define a new structure-inducing objective $\mathcal{L}_{\mathrm{SI}}$, which pushes a pre-training encoder $f_{\boldsymbol{\theta}}$ to embed samples such that samples are close in the latent space if and only if they are linked in G_{PT}.

3 Our New PT Framework: Structure-Inducing Pre-training (SIPT)

3.1 General Pre-Training Problem Formulation

Given a dataset $\boldsymbol{X}_{\mathrm{PT}} \in \mathcal{X}^{N_{\mathrm{PT}}}$, a PT method aims to learn an encoder $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Z}$ such that $f_{\boldsymbol{\theta}}$ can be transferred to FT tasks that are unknown at pre-training time. While we can leverage additional information at PT time to inform the training of $f_{\boldsymbol{\theta}}$ (*e.g.*, PT-specific labels $\boldsymbol{Y}_{\mathrm{PT}}$), the encoder $f_{\boldsymbol{\theta}}$ *must take only samples from $\mathcal{X}$ as inputs so that it can be used for fine-tuning*. Pre-training methods typically solve this problem by training $f_{\boldsymbol{\theta}}$ to minimize a pre-training loss $\mathcal{L}_{\mathrm{PT}}$ over $\boldsymbol{X}_{\mathrm{PT}}$.

3.2 SIPT Framework

Our pre-training problem framework includes two small, but important, differences from the standard formulation (Figure 2).

First, we assume that we have as an additional input to the PT problem a graph $G_{\mathrm{PT}} = (V, E)$ where vertices denote pre-training samples within $\boldsymbol{X}_{\mathrm{PT}}$ (*e.g.*, $\{\boldsymbol{x}_{\mathrm{PT}} | \boldsymbol{x}_{\mathrm{PT}} \in \boldsymbol{X}_{\mathrm{PT}}\} \subseteq V$) and edges represent user-specified relationships. Importantly, while we take the graph G_{PT} an input to the PT problem, *we cannot use it as a direct input to $f_{\boldsymbol{\theta}}$*. Just like in traditional pre-training, $f_{\boldsymbol{\theta}}$ must take as input only samples from $\mathcal{X}$. *This is because otherwise, we can not apply $f_{\boldsymbol{\theta}}$ to the same, general class of FT tasks over domain $\mathcal{X}$.*

Second, we decompose the PT loss $\mathcal{L}_{\mathrm{PT}}$ into two components, weighted with hyperparameter $0 \leq \lambda_{\mathrm{SI}} \leq 1$:

$$\mathcal{L}_{\mathrm{PT}} = (1 - \lambda_{\mathrm{SI}})\mathcal{L}_{\mathrm{M}} + \lambda_{\mathrm{SI}}\mathcal{L}_{\mathrm{SI}}.$$

$\mathcal{L}_{\mathrm{M}}$ is a traditional, intra-sample objective (*e.g.*, a language model), and $\mathcal{L}_{\mathrm{SI}}$ is a new, structure-inducing objective designed to regularize the per-sample latent space geometry in accordance with the relationships (edges) in G_{PT}. Under our framework, $\mathcal{L}_{\mathrm{SI}}$ is only allowable for G_{PT}, $f_{\boldsymbol{\theta}}$, and $\mathcal{Z}$ if it permits some stable optima at which point a radius nearest-neighbor connectivity algorithm under some distance function in $\mathcal{Z}$ will recover G_{PT} (see the formal constraint in Appendix C.1). Note that this constraint strikes a connection between our framework and the wealth of existing research focused on *graph representation learning* [24, 16, 60, 59, 48, 32]. These techniques do indeed offer

valuable insights into how to sample minibatches over graph-structured data and devise losses for graph embeddings; however, many methods for actually modeling graph-structured data, including deep attributed graph embeddings and graph convolutional neural networks, should not be seen as replacements for our techniques here as they are typically not adaptable to contexts in which the graph is not known at inference time, and so *they could not be used in our pre-training setting where* $f_{\boldsymbol{\theta}}$ *must take in only inputs from* $\mathcal{X}$ *directly*.

As the new loss term added $\mathcal{L}_{\text{SI}}$ is explicitly designed to *induce the structure of* G_{PT} *in* $\mathcal{Z}$, we call methods trained under our framework *structure-inducing pre-training* (SIPT) methods. In Appendix C we show that many existing PT approaches can be re-realized as SIPT methods, including classification-based PT objectives, contrastive methods, or existing graph alignment methods.

3.3 Theoretical Analyses

Under our framework, one can link the structure of the PT graph G_{PT} to eventual FT task performance. In particular, as a SIPT embedder f over graph G_{PT} approaches optimality under the loss $\mathcal{L}_{\text{SI}}$, it produces an embedding space such that nearest-neighbor performance for any downstream task is lower bounded by the performance that could be obtained via a nearest neighbor algorithm over graph G_{PT} (Theorem 1). This fact directly connects the geometry of the graph G_{PT} with the eventual fine-tuning performance of a SIPT embedder f. Furthermore, it demonstrates the advantage of employing an *explicit* constraint rather than an implicit one; by controlling the structure of G_{PT}, users can directly choose to add different inductive biases to the PT process, in a manner that has a provable impact on the eventual suitability for downstream FT tasks.

Theorem 1. Let $\boldsymbol{X}_{\text{PT}}$ be a PT dataset, G_{PT} be a PT graph, and let $f_{\boldsymbol{\theta}^*}$ be an encoder pre-trained under a PT objective permissible under our framing that realizes a $\mathcal{L}_{\text{SI}}$ value no more than ℓ^*. Then, under embedder f, the nearest-neighbor accuracy for a FT task y converges as dataset size increases to at least the local consistency (Definition 5) of y over G_{PT}.

In Appendix Section D.1, we establish Corollaries 1-2 that illustrate the importance of choosing graphs G_{PT} which impose *deep* structural constraints. We provide complete proofs for all theoretical results and semi-synthetic experiments validating this theory in Appendix Sections D and E.

4 Validating SIPT via Real-world Experiments

4.1 Experimental Goals and General Procedures

This section shows that SIPT methods leveraging rich pre-training graphs G_{PT} consistently match or outperform comparable pre-training baselines. To do so, we pre-train new models across three separate biomedical domains under the SIPT framework. In each case, we use rich pre-training graphs G_{PT} defined from publicly available data and designed to capture different forms of inductive biases relevant to the respective domains. We assess the efficacy of these models via fine-tuning performance across a battery of fine-tuning tasks in each domain. Furthermore, we compare these models against comparable existing pre-training methods, including (wherever possible) methods leveraging both primarily intra-sample pre-training objectives (*e.g.*, language modeling) and existing methods that augment language model pre-training with a per-sample objective (*e.g.*, supervised classification given a set of pre-training labels). We also perform ablation studies and hyperparameter tuning analyses to assess the extent to which the SIPT objective is responsible for resulting performance gains.

4.2 Datasets and Tasks

We examine three data modalities for our experiments: PROTEINS, containing protein sequences; ABSTRACTS, containing free-text biomedical abstracts; and NETWORKS, containing sub-graphs of protein-protein interaction (PPI) networks. These modalities, and the datasets we use to assess each modality, are summarized in Table 1.

	PROTEINS	ABSTRACTS	NETWORKS
Data Modality ($\boldsymbol{x}_i$ is a...)	Protein Sequence	Biomedical Paper Abstract	PPI Network Ego-graph
PT Dataset	Tree-of-life [144]	Microsoft Academic Graph [108, 36]	[37]
G_{PT}: $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ iff	$\boldsymbol{x}_i$ interacts with $\boldsymbol{x}_j$	$\boldsymbol{x}_i$'s paper cites $\boldsymbol{x}_j$'s paper	$\boldsymbol{x}_i$'s central protein agrees on all but 9 Gene Ontology (GO) labels with $\boldsymbol{x}_j$'s central protein.
Per-token baseline	TAPE [84]	SciBERT [4]	Attribute Masking [37]
Per-sample baseline	PLUS [73]	None	Multi-task learning [37]
FT Dataset	TAPE [84]	SciBERT [4]	[37]

Table 1: A summary of our datasets, tasks, and benchmarks. For example, for the PROTEINS domain, our pre-training dataset is the set of protein sequences contained in the tree-of-life dataset [144], proteins are linked in our pre-training graph G_{PT} if and only if they interact according to the tree-of-life graph, and we compare over the fine-tuning tasks in the TAPE benchmark against both the raw, per-token baseline publicly available in the TAPE model [84] as well as the per-sample baseline published in the PLUS pre-training model [73].

FT Dataset	FT Task Name	Abbr.	Description	Metric
TAPE [84]	Remote Homology	RH	Per-sequence classification task to predict protein fold category.	Accuracy
	Secondary Structure	SS	Per-token classification task to predict amino acid structural properties.	Accuracy
	Stability	ST	Per-sequence, regression task to predict stability.	Spearman's ρ
	Fluorescence	FL	Per-sequence, regression task to predict fluorescence.	Spearman's ρ
	Contact Prediction	CP	Intra-sequence classification to predict which pairs of amino acids are in contact in the protein's 3D conformation.	Precision @ $L/5$
SciBERT [4]	Paper Field	PF	Per-sentence classification problem to predict a paper's area of study from its title.	Macro-F1
	SciCite	SC	Per-sentence classification problem to predict citation intent	Macro-F1
	ACL-ARC	AA	Per-sentence classification problem to predict citation intent	Macro-F1
	SciERC	SRE	Per-sentence relation extraction	Macro-F1
NETWORKS [37]			Multi-label binary classification into 40 Gene Ontology terms	Macro-AUROC

Table 2: Fine-tuning tasks.

Domain	Task	Vs. Per-Token PT		vs. Per-Sample	
		RRE	Δ	RRE	Δ
PROTEINS	RH	**7.0%** ±1.2	↑	**8.4%** ±2.4	↑
	FL	-0.8% ±1.3	∼	**12.8%** ±1.1	↑
	ST	**13.1%** ±2.5	↑	2.2% ±2.8	∼
	SS	**4.5%** ±0.2	↑	**4.5%** ±0.2	↑
	CP	**10.5%** *	↑	N/A	
ABSTRACTS	PF	0.3% ±0.2	∼	N/A	
	SC	2.4% ±4.1	∼	N/A	
	AA	**17.7%** ±6.5	↑	N/A	
	SRE	**6.7%** ±0.4	↑	N/A	
NETWORKS		7.8% ±5.2	∼	5.1% ±2.7	↑

Table 3: Relative reduction of error (RRE; defined to be $\frac{\text{[baseline error]}-[G_{\text{PT}}\text{ model error]}}{\text{[baseline error]}}$) of models trained under our framework vs. published per-token or per-sample baselines. Higher numbers indicate models under our framework reduce error more and thus outperform baselines. The Δ column indicates whether the model offers a statistically significant improvement (↑), no significant change (∼), or a statistically significant decrease (↓). Statistical significance is assessed via a t-test at significance level $p < 0.1$. Per-sample analysis and variance estimates for CP were infeasible due to the computational cost of this task.

The PROTEINS and ABSTRACTS domains are sequential domains; in each of these, intra-sample language model pre-training tasks can be used as natural baselines. In the NETWORKS domain, where each sample consists of a subgraph of a protein-protein interaction graph, not a sequence, "language modeling" is adapted to a task of masked node prediction, as in [37].

Table 1 also lists all FT benchmarks we use for validation, and FT tasks across all domains are also detailed in Table 2. In all cases, we adopt a traditional transfer-learning approach, where the entire pre-training encoder is transferred, and only the fine-tuning task-specific head is learned from scratch. Further details can also be found in Appendix section F.

4.3 $\mathcal{L}_{\text{SI}}$ and Training Procedures

As discussed in the definition of our framework (Section 3.2), a SIPT method differs from a standard PT method by (1) the choice of graph G_{PT} (Table 1) and (2) the design of the new, structure-inducing loss $\mathcal{L}_{\text{SI}}$. To define $\mathcal{L}_{\text{SI}}$ in our experiments, we leverage ideas from *structure-preserving metric learning* (SPML) [107, 95, 94]. SPML is a form of metric learning where positive relationships are defined by edges in a graph rather than a shared supervised label. We adapt two losses, a traditional contrastive loss [31] and a multi-similarity loss [113], from supervised metric learning to the graph-based, structure-preserving context of $\mathcal{L}_{\text{SI}}$ terms in SIPT.

In addition to these losses, in the ABSTRACTS and PROTEINS domains, we use a warm-start procedure to initialize pre-training from existing language models rather than beginning from scratch. This saves significant computational time and allows for a powerful ablation study to isolate performance improvements to the introduction of our $\mathcal{L}_{\text{SI}}$ term. We also perform extensive hyperparameter tuning studies on these two domains to identify appropriate values for λ_{SI}, and adapt those findings to the NETWORKS domain. Further details about the experimental setup, including formal statements of our contrastive and multi-similarity losses, are in Appendix Section F.

4.4 Results

SIPT matches or outperforms baselines across all 3 domains and 10 FT tasks

We compute the relative reduction of error [1] of the best performing SIPT model vs. the per-token or per-sample baselines across all FT tasks (Table 3). *We can see that in 10/15 cases, SIPT improves*

[1] $\text{RRE} = \frac{(1-\text{baseline})-(1-\text{SIPT})}{(1-\text{baseline})}$. RRE is unitless and suitable for comparing across tasks.

over existing methods, and in no case does it do worse than either baseline. We see improvements of up to approximately 17% (0.05 macro-F1 raw change) on AA, 6% on SRE (0.01 macro-F1 raw change), and 4% on RH (2% accuracy raw change). *SIPT models further establish a new SOTA on AA and RH and match SOTA on FL, ST, & PF.* In appendix Figure 6, we also show how performance evolves as a function of FT iterations for the NETWORKS dataset, demonstrating performance gains accrue rapidly during fine-tuning and are stable throughout training. Raw results for all settings are present in Appendix Section F.9.

Note that these performance gains, calculated both against standard per-token approaches and methods that impose additional per-sample objectives, persist (though for NETWORKS, the gains are not quite statistically significant) over all three data modalities and all different G_{PT} types we use here. This shows that explicitly regularizing the per-sample latent space geometry offers value across NLP, non-language sequences, and non-sequential domains, as well as while leveraging graphs including those defined by external knowledge, by self-supervised signals in the data directly, and by nearest-neighbor methods over multi-task label spaces.

Observed gains are uniquely attributable to the novel loss $\mathcal{L}_{\mathrm{SI}}$

As outlined in the Methods section, our experimental design permits us to determine how much of the observed gains in Table 3 are due to the novel loss component, as opposed to, for example, continued training, new PT data, or the batch selection procedures used in our method which also indirectly leverage the knowledge inherent in G_{PT}. Unsurprisingly, some gains are observed due to these other factors, and performance gains shrink when considering these ablation studies. However, even when comparing against the maximal performance baseline or ablation study overall, neither the direction of observed relationships nor the statistical significance of observed comparisons changes. *Therefore, we can conclusively state that the performance improvements observed here are uniquely attributable to the new, structure-inducing components introduced by our framework.* Full ablation study results can be found in the Methods section (Tables 7-8).

5 Conclusion

We show that despite the breadth of research into PT methods, methods for imposing *explicit* and *deep* structural constraints over the per-sample, pre-training latent space $\mathcal{Z}$ are under-explored (Figure 4). Our theoretical and empirical analyses *show that this deficit matters in practice*. In particular, we define a new pre-training framework, *structure-inducing pre-training* (SIPT), under which the PT loss is subdivided into two components: one which is designed to capture intra-sample (*e.g.* per-token) relationships and one which is designed to constrain the per-sample latent space to capture relationships between samples given by a user-specified pre-training graph G_{PT}. Under our framework, we show both theoretically and via experiments that the structure induced in $\mathcal{Z}$ can be directly connected to eventual fine-tuning performance. Empirically, we show that novel SIPT methods leveraging a variety of pre-training graphs can consistently outperform compelling existing PT methods across three real-world domains.

Our work highlights several important directions for future research. For example, are there losses better suited than metric learning losses for pre-training graphs—*e.g.*, can we leverage the graph distance alongside the intra-batch distance to improve negative sampling strategies? In addition, can we produce theoretical results on the convergence of pre-trained models? Can we advance the understanding of when and how pre-trained models converge to solutions that recover G_{PT}? In a different direction, can pre-trained models reflect forms of structure beyond nearest neighbor relationships—*e.g.*, such as by leveraging higher-order topological considerations or by matching a distance function rather than a discrete graph? We anticipate that further analyses of these and other questions will lead to new pre-training methods and enable pre-training to be successful across scientific domains.

References

[1] Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online, June 2021. Association for Computational Linguistics.

[2] Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, (12), 2019.

[3] Parishad BehnamGhader, Hossein Zakerinia, and Mahdieh Soleymani Baghshah. Mg-bert: Multi-graph augmented bert for masked language modeling. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 125–131, 2021.

[4] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A Pretrained Language Model for Scientific Text. In *EMNLP*, 2019.

[5] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[6] Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew Peters, Arie Cattan, and Ido Dagan. CDLM: Cross-document language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2648–2662, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[7] Iacer Calixto, Alessandro Raganato, and Tommaso Pasini. Wikipedia entities as rendezvous across languages: Grounding multilingual language models by predicting Wikipedia hyperlinks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3651–3661, Online, June 2021. Association for Computational Linguistics.

[8] Fredrik Carlsson, Amaru Cuba Gyllensten, Evangelia Gogoulou, Erik Ylipää Hellqvist, and Magnus Sahlgren. Semantic re-tuning with contrastive tension. In *International Conference on Learning Representations*, 2021.

[9] Bo Chen, Jing Zhang, Xiaokang Zhang, Xiaobin Tang, Hong Chen, Cuiping Li, Peng Zhang, Jie Tang, et al. Code: Contrastive pre-training with adversarial fine-tuning for zero-shot expert linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11846–11854, 2022.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[11] Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online, June 2021. Association for Computational Linguistics.

[12] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *ICLR*, 2019.

[13] Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. Structural Scaffolds for Citation Intent Classification in Scientific Publications. In *Proceedings of the 2019 Conference of the NAACL: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[14] Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel Weld. SPECTER: Document-level representation learning using citation-informed transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2270–2282, Online, July 2020. Association for Computational Linguistics.

[15] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics.

[16] Ganqu Cui, Jie Zhou, Cheng Yang, and Zhiyuan Liu. Adaptive graph encoder for attributed graph embedding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 976–985, 2020.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019.

[18] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[19] Keyur Faldu, Amit Sheth, Prashant Kikani, and Hemang Akabari. Ki-bert: Infusing knowledge context for better language and domain understanding. *arXiv preprint arXiv:2104.08145*, 2021.

[20] Zhihao Fan, Zhongyu Wei, Jingjing Chen, Siyuan Wang, Zejun Li, Jiarong Xu, and Xuanjing Huang. A unified continuous learning framework for multi-modal knowledge discovery and pre-training. *arXiv preprint arXiv:2206.05555*, 2022.

[21] Yin Fang, Haihong Yang, Xiang Zhuang, Xin Shao, Xiaohui Fan, and Huajun Chen. Knowledge-aware contrastive molecular graph learning. *arXiv preprint arXiv:2103.13047*, 2021.

[22] Yin Fang, Qiang Zhang, Haihong Yang, Xiang Zhuang, Shumin Deng, Wen Zhang, Ming Qin, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Molecular contrastive learning with chemical element knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3968–3976, 2022.

[23] Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. Entities as experts: Sparse memory access with entity supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4937–4951, Online, November 2020. Association for Computational Linguistics.

[24] Hongchang Gao and Heng Huang. Deep attributed network embedding. In *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI))*, 2018.

[25] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[26] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. DeCLUTR: Deep contrastive learning for unsupervised textual representations. 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 879–895, Online, August 2021. Association for Computational Linguistics.

[27] Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. In *International Conference on Learning Representations*, 2021.

[28] Yu Guo, Zhengyi Ma, Jiaxin Mao, Hongjin Qian, Xinyu Zhang, Hao Jiang, Zhao Cao, and Zhicheng Dou. Webformer: Pre-training with web pages for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1502–1512, 2022.

[29] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online, July 2020. Association for Computational Linguistics.

[30] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[31] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality Reduction by Learning an Invariant Mapping. In *CVPR*, 2006.

[32] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

[33] Bin He, Xin Jiang, Jinghui Xiao, and Qun Liu. Kgplm: Knowledge-guided language model pre-training via generative and discriminative learning. *arXiv preprint arXiv:2012.03551*, 2020.

[34] Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online, November 2020. Association for Computational Linguistics.

[35] Jie Hou, Badri Adhikari, and Jianlin Cheng. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, (8), 2018.

[36] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *arXiv: 2005.00687*, 2020.

[37] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for Pre-training Graph Neural Networks. In *ICLR*, 2020.

[38] Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. WhiteningBERT: An easy unsupervised sentence embedding approach. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 238–244, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[39] Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. In *NeurIPS*, 2020.

[40] Xiaoze Jiang, Yaobo Liang, Weizhu Chen, and Nan Duan. Xlm-k: Improving cross-lingual language model pre-training with multilingual knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10840–10848, 2022.

[41] Xunqiang Jiang, Yuanfu Lu, Yuan Fang, and Chuan Shi. Contrastive pre-training of gnns on heterogeneous graphs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 803–812, New York, NY, USA, 2021. Association for Computing Machinery.

[42] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[43] David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. Measuring the Evolution of a Scientific Field through Citation Frames. *Transactions of the ACL*, 6, 2018.

[44] Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. SentiLARE: Sentiment-aware language representation learning with linguistic knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6975–6988, Online, November 2020. Association for Computational Linguistics.

[45] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020.

[46] Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. Self-guided contrastive learning for BERT sentence representations. 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 2528–2540, Online, August 2021. Association for Computational Linguistics.

[47] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR, 18–24 Jul 2021.

[48] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[49] Michael Schantz Klausen, Martin Closter Jespersen, Henrik Nielsen, Kamilla Kjaergaard Jensen, Vanessa Isabell Jurtz, Casper Kaae Sønderby, Morten Otto Alexander Sommer, Ole Winther, Morten Nielsen, Bent Petersen, and Paolo Marcatili. NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins*, (6), 2019.

[50] Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. A mutual information maximization perspective of language representation learning. In *International Conference on Learning Representations*, 2020.

[51] Adhiguna Kuncoro, Lingpeng Kong, Daniel Fried, Dani Yogatama, Laura Rimell, Chris Dyer, and Phil Blunsom. Syntactic structure distillation pretraining for bidirectional encoders. *Transactions of the Association for Computational Linguistics*, 8:776–794, 2020.

[52] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *ICLR*, 2019.

[53] Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. The inductive bias of in-context learning: Rethinking pretraining example design. In *International Conference on Learning Representations*, 2022.

[54] Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. Pre-training via paraphrasing. *Advances in Neural Information Processing Systems*, 33:18470–18481, 2020.

[55] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[56] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online, November 2020. Association for Computational Linguistics.

[57] Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. StructuralLM: Structural pre-training for form understanding. 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 6309–6318, Online, August 2021. Association for Computational Linguistics.

[58] Da Li, Ming Yi, and Yukai He. LP-BERT: multi-task pre-training knowledge graph BERT for link prediction. *CoRR*, abs/2201.04843, 2022.

[59] Michelle M Li, Kexin Huang, and Marinka Zitnik. Representation learning for networks in biology and medicine: Advancements, challenges, and opportunities. *arXiv:2104.04883*, 2021.

[60] Ye Li, Chaofeng Sha, Xin Huang, and Yanchun Zhang. Community detection in attributed graphs: An embedding approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[61] Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238, Online, June 2021. Association for Computational Linguistics.

[62] Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908, 2020.

[63] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-Task Deep Neural Networks for Natural Language Understanding. In *ACL*, 2019.

[64] Ye Liu, Yao Wan, Lifang He, Hao Peng, and S Yu Philip. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6418–6425, 2021.

[65] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*, 2019. arXiv: 1907.11692.

[66] Yinquan Lu, Haonan Lu, Guirong Fu, and Qun Liu. Kelm: Knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs. *arXiv preprint arXiv:2109.04223*, 2021.

[67] Fuli Luo, Pengcheng Yang, Shicheng Li, Xuancheng Ren, and Xu Sun. Capt: contrastive pre-training for learning denoised sequence representations. *arXiv preprint arXiv:2010.06351*, 2020.

[68] Zhengyi Ma, Zhicheng Dou, Wei Xu, Xinyu Zhang, Hao Jiang, Zhao Cao, and Ji-Rong Wen. *Pre-Training for Ad-Hoc Retrieval: Hyperlink is Also You Need*, page 1212–1221. Association for Computing Machinery, New York, NY, USA, 2021.

[69] Matthew McDermott, Brendan Yap, Harry Hsu, Di Jin, and Peter Szolovits. Adversarial contrastive pre-training for protein sequences. *arXiv preprint arXiv:2102.00466*, 2021.

[70] Matthew B. A. McDermott, Bret Nestor, Evan Kim, Wancong Zhang, Anna Goldenberg, Peter Szolovits, and Marzyeh Ghassemi. A Comprehensive Evaluation of Multi-task Learning and Multi-task Pre-training on EHR Time-series Data. *arXiv: 2007.10185*, 2020.

[71] Yu Meng, Chenyan Xiong, Payal Bajaj, saurabh tiwary, Paul Bennett, Jiawei Han, and XIA SONG. Coco-lm: Correcting and contrasting text sequences for language model pretraining. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 23102–23114. Curran Associates, Inc., 2021.

[72] Zaiqiao Meng, Fangyu Liu, Thomas Hikaru Clark, Ehsan Shareghi, and Nigel Collier. Mixture-of-partitions: Infusing large biomedical knowledge graphs into bert. *arXiv preprint arXiv:2109.04810*, 2021.

[73] Seonwoo Min, Seunghyun Park, Siwon Kim, Hyun-Soo Choi, and Sungroh Yoon. Pre-Training of Deep Bidirectional Protein Sequence Representations with Structural Information. *arXiv: 1912.05625*, 2020.

[74] Youssef Mroueh, Tomaso Poggio, Lorenzo Rosasco, and Jean-Jacques E Slotine. Multiclass learning with simplex coding. In *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 2*, pages 2789–2797, 2012.

[75] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[76] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[77] Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, 2019.

[78] Nina Poerner, Ulli Waltinger, and Hinrich Schütze. E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online, November 2020. Association for Computational Linguistics.

[79] Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. ERICA: Improving entity and relation understanding for pre-trained language models via contrastive learning. 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 3350–3363, Online, August 2021. Association for Computational Linguistics.

[80] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1150–1160, New York, NY, USA, 2020. Association for Computing Machinery.

[81] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[82] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[83] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[84] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating Protein Transfer Learning with TAPE. In *NeurIPS*. Curran Associates, Inc., 2019.

[85] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8844–8856. PMLR, 18–24 Jul 2021.

[86] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[87] Danilo Neves Ribeiro and Kenneth Forbus. Combining analogy with language models for knowledge extraction. In *3rd Conference on Automated Knowledge Base Construction*, 2021.

[88] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[89] Gabriel J. Rocklin, Tamuka M. Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K. Mulligan, Aaron Chevalier, Cheryl H. Arrowsmith, and David Baker. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347), 2017.

[90] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.

[91] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021.

[92] Sarkisyan et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603), 2016.

[93] Nikunj Saunshi, Sadhika Malladi, and Sanjeev Arora. A mathematical exploration of why language models help solve downstream tasks. In *International Conference on Learning Representations*, 2021.

[94] Blake Shaw, Bert Huang, and Tony Jebara. Learning a Distance Metric from a Network. In *NeurIPS*, 2011.

[95] Blake Shaw and Tony Jebara. Structure preserving embedding. In *ICML*, 2009.

[96] Tao Shen, Yi Mao, Pengcheng He, Guodong Long, Adam Trischler, and Weizhu Chen. Exploiting structured knowledge in text via graph-guided representation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8980–8994, Online, November 2020. Association for Computational Linguistics.

[97] Bosheng Song, Zimeng Li, Xuan Lin, Jianmin Wang, Tian Wang, and Xiangzheng Fu. Pre-training model for biological sequence data. *Briefings in functional genomics*, 20(3):181–195, 2021.

[98] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

[99] Yusheng Su, Xu Han, Zhengyan Zhang, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models. *AI Open*, 2:127–134, 2021.

[100] Kailai Sun, Zuchao Li, and Hai Zhao. Multilingual pre-training with universal dependency learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8444–8456. Curran Associates, Inc., 2021.

[101] Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. CoLAKE: Contextualized language and knowledge embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3660–3670, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.

[102] Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*, 2021.

[103] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv:1904.09223 [cs]*, 2019. arXiv: 1904.09223.

[104] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05), 2020. Number: 05.

[105] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*, 2018.

[106] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[107] Jean-Philipe Vert and Yoshihiro Yamanishi. Supervised graph inference. In *NeurIPS*, 2004.

[108] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Darrin Eide, Yuxiao Dong, Junjie Qian, Anshul Kanakia, Alvin Chen, and Richard Rogahn. A Review of Microsoft Academic Services for Science of Science Studies. *Frontiers in Big Data*, 2019.

[109] Liang Wang, Wei Zhao, Zhuoyu Wei, and Jingming Liu. SimKGC: Simple contrastive knowledge graph completion with pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4281–4294, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[110] Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020.

[111] Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. Structbert: Incorporating language structures into pre-training for deep language understanding. In *International Conference on Learning Representations*, 2020.

[112] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 03 2021.

[113] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-Similarity Loss With General Pair Weighting for Deep Metric Learning. In *CVPR*, 2019.

[114] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[115] Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. CLEVE: Contrastive Pre-training for Event Extraction. 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 6283–6297, Online, August 2021. Association for Computational Linguistics.

[116] Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*, 2020.

[117] Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *International Conference on Learning Representations*, 2020.

[118] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. Luke: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, 2020.

[119] Ruiqing Yan, Lanchang Sun, Fang Wang, and Xiaoming Zhang. A general method for transferring explicit knowledge into language model pretraining. *Security and Communication Networks*, 2021, 2021.

[120] Xiaoyan Yan, Fanghong Jian, and Bo Sun. Sakg-bert: Enabling language representation with knowledge graphs for chinese sentiment analysis. *IEEE Access*, 9:101695–101701, 2021.

[121] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. ConSERT: A contrastive framework for self-supervised sentence representation transfer. 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 5065–5075, Online, August 2021. Association for Computational Linguistics.

[122] Junhan Yang, Zheng Liu, Shitao Xiao, Chaozhuo Li, Guangzhong Sun, and Xing Xie. Graphformers: Gnn-nested language models for linked text representation. *arXiv preprint arXiv:2105.02605*, 2021.

[123] Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019.

[124] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. LinkBERT: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8003–8016, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[125] Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. VIME: Extending the Success of Self- and Semi-supervised Learning to Tabular Domain. In *NeurIPS*, 2020.

[126] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12121–12132. PMLR, 18–24 Jul 2021.

[127] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5812–5823. Curran Associates, Inc., 2020.

[128] Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. Jaket: Joint pre-training of knowledge graph and language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11630–11638, 2022.

[129] Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. Dict-BERT: Enhancing language model pre-training with dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1907–1918, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[130] Zheng Yuan, Yijia Liu, Chuanqi Tan, Songfang Huang, and Fei Huang. Improving biomedical pretrained language models with knowledge. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 180–190, Online, June 2021. Association for Computational Linguistics.

[131] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graphsaint: Graph sampling based inductive learning method. In *International Conference on Learning Representations*, 2020.

[132] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Homophily, structure, and content augmented network representation learning. In *ICDM*, 2016.

[133] Ningyu Zhang, Shumin Deng, Xu Cheng, Xi Chen, Yichi Zhang, Wei Zhang, Huajun Chen, and Hangzhou Innovation Center. Drop redundant, shrink irrelevant: Selective knowledge injection for language pretraining. In *In IJCAI*, 2021.

[134] Shichang Zhang, Ziniu Hu, Arjun Subramonian, and Yizhou Sun. Motif-driven contrastive learning of graph representations. *arXiv preprint arXiv:2012.12533*, 2020.

[135] Taolin Zhang, Zerui Cai, Chengyu Wang, Minghui Qiu, Bite Yang, and Xiaofeng He. Smedbert: A knowledge-enhanced pre-trained language model with structured semantics for medical text mining. *arXiv preprint arXiv:2108.08983*, 2021.

[136] Xiao-Chen Zhang, Cheng-Kun Wu, Zhi-Jiang Yang, Zhen-Xing Wu, Jia-Cai Yi, Chang-Yu Hsieh, Ting-Jun Hou, and Dong-Sheng Cao. Mg-bert: leveraging unsupervised atomic representation learning for molecular property prediction. *Briefings in Bioinformatics*, 2021.

[137] Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. GreaseLM: Graph REASoning enhanced language models. In *International Conference on Learning Representations*, 2022.

[138] Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1601–1610, Online, November 2020. Association for Computational Linguistics.

[139] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the ACL*. ACL, 2019.

[140] Zhuosheng Zhang and Hai Zhao. Structural pre-training for dialogue comprehension. 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 5134–5145, Online, August 2021. Association for Computational Linguistics.

[141] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

[142] Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, and Xiang Ren. Pre-training text-to-text transformers for concept-centric common sense. In *International Conference on Learning Representations*, 2021.

[143] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Generalizing graph neural networks beyond homophily. In *NeurIPS*, 2020.

[144] Marinka Zitnik, Rok Sosič, Marcus W. Feldman, and Jure Leskovec. Evolution of resilience in protein interactomes across the tree of life. *Proceedings of the National Academy of Sciences*, 116(10), 2019.

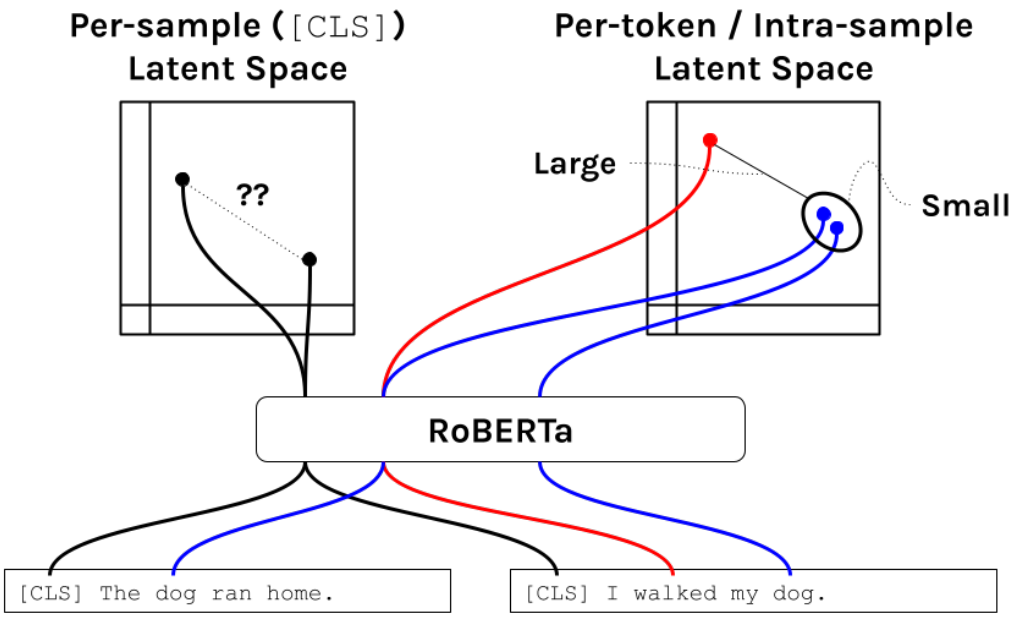

Figure 3: **Per-sample vs. Per-token Latent Space** Language model pre-training methods produce both per-sample and per-token latent spaces. Traditional language modelling objectives (illustrated here via the RoBERTa [65] model, which uses only a masked language model loss during pre-training) only constrain the per-token latent space.

A Further Details on Why LMPT is Less Effective in non-NLP Domains

A.1 Per-token vs. Per-sample Latent Space: Definition of $\mathcal{Z}$

Let $f_{\boldsymbol{\theta}}$ be a pre-training (PT) model trained over a dataset $\boldsymbol{X} \in \mathcal{X}^{N_{\text{PT}}}$. Furthermore, let us assume that samples $\boldsymbol{x} \in \mathcal{X}$ are composed of smaller units (*e.g.* tokens, sequence time-points, nodes in a network, etc.). Let us denote this by saying that $\boldsymbol{x} = w_1, w_2, \ldots, w_{n_{\boldsymbol{x}}}$. Finally, as is true in natural language processing (NLP) and NLP-derived settings, we assume that $f_{\boldsymbol{\theta}}$ can be seen to produce output embeddings for both the entire sample $\boldsymbol{x}$—which we will denote by $f_{\boldsymbol{\theta}}(\boldsymbol{x})$—and for the internal tokens individually—which will denote by $f_{\boldsymbol{\theta}}(w_j|\boldsymbol{x})$. For example, in the BERT model [17], $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ will be given by the output embedding of the `[CLS]` token of $\boldsymbol{x}$ and $f_{\boldsymbol{\theta}}(w_j|\boldsymbol{x})$ will be given by the output embedding of the j-th token in $\boldsymbol{x}$.

We can then formally define the per-sample latent space, $\mathcal{Z}^{(\text{S})}$ (which we will also refer to as $\mathcal{Z}$ without the superscript), and the per-token (aka intra-sample) latent space $\mathcal{Z}^{(\text{T})}$ (Definitions 1 & 2, and Figure 3).

Definition 1. Per-Sample Latent Space We define the *per-sample latent space* induced by $f_{\boldsymbol{\theta}}$ as $\mathcal{Z}^{(\text{S})} = \{f_{\boldsymbol{\theta}}(\boldsymbol{x})|\boldsymbol{x} \in \mathcal{X}\}$. We will also use $\mathcal{Z}$ with no superscript to refer to this space.

Definition 2. Per-token/Intra-sample Latent Space We define the *per-token latent space* (also known as the *intra-sample latent space*) induced by $f_{\boldsymbol{\theta}}$ as $\mathcal{Z}^{(\text{T})} = \{f_{\boldsymbol{\theta}}(w_j|\boldsymbol{x})|\boldsymbol{w}_j \in \boldsymbol{x}, \boldsymbol{x} \in \mathcal{X}\}$.

Both of these spaces are very different and are useful in different contexts; for a task like named entity recognition, where the unit of classification is a single or short span of tokens, analyzing the per-token latent space will be more informative, whereas for a task like sentiment analysis, where the unit of classification is an entire sample (sentence), the per-sample latent space would be preferred [17]. Furthermore, another key difference between these spaces is that the traditional PT language model objective only induces significant constraints on the geometry of the per-token latent space and does not impact the per-sample latent space at all. This illustrates a gap in the capabilities of PT methods. In our work, we are concerned with precisely this gap and focus our attention on $\mathcal{Z}$ (*i.e.* $\mathcal{Z}^{(\text{S})}$). We focus our attention on the per-sample latent space for 3 reasons:

1. There has been significantly more research on how to regularize the per-token latent space than the per-sample latent space, as we show in our extensive review (Table 4).
2. In many domains outside of NLP, the per-sample latent space is often of much greater interest than the intra-sample latent space. For example, in modelling protein sequences [84], drug structures [37], or electronic health record time series [70], per-sample tasks are of much greater interest than intra-sample tasks.
3. Even within NLP, modern methods struggle much more with representing whole passages of text rather than short, isolated spans. This is evidenced by the battery of work examining sentence representations atop pre-trained language models [56, 25].

A.2 Why is NLP Different than Other Domains?

In this work, we have implicitly argued that because a PT objective like masked language modelling (MLM) will not necessarily directly enrich the per-sample latent space $\mathcal{Z}^{(\mathrm{S})}$, it may yield models less well suited to downstream per-sample tasks than other approaches. One seeming contradiction to this is that methods in NLP like RoBERTa [65] (for which MLM is the only PT objective) succeed across both per-token and per-sample tasks.

In fact, this observation does not contradict our hypothesis but reflects a unique advantage of the natural language modality that does not apply in other domains. In particular, in the NLP domain (and not in other domains), we can leverage the flexibility of the language to sidestep any deficit in $\mathcal{Z}^{(\mathrm{S})}$ by re-framing per-sample tasks as per-token, language modelling tasks. Significant literature exists documenting this phenomenon through the lenses of prompting, cloze-filling models, text-to-text transformers, and theoretical analyses [5, 93, 91, 83, 18]. For example, [93] examines the efficacy of pre-trained language models on sentiment analysis explicitly and show that the language modelling component alone can be used in a per-token manner to indirectly solve a review sentiment analysis task by judging the likelihood of following the review with a ":)" emoji vs. a ":(" emoji. In this way, they shift the *per-sample* task of sentiment analysis to a *per-token* task via the (inserted) emoji.

However, language model pre-training has also inspired many derived methods to be used in other non-NLP domains. For example, in modelling graphs, [37] has examined vertex or edge-masking strategies reminiscent of MLM, with vertices and edges analogous to tokens and entire graphs whole samples; in modelling time series data, [70] has examined masked imputation models, with timepoints analogous to tokens and whole time series to samples; and in modelling protein sequences, [73] has used masked language modelling directly, with individual amino acids representing tokens and entire proteins representing samples. *In all three of these domains, we cannot re-frame per-sample tasks as "per-token" tasks as we can in NLP, and accordingly, the problem of insufficient per-sample latent space regularization will likely be much more severe in these domains. Accordingly, existing work, including the three works referenced above, all find that augmenting the language model pre-training task with additional, per-sample level supervised tasks can be beneficial, or even absolutely essential, to improving performance [37, 125, 70, 73].*

B Our Review of NLP-derived PT Methods

B.1 Defining Explicit and Deep Structural Constraints

Central to our hypothesis is the claim that most NLP-derived PT methods today do not impose explicit, deep constraints on the (per-sample) latent space geometry of $\mathcal{Z}$. To justify this claim, we define "explicit" and "deep" structural constraints (Definitions 3-4).

Definition 3. Explicit vs. Implicit Structural Constraints:

A PT objective $\mathcal{L}_{\mathrm{PT}}$ imposes a structural constraint that is *explicit* (vs. implicit) to the degree that it (as $f_{\boldsymbol{\theta}}$ approaches optimality) permits us to reason directly about the relationship (in particular, the distance) between any two samples $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ in the latent space $\mathcal{Z}$.

Definition 4. Deep vs. Shallow Structural Constraints:

A PT objective $\mathcal{L}_{\text{PT}}$ imposes a structural constraint that is *deep* (vs. shallow) on the basis of how much information (*e.g.*, how many dimensions) would be required to fully satisfy the constraint.

For example, consider a classification PT loss according to labels $y_i \in \mathcal{Y}$ and a logit layer which maps $\boldsymbol{z}_i \mapsto \tilde{y_i}$. This method produces an *explicit* structural constraint because near optimality, we can infer that the relative (cosine) distance between two samples $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ is small if and only if $y_i = y_j$. However, this constraint is also *shallow*, because to fully satisfy this constraint, we need only embed each class $c \in \mathcal{Y}$ with a unique position $\boldsymbol{p}_c \in \mathcal{Z}$, then compress all samples $\boldsymbol{z}_i$ near their class prototype $\boldsymbol{p}_{y_i}$, which can be accomplished in a very low dimensional space $\mathcal{Z}$. In particular, these class prototypes and embeddings can be arranged to optimality in $\mathbb{R}^2$ simply by distributing all class prototypes uniformly along the unit circle, then pushing all sample embeddings towards the prototypes along the unit circle.

In contrast, consider a contrastive method that asserts that $\boldsymbol{z}_i = f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ should be close to $\boldsymbol{z}'_i = f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}_i})$, under some noising/augmentation procedure $\boldsymbol{x}_i \mapsto \tilde{\boldsymbol{x}_i}$, but simultaneously far from other samples $\boldsymbol{z}_j$. While this method constrains the latent space to be smooth with respect to the noising process, it offers only an *implicit* constraint on $\mathcal{Z}$ as it is generally not possible to infer how the distance between distinct samples $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ is constrained. However, it imposes a *deeper* constraint than does the classification objective because the implicit connections between samples induced by the noising procedure reflect relationships that can not necessarily be captured in a low-dimensional space (dependent on dataset size and density).

B.2 Pre-training Review Methodology

Papers were selected via a manual search of the natural language processing (NLP) and NLP-derived pre-training methods (*i.e.*, methods focused primarily on other domains or on multi-modal domains were excluded) via Google Scholar as well as by crawling through references of papers already included. Citation counts for each work were obtained via Google Scholar on August 2nd, 2022. Publication date (used to calculate citations per month since publication date) was computed as the earlier of either (1) the paper's venue-specific date of publicatoin or (2) the first submission date to the arXiv or BioRxiv platform, as referenced via an exact title match. A manual review was done to classify how pre-training methods constrain latent space geometry and assign subjective, numerical "shallow-deep" and "explicit-implicit" axes scores. In total, over 90 methods were examined, of which 71 were suitable for inclusion in numerical review results (Figure 4 and Table 4). All methods considered are summarized and categorized (and reasons for exclusions are given) in Appendix G.

B.3 Review Findings

To show that existing methods largely do not provide means to impose structural constraints that are simultaneously deep and explicit, we survey over 90 existing PT methods on the basis of how their objective functions constrain the $\mathcal{Z}$ (Figure 4, Appendix Sections B.2, B.4, G). Throughout all examined methods, we find that *deep, explicit structural constraints are almost never employed*. Instead, most methods either (1) impose no per-sample PT objectives at all ([5, 65, 84]), (2) use explicit, but shallow, supervised PT objectives (*e.g.*, BERT's "Next-sentence Prediction" (NSP) objective, ALBERT's "Sentence-order Prediction" (SOP) objective, or various multi-task objectives [17, 52, 63]), or (3) use implicit, but deep, un- or self-supervised contrastive PT objectives (*e.g.*, contrastive sentence embedding losses [26, 50, 116, 87, 71]).

Across all surveyed methods, we find that only four methods impose simultaneously explicit and deep constraints: KEPLER [112], CK-GNN [21], XLM-K [40], and WebFormer [28]. All four can be described as some form of per-sample graph alignment, in which the output embeddings of pairs of samples $\boldsymbol{z}_i = f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ and $\boldsymbol{z}_j = f_{\boldsymbol{\theta}}(\boldsymbol{x}_j)$ are constrained based on whether or not an edge exists between nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in some pre-training graph G_{PT}. This form of constraint is explicit, as the graph G_{PT} contains concrete relationships that will be induced in the output latent space, but also deep, as the geometry of the graph G_{PT} can be arbitrarily complex.

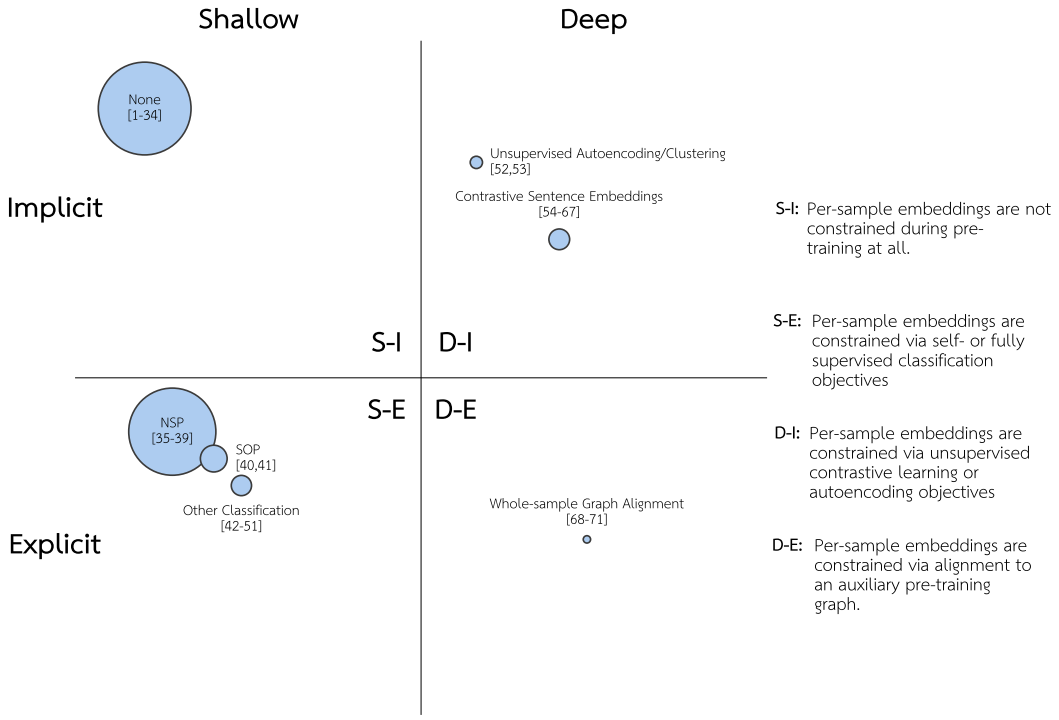

Figure 4: **Existing Pre-training (PT) Methods:** A summary of 71 existing natural language processing (NLP) and NLP-derived PT methods, categorized into clusters based on how they impose structural constraints over the PT (per-sample) latent space. Clusters are arranged on axes via manual judgements on whether the imposed constraint is *shallow* vs. *deep* and *implici* vs. *explicit*. Clusters are sized such that the area corresponds to the number of citations methods included in that cluster have received on average per month since first publication, according to Google Scholar's citation count. "None" captures models that leverage no pre-training loss over the per-sample embedding. "NSP" refers to "Next-sentence Prediction," the per-sample PT task introduced in BERT [17]. "SOP" refers to "Sentence-order Prediction," the per-sample PT task introduced in ALBERT [52]. Note that over 90 studies in total were considered in our review, but only 71 met the inclusion criteria to be included in this figure. These methods are described in more detail in Appendix Sections B.2, B.4, G and Table 4.

However, all these methods have major limitations. In KEPLER and XLM-K, the per-sample embeddings are only constrained to a restricted set of samples corresponding to entity descriptions from a knowledge graph. As such, there are no constraints implied on the general domain of free-text samples in $\mathcal{X}$ alone [112, 40]. In CK-GNN, the graph connectivity is derived from a cluster-restricted 1-nearest-neighbor graph in an alternate modality's distance space, which may offer a limited higher-order structure, and unlike the NLP approaches, this method has no intra-sample (*e.g.* per-token) pre-training task [21]. Finally, in WebFormer, the graph used is inferred from the structure of the HyperText Markup Language (HTML) underlying web-pages, and relationships are only constrained at the per-sample level for limited structural relationships within the HTML. Further, WebFormer is a specialized model specifically for processing web content (text and HTML elements), so their approach can't be directly generalized to other domains [28]. Moreover, none of these methods offer general insights for how to realize this style of deep, explicit per-sample constraints in other contexts, nor do they provide any theoretical guidance on how these constraints relate to performance for fine-tuning tasks [112, 21, 40, 28].

Overall, our review of pre-training methods establishes unequivocally that pre-training methods capable of providing explicit, deep structural constraints are significantly under-explored. Across all the methods we reviewed, only four methods leverage constraints are explicit and deep, all of which have significant limitations, and there is no general consensus on how to constrain $\mathcal{Z}$ explicitly and deeply. These findings motivate our new framework, which offers insight into how to realize deep, explicit structural constraints in pre-training models across diverse contexts and provides theoretical guidance on how structural constraints relate to fine-tuning performance.

B.4 Further Analysis of Reviewed Methods

This work has extensively examined how existing pre-training methods constrain the *per-sample* latent space. However, it is also worth examining how these methods constrain the per-token latent space to demonstrate the extent to which per-sample objectives are under-explored in current pre-training research. To that end, we break down all of the studies included in our review not only by how they constrain their per-sample latent spaces but also by how they constrain their per-token latent spaces (Table 4). These groupings are also done at a greater granularity than the previously examined categories to offer more insight into which methods use which techniques. We see that not only are there more types of per-token latent space constraints leveraged (10 vs. 7), but also methods consistently leverage a much greater diversity of per-token constraints vs. per-sample constraints (1.45 per-token constraints per method vs. 0.58 per-sample constraints, on average). We can further see from Figure 4 that the citation volume for works in this space is also heavily concentrated around methods that first employ no per-sample PT objective, followed by methods that only impose shallow, explicit methods, which further establishes this research gap.

Method	Masked, discriminative, or standard language modelling	Template/prompt-style multi-task language model training	Concatenate related sentences together	Named entity masking	Relation masking	Per-token knowledge graph alignment	Named entity recognition and linking	(Unconstrained) attention over a KG	Joint token and entity embeddings	Syntatic Knowledge Distillation	Single-task classification	Multi-task classification	Whole-sample graph alignment	Per-sample augmentation-based contrastive alignment	Multi-lingual cross-sample contrastive alignment	Unsupervised clustering	Contextual autoencoding
	Per-token										**Per-sample**						
[76] ELMO	✓																
[5] GPT-3	✓																
[83] T5		✓															
[65] RoBERTa	✓																
[81] GPT-1	✓																
[82] GPT-2	✓																
[55] BART	✓																
[15] Unsupervised Cross Lingual	✓																
[12] ELECTRA	✓																
[42] SpanBERT	✓																
[18] UniLM	✓																
[29] DAPT	✓																
[103] ERNIE (Sun et. al.)	✓			✓													
[77] KnowBERT	✓						✓	✓	✓								
[84] TAPE	✓																
[118] LUKE	✓			✓													
[90] T0pp		✓															
[117] Pretrained Encyclopedia	✓			✓													
[85] MSA	✓		✓														
[101] COLAKE	✓			✓	✓												
[34] BERTMK	✓								✓								
[79] ERICA	✓					✓											
[128] JAKET	✓								✓								
[142] CALM		✓															
[130] KeBioLM	✓						✓	✓	✓								
[136] MG-BERT (Molecules)	✓																
[6] CDLM	✓		✓														
[33] KgPLM	✓			✓													
[53] kNN PT	✓		✓														
[58] LP-BERT	✓			✓	✓												
[3] MG-BERT (NLP)	✓								✓								
[100] UD-PrLM	✓					✓											
[88] ESM-1B	✓																
[2] UniRep	✓																
[17] BERT	✓										✓						
[139] ERNIE (Zhang et. al.)	✓			✓					✓		✓						
[99] CokeBERT	✓			✓					✓		✓						
[140] SPIDER	✓					✓					✓						
[51] Syntatic-Distilled BERT	✓									✓	✓						
[52] ALBERT	✓										✓						
[135] SMedBERT	✓					✓			✓		✓						
[63] MT-DNN	✓											✓					
[37] Graph-PT	✓											✓					
[44] SentiLARE	✓										✓						
[73] PLUS	✓						✓				✓						
[70] EHR-PT	✓											✓					
[104] ERNIE 2.0 (Sun et. al.)	✓			✓								✓					
[102] ERNIE 3.0 (Sun et. al.)	✓			✓	✓							✓					
[129] Dict-BERT	✓		✓				✓				✓						
[124] LinkBERT	✓										✓						
[111] StructBERT	✓										✓						
[54] MARGE																✓	✓
[30] REALM	✓		✓	✓												✓	
[127] GraphCL														✓			
[80] GCC														✓			
[26] DeCLUTR	✓													✓			
[116] CLEAR	✓													✓			
[126] JOAO														✓			
[71] COCO-LM	✓													✓			
[50] InfoWord	✓													✓			
[134] MICRO-Graph														✓		✓	
[8] STS-CT	✓													✓			
[67] CAPT														✓			
[141] GearNet	✓					✓								✓			
[11] InfoXLM	✓														✓		
[96] GLM	✓			✓										✓			
[22] KCL														✓			
[112] KEPLER	✓												✓				
[21] CK-GNN													✓				
[40] XLM-K	✓												✓				
[28] Webformer	✓											✓	✓	✓			

Table 4: Existing Pre-training (PT) Methods: A subset of existing PT methods, broken down by how they constrain per-token and per-sample latent space geometries.

C Further Details on SIPT

C.1 Constraints on $\mathcal{L}_{\text{SI}}$ in our Framework

Formally, for $\mathcal{L}_{\text{SI}}$ to be valid, then there must exist a distance function $d : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$, radius $r \in \mathbb{R}$, and loss value $\ell^* \in \mathbb{R}$ such that at any solution $\boldsymbol{\theta}^*$ for which $\mathcal{L}_{\text{SI}}(\boldsymbol{\theta}^*) < \ell^*$, the learned embeddings $\boldsymbol{z}_i = f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i)$ must recover the graph G_{PT} under a radius nearest neighbor connectivity algorithm via distance function d and radius r—*i.e.*, it must be the case that $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ if and only if $d(f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i), f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_j)) < r$. Furthermore, for the particular graph G_{PT} and latent space $\mathcal{Z}$, the set of $\boldsymbol{\theta}^*$ such that $\mathcal{L}_{\text{SI}}(\boldsymbol{\theta}^*) < \ell^*$ must be non-empty (*i.e.* such a solution must exist).

C.2 Realizing Existing Methods in our Framework

Let $\boldsymbol{X} \in \mathcal{X}^{N_{\text{PT}}}$ be the pre-training dataset throughout this section. In cases where we have some auxiliary information (*e.g.*, supervised, per-sample, pre-training labels), they will be denoted by $\boldsymbol{Y} \in \mathcal{Y}^{N_{\text{PT}}}$.

Methods with no per-sample objectives

Naturally, we can realize any method that only employs a per-token pre-training objective within our framework simply by setting $\lambda_{\text{SI}} = 0$. This realization is trivial and offers no insight into the suitability of these pre-training methods for downstream per-sample tasks.

Methods with a supervised, single-task per-sample objective (*e.g.*, BERT [17])

A simple, single-task, per-sample, classification pre-training objective induces a geometric constraint in the output latent space on the basis of the inner product "distance" between samples of the same vs. different class labels. We can use this observation to realize a reduction from a valid SIPT objective to the original classification objective. In particular, we can introduce a graph $G = (\{\boldsymbol{x}_i \in \boldsymbol{X}\}, \{(\boldsymbol{x}_i, \boldsymbol{x}_j) | y_i = y_j\})$ which consists of cliques corresponding to each unique label $c \in \mathcal{Y}$. Then, leveraging any structure-preserving metric learning loss with a cosine distance objective will, at optimality, recover a solution that also satisfies the original classification objective, where we use centroids of the induced clique embeddings to represent class embeddings.

Methods with a supervised, multi-task per-sample objective (*e.g.*, MT-DNN [63])

A slightly more complicated case is when methods employ a multi-task, per-sample classification objective. In this case, there are two ways to realize this task within the SIPT framework. First, we can simply transform the multi-task objective into a single-task objective by constructing a new label-space consisting of the Cartesian product of all label spaces for each task individually. This will greatly increase the number of "labels" in the task, but then the problem can be realized via a graph of disconnected cliques much like in the single-task setting.

However, there is another manner in which we can realize this objective in the SIPT framework; In particular, suppose our collection of tasks consists of k label spaces: $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_k$. Then, we can construct a graph $G = (V, E)$ such that:

1. the vertices consist of all pre-training samples $\boldsymbol{x}_i$ as well as auxiliary nodes corresponding to each label $c_h^{(j)} \in \mathcal{Y}_h$ across each task: $V = \{\boldsymbol{x}_i \in \boldsymbol{X}\} \cup \mathcal{Y}_1 \cup \cdots \cup \mathcal{Y}_k$
2. the edges contain links between each sample $\boldsymbol{x}_i$ and label $y_h^{(i)}$ across all tasks $1 \leq h \leq k$: $E = \{(\boldsymbol{x}_i, c_h^{(j)}) | y_h^{(i)} = c_h^{(j)}\}$.

Then, we can see that if we solve the SIPT problem under a structure-preserving metric learning loss, we will naturally have produced embeddings for each $\boldsymbol{x}_i$ which are close (in inner-product distance space) to the class embeddings corresponding to their labels for each task, while they are also far from other, non-matching class embeddings, as desired. This second approach is more useful to us

in considering the ramifications of this style of constraint because it enables us to make more rigid theoretical guarantees via the SIPT theory.

Methods with a based contrastive per-sample objective (*e.g.*, GraphCL [127])

It is challenging to realize contrastive learning approaches within the SIPT framework, but it is still possible. Here, we highlight two distinct types of contrastive learning approaches we can capture within SIPT: a noising/augmentation-based approach, in which sample embeddings are constrained to be similar to embeddings of noised versions of said samples; and a multi-modal (or multi-lingual) contrastive approach, in which there exists a 1:1 mapping between two different sub-modalities within $\boldsymbol{X}$ which is used to join those two modalities into a unified latent space (*e.g.* a model which constrains embeddings of English sentences to be close to embeddings of their french translations, but far from unrelated sentences).

To consider the augmentation/noising policy type first, let $h : \boldsymbol{x}_i \mapsto \tilde{\boldsymbol{x}}_i$ represent the noising transformation. Then, to build an analogous SIPT model to this model, we construct an augmented dataset consisting of all original data points alongside all possible transformed versions of the original data points under h: $\boldsymbol{X}' = \boldsymbol{X} \cup \left(\bigcup_{i=1}^{N_{\mathrm{PT}}} \mathrm{Im}\left(h|_{\boldsymbol{x}_i}\right)\right)$. Note that even in contexts where h is continuous (and thus has an infinite image), we can still construct this dataset in practice because training is only performed over a finite number of steps, meaning our augmented dataset $\boldsymbol{X}'$ need only be expanded to cover a finite number of augmentations. Then, the associated pre-training graph is simple; we simply use every sample in the augmented dataset $\boldsymbol{X}'$ as a vertex and connect any two samples if and only if one is a transformed version of the other. This forms a graph of many disconnected stars (one star for each original datapoint $\boldsymbol{x}_i$), and thus it does not directly enforce any particular geometry via our current theory. However, in cases where dataset size is sufficiently large, h sufficiently expressive, and data density sufficiently high, then the natural continuity of any neural network model will induce additional, auxiliary connections across these stars (if, for example, the noised versions of two distinct samples have a high probability of being very similar), which increases the depth of the geometric constraints enforced. Quantifying the exact parameters of these interactions, however, we leave to future work.

In the case of the multi-modal/multi-lingual contrastive alignment objective across k modalities, our setup is much simpler: we simply let G_{PT} be a k-partite graph whose samples consist of individual data points (across all modalities) and edges connect samples that compose a matching pair across modalities (*e.g.* edges link English sentences to their french translations). The extent to which this constrains the output geometry in practice, then, comes down to several questions: (1) Is the cross-modal alignment a one-to-one, one-to-many, or many-to-many alignment (which impacts the geometry of the resulting graph), (2) How large and dense is the dataset (which impacts the extent to which additional, indirect edges will be induced due to continuity in practice), and (3) How do other pre-training objectives constrain the individual modalities separately? In a case where this graph is one-to-one, and no other constraints are induced in each modality separately, this objective will offer only minimal constaints as the resulting graph will consistent of many disconnected 2-cliques.

Methods with a per-sample graph-alignment objective (*e.g.*, KEPLER [112])

Methods that explicitly align samples with a provided pre-training graph (KEPLER [112], CK-GNN [21], XLM-K [40], and WebFormer [28]) are naturally already realized within SIPT, so need no further commentary here.

C.3 Structure-inducing Losses Examined in this Study

Multi-similarity loss

The multi-similarity loss, parametrized by w_+, w_-, and t, is given below:

$$\mathcal{L}_{\mathrm{SI}} = \frac{1}{Nw_+} \log \left(1 + \sum_{(i,j)\in E} e^{-w_+(\langle f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), f_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle - t)} \right) + \frac{1}{Nw_-} \log \left(1 + \sum_{(i,j)\notin E} e^{w_-(\langle f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), f_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\rangle - t)} \right),$$

Contrastive loss

Our contrastive loss is modeled after [31]'s version. For this loss, we assume we are given the following mappings: 'pos', which maps $\boldsymbol{x}$ into a positive node (*i.e.*, linked to $\boldsymbol{x}$ in G_{PT}), and 'neg', which maps $\boldsymbol{x}$ into a negative node (*i.e.*, not linked to $\boldsymbol{x}$ in G_{PT}). The union of a seed minibatch B of points $\boldsymbol{X}_B$ and its images under 'pos' and 'neg' mappings form a full minibatch. This loss is specified by the positive and negative margin parameters μ_+ and μ_- as:

$$\mathcal{L}_{\mathrm{SI}}^{(\mathrm{CL})} = \frac{1}{N} \sum_{\boldsymbol{x}_i \in \boldsymbol{X}} \max(\mathcal{D}(\boldsymbol{x}_i, \mathrm{pos}(\boldsymbol{x}_i)) - \mu_+, 0) + \frac{1}{N} \sum_{\boldsymbol{x}_i \in \boldsymbol{X}} \max(\mu_- - \mathcal{D}(\boldsymbol{x}_i, \mathrm{neg}(\boldsymbol{x}_i)), 0).$$

C.4 Additional Choices within the SIPT Framework

In addition to a loss term, we can use negative sampling to improve efficiency. Using the full graph G_{PT}, which is not available in many contexts where negative sampling is employed, we can leverage the distance between samples calculated on G_{PT}, which provides a complementary source of information beyond embedding space distance alone. For example, one could use this to limit negative samples within the same connected component, but more complex strategies based on graph sampling (*e.g.* [131]) could also be used.

D Further Details and Proofs of SIPT Theory

D.1 Corollaries of Theorem 1

Corollary 1. *Let $\boldsymbol{X}_{\mathrm{PT}} \in \mathcal{X}^N$, be a PT dataset with corresponding labels $\boldsymbol{y} \in \mathcal{Y}_{\mathrm{PT}}^N$. Define $G_{\mathrm{PT}} = (\boldsymbol{X}_{\mathrm{PT}}, E)$ such that $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ if and only if $y_i = y_j$.*

Then, the local consistency for a given FT task $\boldsymbol{y}^{(FT)}$ over G_{PT} (and thus by Theorem 1, the nearest-neighbor accuracy for any optimized SIPT embedder) is upper bounded by the probability that a sample x_i's fine-tuning label $y_i^{(FT)}$ agrees with the majority class label for task $\boldsymbol{y}^{(FT)}$ over the clique consisting of all nodes with the same pre-training *label y_i as x_i.*

Corollary 2. *Let $\boldsymbol{X}_{\mathrm{PT}}$ be a PT dataset that can be realized over a valid manifold $\mathcal{M}$. Assume $\boldsymbol{X}_{\mathrm{PT}}$ is sampled with full support over $\mathcal{M}$. Let $G_{\mathrm{PT}}(\boldsymbol{X}_{\mathrm{PT}}, E)$ be an r-nearest-neighbor graph over $\mathcal{M}$ (*e.g.*, $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ if and only if the geodesic distance between the two points on $\mathcal{M}$ is less than r: $\mathcal{D}_{\mathcal{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) < r$). Let $y^{(FT)}$ be a FT classification task that is almost everywhere smooth on the manifold.*

Then, as PT dataset size (and thus the size of G_{PT}) tends to ∞, and r tends to zero, the local consistency of $y^{(FT)}$ over G_{PT} (and thus by Theorem 1 the nearest-neighbor accuracy of an SIPT embedder) will likewise tend to 1.

Informally, these corollaries establish that when a shallow structural constraint is used (*e.g.* a supervised classification objective), then the associated SIPT-equivalent model permits only minimal guarantees for FT performance, driven by the extent to which an FT task label is consistent within the classes under the supervised PT objective. In contrast, if a deep structural constraint is used, realized in Corollary 2 via G_{PT} being a nearest-neighbor graph over an arbitrary manifold $\mathcal{M}$, then a SIPT model permits a theoretical guarantee for FT performance that approaches unity as the pre-training dataset size grows for any FT task that is smooth over $\mathcal{M}$.

D.2 Proof of Theorem 1

We begin by defining the notion of "Local Consistency," which (informally) quantifies how "smooth" a given fine-tuning task label is over a graph G_{PT} (Definition 5). In addition, note that throughout all proofs, we will assume that the PT and FT datasets are iid, that FT tasks, though they may be unobserved over PT samples, are well defined over the entire PT and FT domain and thus true labels do exist (though they may be unknown) for PT samples, and that the sampling distribution of the PT/FT data has full support over the label-space of any considered task.

Definition 5 (Local Consistency)**.** Let $y : X \to \mathcal{Y}$ be a task over a domain X, and let $G = (V, E)$ be a graph such that $X \subseteq V$. The *local consistency* $\mathrm{LC}_G(y)$ is the probability that a node's label $y(x)$ agrees with the majority of labels of x's neighbors in G:

$$\mathrm{LC}_G(y) = \mathbb{P}\left(y(x) = \operatorname*{argmax}_{c \in \mathcal{Y}} \sum_{x' \in X \mid (x,x') \in E} \mathbb{1}_{y(x')=c} \right).$$

Note this is closely related to *homophily* [143, 39, 132].

With Local Consistency defined, we can now formally prove Theorem 1, reproduced below.

Theorem 1. Let $\boldsymbol{X}_{\mathrm{PT}}$ be a PT dataset, G_{PT} be a PT graph, and let $f_{\boldsymbol{\theta}^*}$ be an encoder pre-trained under a PT objective permissible under our framing that realizes a $\mathcal{L}_{\mathrm{SI}}$ value no more than ℓ^*. Then, under embedder f, the nearest-neighbor accuracy for a FT task y converges as dataset size increases to at least the local consistency (Definition 5) of y over G_{PT}.

Proof. Given f realizes SIPT-optimal embeddings, we know that if we define a r-NN predictor via the same radius r^* at which f achieves optimality, then this predictor will be correct exactly as often as the label of a given node in the graph G_{PT} agrees with the labels of its neighbors—which is $\mathrm{LC}_{G_{\mathrm{PT}}}(y)$. This classifier may not be well defined for small FT dataset sizes. However, as if data is not sufficiently dense, there may be no data points within the radius r of a given query. Similarly, without sufficient PT data, the LC computed over the empirical distribution of the graph G_{PT} may be a poor proxy for the true distribution. As PT and FT dataset sizes increase, however, we can achieve at least this performance. We may be able to achieve even higher performance if other effects motivate stronger performance at radii smaller than r^*, but this is not guaranteed. □

D.3 Proof of Corollary 1

Corollary 1. *Let $\boldsymbol{X}_{\mathrm{PT}} \in \mathcal{X}^N$, be a PT dataset with corresponding labels $\boldsymbol{y} \in \mathcal{Y}_{\mathrm{PT}}^N$. Define $G_{\mathrm{PT}} = (\boldsymbol{X}_{\mathrm{PT}}, E)$ such that $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ if and only if $y_i = y_j$.*

Then, the local consistency for a given FT task $\boldsymbol{y}^{(FT)}$ over G_{PT} (and thus by Theorem 1, the nearest-neighbor accuracy for any optimized SIPT embedder) is upper bounded by the probability that a sample x_i's fine-tuning label $y_i^{(FT)}$ agrees with the majority class label for task $\boldsymbol{y}^{(FT)}$ over the clique consisting of all nodes with the same pre-training *label y_i as x_i.*

Proof. This follows directly from the definition of Local Consistency, G_{PT}, and the law of total probability. In particular,

$$\begin{aligned}
\mathrm{LC}_{G_{\mathrm{PT}}}(y_{\mathrm{FT}}) &= \mathbb{P}\left(y_{\mathrm{FT}}(\boldsymbol{x}_i) = \operatorname*{argmax}_{\ell \in \mathcal{Y}_{\mathrm{FT}}} \sum_{\boldsymbol{x}_j \in \boldsymbol{X}_{\mathrm{PT}} \mid (\boldsymbol{x}_i, \boldsymbol{x}_j) \in E(G_{\mathrm{PT}})} \mathbb{1}_{y_{\mathrm{FT}}(\boldsymbol{x}_i)=\ell} \right) \\
&= \mathbb{P}\left(y_{\mathrm{FT}}(\boldsymbol{x}_i) = \mathrm{MC}(\boldsymbol{x}_i, y_{\mathrm{FT}}) \right) \\
&= \sum_{\ell_{\mathrm{PT}} \in \mathcal{Y}_{\mathrm{PT}}} \mathbb{P}(y_i = \ell_{\mathrm{PT}}) \mathbb{P}(y_{\mathrm{FT}}(\boldsymbol{x}_i) = \mathrm{MC}(\boldsymbol{x}_i, y_{\mathrm{FT}}) | y_i = \ell),
\end{aligned}$$

With Local consistency found, a simple application of Theorem 1 completes the proof. □

Note that this has a dependence on the PT dataset size as the probabilities $\mathbb{P}$ are taken over the empirical distribution induced by the dataset $\boldsymbol{X}_{\mathrm{PT}}$ and graph G_{PT} inherent in local consistency — if $\boldsymbol{X}_{\mathrm{PT}}$ is too small, these empirical distributions will be poor proxies for the true distribution and this bound will not hold tightly. However, once saturation is reached, it will not improve beyond this fixed upper bound relating to task correlation.

D.4 Proof of Corollary 2

Corollary 2. *Let $\boldsymbol{X}_{\mathrm{PT}}$ be a PT dataset that can be realized over a valid manifold $\mathcal{M}$. Assume $\boldsymbol{X}_{\mathrm{PT}}$ is sampled with full support over $\mathcal{M}$. Let $G_{\mathrm{PT}}(\boldsymbol{X}_{\mathrm{PT}}, E)$ be an r-nearest-neighbor graph over $\mathcal{M}$ (*e.g.*, $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ if and only if the geodesic distance between the two points on $\mathcal{M}$ is less than r: $\mathcal{D}_{\mathcal{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) < r$). Let $y^{(FT)}$ be a FT classification task that is almost everywhere smooth on the manifold.*

Then, as PT dataset size (and thus the size of G_{PT}) tends to ∞, and r tends to zero, the local consistency of $y^{(FT)}$ over G_{PT} (and thus by Theorem 1 the nearest-neighbor accuracy of an SIPT embedder) will likewise tend to 1.

Proof. As $r \to 0$, provided PT dataset size increases at a sufficient associated rate so as to maintain a constant minimum degree of G, we have the property that the total diameter over $\mathcal{M}$ contained in a node's local neighborhood within G_{PT} likewise decreases. Given some fixed node $\boldsymbol{x} \in \mathcal{M}$ that is within the interior of a set of constant y_{FT} label, this implies that, eventually, it will grow sufficiently small that all of $\boldsymbol{x}$'s neighbors share the same label as $\boldsymbol{x}$ under y_{FT}.

More concretely, it is clear that this point will occur exactly when r is the geodesic distance between $\boldsymbol{x}$ and the boundary of the surrounding constant-label patch containing $\boldsymbol{x}$. But, it is clear that the only sections of $\mathcal{M}$ will not have the property that neighborhoods around points will be constant w.r.t. y_{FT} labels will almost everywhere be patches within distance r of the points where y_{FT} changes.

This implies that as $r \to 0$, then almost everywhere will the neighborhoods around a node $\boldsymbol{x}$ be constant w.r.t. y_{FT}. However, this implies that almost everywhere would y_{FT} display perfect local consistency, as desired. □

E Semi-synthetic Experiments Validating Theoretical Results

We can further validate the theoretical analyses of our framework via semi-synthetic experiments. In particular, we create several datasets of natural language sentences augmented with synthetic graphs with known relationships to certain FT tasks (e.g., low or high local consistency, low or high rates of noise). We then use these datasets to validate three important properties of PT methods: First, do PT methods trained with a $\mathcal{L}_{\mathrm{SI}}$ and G_{PT} yield Nearest-neighbor FT performance in accordance with our theory? In particular, do (a) FT tasks with high local consistency over the PT graph offer better performance, and (b) those with very low local consistency offer worse performance? Second, do PT methods trained with a $\mathcal{L}_{\mathrm{SI}}$ and G_{PT} suffer significantly when pre-training graphs are polluted with noise? Finally, third, do the latent space geometry regularizing properties of $\mathcal{L}_{\mathrm{SI}}$ yield methods whose embeddings more clearly cluster than embeddings produced by traditional pre-training alone?

E.1 Pre-training & fine-tuning datasets

Across all experiments, our synthetic datasets consist of free-text sentences from https://www.kaggle.com/mikeortman/wikipedia-sentences (CC BY-SA 4.0 License).

Topics were assigned to these sentences by running Latent Dirichlet Allocation via Scikit-learn [75] over a Bag-of-words representation to 100 topics, with otherwise default parameters. Given the probabilities over all 100 topics, we treated the prediction of the most probable topic as a 100-class multi-class classification problem for our FT task in these experiments.

To test across various graphs, we produce a number of pre-training graphs per experiment, as detailed below.

E.2 Pre-training graphs

We use graphs spanning 3 categories. (1) A graph (CLIQUES) consisting of disconnected cliques, where sentences are linked in the graph if they share the same topic label. (2) Graphs composed of nearest-neighbor graphs defined over simplicial manifolds built using topic probabilities to localize sentences onto simplices. We explore manifolds with a range of topological complexity, including: PLANE, MÖBIUS, SPHERE, and TORUS. Finally, (3) we define three graphs according to a mechanistic process that allows us to control how topic labels relate to graph structure: first, so that topics are maximally conserved within local neighborhoods (NEIGHBORHOOD); second, by assigning sentences to nodes in the graph such that each graph motif corresponds to a unique topic (MOTIF); and third, such that node topics are driven by non-local graph structural features, on the basis of graphlet degree vectors (STRUCTURAL). Details for each pre-training graph formation are given below.

CLIQUES Graph Setup

To construct the Cliques graph setting, we choose a random subset of sentences as $\boldsymbol{X}_{\mathrm{PT}}$ and define $G_{\mathrm{PT}} = (\boldsymbol{X}_{\mathrm{PT}}, E)$ such that $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$ if and only if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ share the same topic label.

PLANE, MÖBIUS, SPHERE, & TORUS Graphs

For these graphs, we take a more involved practice to localize sentences onto specifiable simplicial manifolds, then construct pre-training graphs via radius nearest neighbor graphs on those manifolds. This involves several steps:

Localizing Sentences on Simplices We can localize any sentence in our overall dataset onto a 2-simplex by mapping them onto the (re-normalized) probabilities associated with their top-3 topics. Doing this means that the simplex on which they are localized has vertices corresponding to possible topics among our 100 total topics.

Stitching Topic-simplices Into Manifolds Given these topic-simplex localized sentences, we need to construct our manifolds. To do so, we first produce any arbitrary simplicial tiling of a 2-manifold. With this tiling, all that remains to localize sentences onto the manifold is to find a self-consistent mapping of topics to simplex vertices (in the tiling) such that all topic-simplices induced by this mapping have sufficiently many associated samples to enable roughly uniform sampling.

Sampling Points After finding a self-consistent map of topics to simplicial tiling vertices that satisfy density requirements, we can sample sentences onto the manifold. To make this process more uniform, we also calculate the relative entropy of each sentence (over the re-normalized probabilities of the top-3 topics), bin those entropies into buckets, then sample first what entropy bucket we wish to draw from such that the induced distribution of sentence entropies is approximately uniform, then sample within that entropy bucket.

Calculating on-Manifold Distances Finally, with sentences sampled and localized onto a simplicial manifold, we then need to compute approximate geodesic distances to enable building radius-nearest-neighbor graphs over these sentences. To do so, we use an approximate algorithm that considers only on-simplex distance (*e.g.*, it does not consider any curvature penalties) which is equivalent to calculating the distance between any pair of points over the simplices presuming they were flattened onto a plane (this flattening naturally does not preserve manifold topology, but along only the shortest path between any particular set of two points it is always possible to do so with a 2-manifold).

The above process describes how to produce a radius-nearest-neighbor graph for any specifiable manifold using our topic-model outputs. We do this for simplicial manifolds that correspond

topologically to a simple plane (PLANE), a möbius strip (MÖBIUS), a sphere (SPHERE), and a torus (TORUS).

STRUCTURAL, NEIGHBORHOOD & MOTIFS Graphs

In order to form these examples, we must (1) define our overall graphs, (2) featurize these graphs in a manner that is reflective of different forms of graph structure, then (3) use these featurizations to assign sentences to graph nodes to form our pre-training dataset.

Graph Construction We sample graphs by first building a base cycle of a parametrized size, then add motifs along this cycle by sampling small graphs from all possible connected graphs of size less than 6 nodes.

Node Featurization Nodes in this graph are then assigned internal features based on three notions of graph topology. For the "Neighborhood" label, a node n is identified according to an index-vector indicating which nodes in the graph are within shortest-path distance 3 of n. For the "Motif" label, n is identified based on its membership either in the base cycle or any of the attached random subgraphs. For the "Structural" label, n is identified based on its graphlet degree vector (of order 4). For structural and homophily features, categorical labels are then produced by feeding these raw representations through a k-means clustering algorithm.

Sentence Assignment We assign sentences to nodes in multiple ways so that we can produce datasets that reflect each of the notions of graph structure discussed previously. In particular, for either the neighborhood, motif, or structural labels, each sentence topic is matched to a node label, then sentences are assigned randomly to nodes in the graph with a matching topic label. Note that this produces a dataset where the graph structure is only partially reflected by the node's features, which is itself another useful test of the SIPT method, as it would not be useful if SIPT could only capture data in contexts where the graph was perfectly reflected by the node features themselves.

Expected local consistency between graphs G_{PT} and the topic prediction FT task

Of all these graphs, we expect that topics will display a low local consistency over the STRUCTURAL graph and a moderately high local consistency over the MOTIF graph (as graph motifs are all connected components), and high local consistency everywhere else.

E.3 Network Architecture & Hyperparameters

The Cliques and Mechanistic experiments use a shallow Transformer model with 2 layers and 10 hidden units. The Manifold experiments use a 3-layer Transformer model with 256 hidden units. Hyperparameters were not tuned but were chosen by hand to produce as small a network as possible while permitting reasonable learning dynamics.

E.4 Experimental setup

To answer our three questions, we will pre-train models under both traditional LM pre-training alone and a new, structure-inducing PT (SIPT) method within our paradigm that augments the loss with a contrastive learning loss over G_{PT}, with $\lambda_{\mathrm{SI}} = 0.1$. Both models use a shallow transformer encoder for $f_{\boldsymbol{\theta}}$ and a character-level tokenization scheme. Final results are reported via the AUROC of 3-nearest-neighbor classifiers over the latent space, per-sample embeddings. In line with our theoretical predictions, we expect to see higher NN FT performance in all settings where the FT task (topic prediction) has high local consistency over the graph G_{PT} (all graphs except STRUCTURAL) and worse performance in the case where the local consistency is very low (STRUCTURAL).

We also assess the stability of our method as the graph G_{PT} is noised using the CLIQUES graph by randomly adding additional edges with varying rates.

a

LC	Pre-training graph	MPT	SIPT
↑	CLIQUES	0.62	**0.94**
↑	PLANE	0.51	**0.94**
↑	MÖBIUS	0.55	**0.89**
↑	SPHERE	0.51	**0.86**
↑	TORUS	0.54	**0.77**
↑	NEIGHBORHOOD	0.94	**0.99**
↑	MOTIF	0.73	**0.88**
↓	STRUCTURAL	**0.90**	0.79

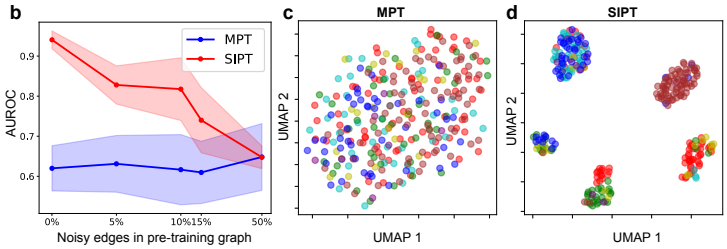

Figure 5: **Semi-synthetic Experiments Results:** **(a)** Comparisons between nearest-neighbor FT AUROC (higher is better) of LM PT models and SIPT models over various graphs with various forms of structural alignment. LC indicates the label consistency between FT task and G_{PT} (Definition 5). **(b)** Nearest-neighbor FT AUROC vs. noise rate. Up to 10% noise SIPT dramatically outperforms LM PT, and at 50% noise, the two approaches are equal. **(c-d)** Embedding space of MPT and SIPT models on the MÖBIUS dataset. Point colors indicate topic labels. SIPT's embedding space reflects the structure of the PT graph, whereas MPT does not.

E.5 Semi-synthetic Results

E.5.1 SIPT improves performance over LM PT by 0.26 ± 0.13 AUROC on graphs where the topic task has a high local consistency

As can be seen in Figure 5a, SIPT offers significant improvements over LM PT in nearest-neighbor FT AUROC across all graph types with strong topic local consistency.

E.5.2 SIPT's empirical results are in agreement with theoretical findings

In line with our theoretical findings, SIPT only under-performs LM PT on the STRUCTURAL graph where the topic task (by design) does not have strong local consistency. This validates our theoretical results by showing that local consistency strongly predicts Nearest-neighbor FT performance.

E.5.3 SIPT is robust to incomplete and noisy pre-training graphs

Figure 5b shows Nearest-neighbor FT AUROC as a function of noise rate on the CLIQUES graph. For up to 15% noise, SIPT shows improvements over LM PT, and even at 50% noise, the two approaches perform comparably.

E.5.4 SIPT pre-trained embeddings show stronger clustering than LM PT embeddings

Figure 5c-d shows embeddings produced under the MÖBIUS graph either by LM PT or SIPT, clustered via UMAP into 2 dimensions. It is clear visually from these figures that SIPT embeddings show clear clusters strongly associated with the topic-modelling FT task, whereas LM PT embeddings do not.

E.6 Conclusions

From these analyses, we see that augmenting PT with per-sample structure-inducing objectives can both (1) offer significant advantages over existing PT architectures and (2) permit analytical reasoning about which FT tasks PT will offer improvements. These findings are not surprising; in these semi-synthetic experiments, we designed our graphs explicitly to have either high or low local consistency with respect to our FT task so that we could probe exactly whether SIPT methods would behave in accordance with theory in tightly controlled settings. In this way, the graphs G_{PT} used here may not be reflective of graphs in the real world, which will be chosen more independently of specific FT tasks. To address this, in the Results section, we demonstrate experimental results over diverse real-world datasets with real, FT-task-independent graphs to show that the gains persist in more realistic scenarios.

F Further Details on Real-world Experiments

F.1 Further Details on the PROTEINS Dataset and FT tasks

PT Dataset We use a dataset of ∼1.5M protein sequences from the Stanford Tree-of-life dataset [144] (https://snap.stanford.edu/tree-of-life/data.html). The associated Github repository for this resource lists an MIT license.

PT Graph Two proteins are linked in G_{PT} if and only if they are documented in the scientific literature to interact, according to the tree-of-life interaction dataset. This is an external knowledge graph.

FT Dataset/Tasks We use the TAPE FT benchmark tasks [84], including Remote homology (RH), a per-sequence classification task to predict protein fold category (metric: accuracy); Secondary structure (SS), a per-token classification task to predict amino acid structural properties (metric: accuracy); Stability (ST) & Fluorescence (FL), per-sequence, regression tasks to predict a protein's stability and fluorescence, respectively (metric: Spearman's ρ); and Contact prediction (CP), an intra-sequence classification task to predict which pairs of amino acids are in contact in the protein's 3D conformation (metric: Precision at $L/5$).

Baselines We compare against the published TAPE model [84], which uses an LM task alone as our per-token comparison point, and the PLUS [73] model, which optimizes for LM and supervised classification jointly, for our per-sample comparison point.

The tasks in the TAPE benchmark [84] on which we test are described more fully below. All these datasets are publicly available. All datasets can be obtained directly on TAPE's Github (https://github.com/songlab-cal/tape#data), which lists no licenses for these datasets though the overall Github is released under a BSD 3-Clause "New" or "Revised" License.

Remote Homology This is a per-sequence, multi-class classification problem, evaluated using accuracy, which tasks a model to predict a protein fold category at a per-sequence level. This task's dataset contains 12,312/736/718 train/val/test proteins and is originally sourced from [35].

Secondary Structure This is a per-token, multi-class classification problem, evaluated using accuracy, which tasks a model to predict the structural properties of each amino acid in the final, folded protein. This task's dataset contains 8,678/2,170/513 train/val/test proteins, and is originally sourced from [49].

Stability This is a per-sequence, continuous regression problem evaluated using the Spearman correlation coefficient, which tasks a model to predict the protein's stability in response to environmental conditions. This task's dataset contains 53,679/2,447/12,839 train/val/test proteins, and is originally sourced from [89].

Fluorescence This is a per-sequence, continuous regression problem evaluated using the Spearman correlation coefficient, which tasks a model to predict how brightly a protein will fluoresce. This task's dataset contains 21,446/5,362/27,217 train/val/test proteins, and is originally sourced from [92].

F.2 Further Details on the ABSTRACTS Dataset and FT tasks

PT Dataset We use a dataset of ∼650K free-text scientific article abstracts from the Microsoft Academic Graph (MAG) dataset [108, 36]. The ABSTRACTS PT data (the Microsoft Academic Graph dataset) is licensed with an Open Data Commons Attribution License (ODC-By) v1.0 license.

PT Graph Two abstracts are linked in G_{PT} if and only if their corresponding papers cite one another. This is a self-supervised graph.

FT Dataset/Task We use a subset of the fine-tuning tasks used in the SciBERT paper [4], including Paper field (PF), SciCite (SC), ACL-ARC (AA), and SciERC Relation Extraction (SRE), all

of which are per-sentence classification problems (metric: Macro-F1). PF tasks models to predict a paper's area of study from its title, SC & AA tasks both predict an "intent" label for citations, and SRE is a relation extraction task.

Baseline We compare against the published SciBERT model [4] as our per-token comparison and lack an associated per-sample comparison as we don't know of any published per-sample models in the academic papers modality.

The tasks in the SciBERT benchmark [4] on which we test are described more fully below. All tasks here are per-sentence, multi-class classification problems (i.e., we do not study any per-token tasks), and all are evaluated in Macro-F1 (out of 1). All FT datasets can be obtained from the SciBERT Github (https://github.com/allenai/scibert), which lists no dataset-specific licenses but is released with an Apache-2.0 license.

Paper Field This problem asks models to predict a paper's area of study given its title. This task's dataset contains 84,000/5,599/22,399 train/val/test sentences. Though the original dataset is derived from the MAG [108], it was formulated into this task format by SciBERT directly [4].

SciCite This problem tasks models to predict an "intent" label for sentences that cite other scientific works within academic articles. This task's dataset contains 7,320/916/1,861 train/val/test sentences, and is originally sourced from [13].

ACL-ARC This problem tasks models to predict an "intent" label for sentences that cite other scientific works within academic articles. This task's dataset contains 1,688/114/139 train/val/test sentences and is originally sourced from [43].

F.3 Further Details on the NETWORKS Dataset and FT tasks

PT Dataset We use a dataset of $\sim$70K protein-protein interaction (PPI) ego-networks here, sourced from [37]. Each individual sample here describes a single protein, realized as a biological network (*i.e.*, an attributed graph) corresponding to the ego-network about that protein (*i.e.*, a small subgraph containing all nodes within the target protein) in a broader PPI graph. Unlike our other domains, this domain does not contain sequences. The NETWORKS PT dataset releases its code and dataset files under an MIT license.

PT Graph The dataset from [37] is labeled with the presence or absence of any of 4000 protein gene ontology terms associated with the central protein in each PPI ego network. Leveraging these labels, two PPI ego-networks are linked in G_{PT} if and only if the Hamming distance between their observed label vectors is no more than 9. This is an alternate-representation nearest-neighbor graph.

FT Dataset/Tasks Our FT task is the multi-label binary classification of the 40 gene-ontology term annotations (metric: macro-AUROC) used in [37]. We use the PT set for FT training and evaluate the model on a held-out random 10% split.

Baselines We compare against both attribute-masking [37] and multi-task supervised PT.

The Networks FT task is a multi-task, binary classification task. Recall that the dataset here consists of PPI ego-networks, which means that an individual sample input to the model is an attributed graph $\boldsymbol{x}$ which contains a central node, corresponding to a protein, along with the ego-graph surrounding that node in a larger PPI graph. This ego-graph can thus be seen to correspond to the central protein, and the FT and PT tasks leverage this association, as both of which flag whether or not that central protein is associated with particular gene-ontology (GO) terms (annotations relating to protein properties or function applied in the literature). The PT tasks contain 4000 possible GO annotations, but the FT tasks correspond to a smaller set of only 40 GO terms, chosen as they were of greater interest than the full set. See the original source ([37]) for more information and full details.

F.4 Further Details on Experimental Procedure

To minimize computational burden, we do not pre-train a structure-inducing model from scratch for PROTEINS and ABSTRACTS datasets. Instead, we initialize a model from the per-token baseline directly, then perform additional pre-training for only a small number of epochs under the new SIPT loss subdivision. We assess both multi-similarity and contrastive $\mathcal{L}_{\text{SI}}$ variants in these domains. On the NETWORKS dataset, we pre-train all models (including baselines) from scratch, and based on early experimental results, we only assess the contrastive loss variant.

F.5 Further Details on Ablation Studies

Note that the warm-start procedure described above on the PROTEINS and ABSTRACTS domains allows a powerful ablation study: by additionally training a PT model from the per-token baseline with $\lambda_{\text{SI}} = 0$, we can uniquely assess the impact of the new loss term, rather than simply additional training or the different PT dataset. We perform this ablation study for all applicable datasets. For the NETWORKS dataset, no additional ablation studies are needed to assess the impact of the loss term, given all models are trained from scratch with the same early-stop procedures.

F.6 Further Details on Choosing λ_{SI}

For the PROTEINS and ABSTRACTS dataset, to choose the optimal value of λ_{SI} for use at PT time, we pre-trained several models and evaluated their efficacy in a link retrieval task on $G_{\text{PT}} = (V, E)$. In particular, we score a node embedder f by embedding all nodes $n \in V$ as $f(n)$, then rank all other nodes n' by the euclidean distance between $f(n)$ and $f(n')$, and assess this ranked list via IR metrics including label ranking average precision (LRAP), normalized discounted cumulative gain (nDCG), average precision (AP), and mean reciprocal rank (MRR), where a node n' is deemed to be a "successful" retrieval for n if $(n, n') \in E$. In this way, note that we choose λ_{SI} in a manner that is independent of the fine-tuning task and can be determined solely based on the PT data. Final results for these experiments are shown in Methods Table 9 for the proteins dataset and Methods Table 10 for scientific articles.

Ultimately, this process suggests that λ_{SI} of 0.1 is a robust setting, and as such, 0.1 was used directly for the NETWORKS task without further optimization.

F.7 Further Details on Architecture & Hyperparameters

The architectures of our encoders for the PROTEINS and ABSTRACTS domains are fully determined from our source models in TAPE [84] and SciBERT [4]. In particular, for proteins and scientific articles, we use a 12-layer Transformer with a hidden size of 768, an intermediate size of 3072, and 12 attention heads. Provided TAPE and SciBERT tokenizers are also used. A single linear layer to the output dimensionality of each task is used s the prediction head, taking as input the output of the final layer's `[CLS]` token as a whole-sequence embedding. We also tested either pre-training for a single or for four additional epochs, based on validation set performance, and ultimately used a single epoch for proteins and four for scientific articles.

For the NETWORKS domain, we match the architecture used in the original source [37] for the mask model runs. Save that for computational efficiency, we scale the batch size up as high as it can go, then proportionally scale up the learning rate to account for the larger batch size. This corresponds to a batch size of 1024, the learning rate of 0.01, a GCNN encoder type of GIN, embedding dimensions of 300, 5 layers, 10% dropout, mean pooling, and a JK strategy of "last".

Fine-tuning hyperparameters (learning rate, batch size, and the number of epochs) were determined based on a combination of existing results, hyperparameter tuning, and machine limitations. On proteins, most hyperparameters were set to follow those reported for a LM PT model in [69], though additional limited hyperparameter searches were performed to validate that these choices were adequate. As the original source for these hyperparameters was an LM PT model, any bias here should be *against* SIPT, meaning this is a conservative choice. Early stopping (based on the number

Task	Batch Size	LR
Remote Homology	16	1e-5
Fluorescence	128	5e-5
Stability	512	1e-4
Secondary Structure	16	1e-5

Table 5: Final hyperparameters for our PROTEINS domain. All tasks used 200 total epochs and performed early stopping after 25 epochs of no validation set improvement. LR, learning rate.

Task	# Epochs	LR
Paper Field	2	5e-5
ACL-ARC	4/5	5e-5
SciCite	3/2	1e-5

Table 6: Final hyperparameters for our ABSTRACTS dataset. All models used a batch size of 32 and no early stopping to match the original SciBERT paper [4]. LR, learning rate. A / B = [LM PT Hyperparameter] / [SIPT Hyperparameter].

of epochs without observing improvement in the validation set performance) was employed, and batch size was set as large as possible given the limitations of the underlying machine. For the PLUS reproduction, we compared hyperparameters analogous to the reported PLUS hyperparameters for other tasks and analogous to our hyperparameters for other tasks and used those that performed best on the validation set. For scientific articles, we performed a grid search to optimize downstream task performance on the validation set, with the learning rate varying between 5e-6 and 5e-5 and the number of epochs between 2 and 5. The same grid search was used in the original SciBERT method. We additionally match the SciBERT benchmark by applying a dropout of 0.1, using the Adam optimizer with linear warm-up and decay, a batch size of 32, and no early stopping. For the NETWORKS, FT hyperparameters were again chosen to match the original source model [37] to save the increase in batch size and learning rate. No additional hyperparameter search was performed.

Final hyperparameters for each downstream task are shown in Tables 5 for proteins and 6 for scientific articles.

F.8 Further Details on Implementation and Compute Environment

We leverage PyTorch for our codebase. FT Experiments and NETWORKS PT were run over various ubuntu machines (versions ranged from 16.04 to 20.04) with a variety of NVIDIA GPUs. PROTEINS and ABSTRACTS PT runs were performed on a Power 9 system, each run using 4 NVIDIA 32 GB V100 GPUs with InfiniBand at half precision.

F.9 Full Results

Here we provide the raw FT results for all tasks. Raw results for the NETWORKS domain are shown in Figure 6, and for the PROTEINS and ABSTRACTS domains, respectively, are shown in Tables 7, 8.

Model	RH	FL	ST	SS	CP
TAPE	21%	**0.68**	0.73	73%	0.32
PLUS	19.8%$\pm$1.7*	0.63	0.76	73%	N/A
LM PT	23.8%$\pm$1.1	0.67$\pm$0.00	0.76$\pm$0.02	73.9%$\pm$0.0	0.38
SIPT-C	25.1%$\pm$0.6	**0.68$\pm$0.00**	**0.77$\pm$0.01**	73.9%$\pm$0.0	0.38
SIPT-M	**26.6%$\pm$1.0**	**0.68$\pm$0.00**	0.76$\pm$0.01	**74.2%$\pm$0.1**	**0.39**

Table 7: Results of the TAPE Transformer [84], the PLUS Transformer [73] (*: our measurements), our LM PT baseline, and two SIPT variants ("-C" indicates the contrastive loss, "-M" the multisimilarity loss). Higher is better.

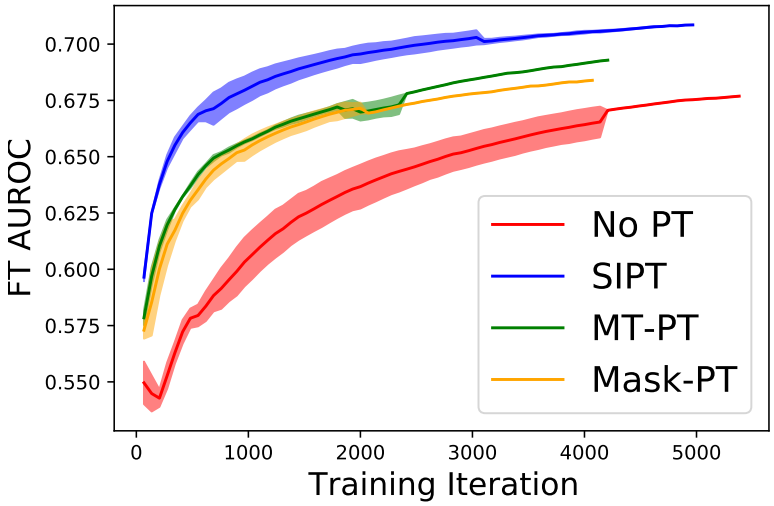

Figure 6: **Fine-tuning (FT) Performance over NETWORKS**: FT AUROC as a function of FT iteration for the NETWORKS dataset. The SIPT method converges faster and performs better than intra-sample (masked node modelling) or per-sample (multi-task classification) pre-training.

Model	PF	SC	AA	SRE
SciBERT	**0.66**	0.85	0.71	0.80
LM PT	**0.66**$\pm$**0.0**	0.85$\pm$0.01	0.70$\pm$0.05	0.80$\pm$0.01
SIPT-C	**0.66**$\pm$**0.0**	**0.86**$\pm$**0.01**	**0.76**$\pm$**0.02**	**0.81**$\pm$**0.00**
SIPT-M	**0.66**$\pm$**0.0**	0.85$\pm$0.00	0.73$\pm$0.05	N/A

Table 8: Results of the original SciBERT [4] model, our own LM PT baseline, and two SIPT variants ("-C" indicates the contrastive loss, "-M" the multisimilarity loss). Higher is better.

F.10 SIPT Results are in Accordance with Theory and Guiding Hypothesis

Results over all real-world domains are consistent with our theoretical analyses and guiding hypothesis. We can also analyze the extent to which induced structure helps non-NLP domains by examining the results of our λ_{SI} tuning procedure. In particular, we find that far less structure-inducing is necessary on our ABSTRACTS dataset ($\lambda_{\text{SI}} = 0.01$) than on our PROTEINS dataset ($\lambda_{\text{SI}} = 0.1$). This agrees with our guiding hypothesis that per-sample latent space regularization is much more necessary on non-NLP domains than on NLP domains.

To demonstrate this, we show the final results for the guiding link-retrieval task for the PROTEINS domain in Table 9 and for the ABSTRACTS domain in Table 10. In both settings, we compare the following models.

Random Nodes are embedded with random vectors to assess chance performance.

Initial Model Nodes are embedded with the base pre-trained model we build on in our experiments without further modifications. This model is TAPE [84] for proteins and SciBERT [4] for scientific articles.

LM PT Nodes are embedded with the final encoder after additional pre-training on our graph-augmented datasets, but without any SIPT (i.e., $\lambda_{\text{SI}} = 0$).

CS RoBERTa *(for scientific articles only)* Nodes are embedded via [29]'s DAPT CS RoBERTa model, which is another LM PT model over scientific abstracts which performed very well on ACL-ARC, the task on which SIPT does best in scientific articles.

Method	λ_{SI}	LRAP	nDCG	AP	MRR
Random Baseline	N/A	0.88%	27.1%	0.88%	0.003
TAPE [84]	N/A	8.50%	34.9%	2.41%	0.226
LM PT Baseline	0	8.92%	38.0%	2.33%	0.238
SIPT (TAPE Initialized)	0.01	9.69%	39.1%	2.56%	0.254
	0.10	10.95%	39.4%	3.46%	0.260
	0.50	10.54%	40.3%	3.43%	0.246
	0.90	10.12%	39.0%	3.16%	0.237
	0.99	14.50%	37.5%	3.13%	0.236

Table 9: PT set link-retrieval performance for a random baseline, the raw TAPE model, and SIPT for various weighting parameters λ_{SI} on the dataset of protein sequences. LRAP, label ranking average precision; nDCG, normalized discounted cumulative gain; AP, average precision; MRR, mean reciprocal rank. Higher values indicate better performance. Highlighted in grey are realizations of SIPT framework that yield better results than the strongest baseline, providing evidence that incorporating sequence-level relational information into PT (*i.e.*, $\lambda_{\text{SI}} > 0$) leads to improved performance.

Method	λ_{SI}	LRAP	nDCG	AP	MRR
Random Baseline	N/A	0.89%	26.0%	0.27%	0.016
SciBERT [4]	N/A	17.22%	52.8%	5.16%	0.272
LM PT Baseline (SciBERT initialized)	0	16.79%	35.4%	5.00%	0.271
DAPT CS RoBERTa [29]	N/A	32.56%	50.3%	12.86%	0.459
LM PT Baseline (CS RoBERTa initialized)	0	30.58%	48.3%	12.36%	0.438
SIPT (SciBERT initialized)	0.01	42.26%	58.7%	14.23%	0.536
	0.10	34.73%	52.5%	9.39%	0.457
	0.50	32.85%	50.8%	8.37%	0.438
	0.90	31.61%	49.8%	7.82%	0.426
	0.99	30.72%	49.0%	6.80%	0.415
SIPT (CS RoBERTa initialized)	0.01	33.32%	51.2%	8.61%	0.448
	0.10	25.46%	44.4%	5.88%	0.359
	0.50	25.08%	44.0%	6.08%	0.355
	0.90	22.43%	41.6%	4.27%	0.317
	0.99	22.38%	41.5%	4.68%	0.316

Table 10: PT set link-retrieval performance for a random baseline, the raw SciBERT model, and SIPT for various weighting parameters λ_{SI} on the scientific articles dataset. LRAP, label ranking average precision; nDCG, normalized discounted cumulative gain; AP, average precision; MRR, mean reciprocal rank. Higher values indicate better performance. Highlighted in grey are realizations of SIPT framework that yield better results than the strongest baseline, providing evidence that incorporating sequence-level relational information into PT (*i.e.*, $\lambda_{\text{SI}} > 0$) leads to improved performance.

SIPT *(for various values of λ_{SI}).* Nodes are represented via SIPT PT models at the specified weighting. For proteins, all SIPT models are initialized from TAPE, but for scientific articles, we test against both initializing from SciBERT and CS RoBERTa (as both are just different, domain-specific LM PT models).

Note that in addition to the discrepancy in the magnitude of improvement (over scientific articles, average precision goes from 12.9% to 14.2%, vs. 2.4% to 3.5% on proteins, which is proportionally much more significant), we can also see that SIPT improves retrieval performance over the baselines for proteins much more than it does for scientific articles. This is, admittedly, largely due to [29]'s CS RoBERTa model's surprisingly good performance without any modifications, however as we also compare SIPT pre-trained from a CS RoBERTa model and it does not demonstrate significant improvements, we still feel this is a fair comparison. These findings are consistent with our hypothesis that SIPT will offer more significant advantages in non-natural language domains.

G Review of Language Model Pre-training Methods

In this supplementary section, we describe all of the models featured in our review (Figure 4 and Table 4) and highlight key details of their approach.

G.1 Language modelling alone

[76] General domain NLP; ELMO leverages a biLSTM to perform language modelling; unlike later methods, for FT tasks, models do not typically re-train the entire LSTM but rather use a weighted combination of model interior hidden states as (at FT time) static word-embeddings.

[65] General domain NLP; RoBERTa includes only a masked language modelling objective.

[81, 82, 5] General domain NLP; The GPT series of models use autoregressive language modelling alone and focus on generative language tasks, not general PT/FT, though GPT-III does show that by reframing many classical NLP fine-tuning tasks as generative language tasks, GPT-III can still offer a compelling zero and few-shot solution to these tasks using only the pre-trained embedder [5].

[55] General domain NLP; BART utilizes a denoising language-model objective across various noising constraints.

[18] General domain NLP; UniLM integrates several different kinds of language modelling, including bidirectional, unidirectional, and sequence-to-sequence LMs. They impose no other PT losses.

[84, 88, 2] Protein sequences; Various methods have explored language modelling alone for protein sequences. One notable entry is the TAPE benchmark, which also introduces a public benchmark of FT tasks for future comparisons.

[136] Molecular Graphs; Molecular Graph BERT (MG-BERT; no relation to MG-BERT [3]) uses masked atom prediction to pre-train a GNN over molecular graphs.

[15] General domain NLP; This paper pre-trains a model for multi-lingual language modelling, using only a multi-lingual masked language modelling objective.

[29] General domain NLP; DAPT advocates for continual pre-training on increasingly task-focused text to improve its relevance to various downstream tasks. DAPT uses a RoBERTa baseline pre-training model, which includes only a masked language modelling objective. It shows significant gains after adaptation. However, as they only adapt the pre-training context to the more focused text, this induces no additional constraints on the latent space geometry.

[42] General domain NLP; SpanBERT changes the traditional masked language modelling task to a task in which contiguous spans are masked wholesale, rather than individual tokens.

G.2 Language modelling & templated tasks/prompting as language modelling

[83] General domain NLP; T5 not only performs a robust analysis of various existing pre-training strategies but also introduces the "text-to-text" style of diverse pre-training, in which various downstream NLP tasks can be re-realized as language modelling tasks via templating and prompting, then integrated into language model pre-training alongside unsupervised objectives (such as traditional masked language modelling, albeit realized as a sequence-to-sequence task). As they realize all these downstream tasks as additional language modelling tasks, they neither officially produce a directly constrained per-sample embedding nor constrain the geometry of $\mathcal{Z}$ beyond traditional masked language modelling.

[142] General domain NLP; CALM builds on ideas from T5 to propose a text-to-text pre-training objective that leverages recognized per-token KG entities from the source text as a generative prompt.

[90] General domain NLP; T0pp extends the architecture of T5 [83] to ingest templated language modelling task from a wide variety of possible input tasks, then evaluates its performance in a zero-shot manner on unseen fine-tuning tasks.

G.3 Language modelling & Per-token KG Integration

[103] General domain NLP; ERNIE 1 augments traditional MLM with entity-specific masking (e.g., masking the word "Mozart" from the sentence "Mozart was a musician") to force the model to recover common-sense knowledge about named entities.

[33] General domain NLP; KgPLM adapts the discriminative training ideas of ELECTRA [12] alongside the idea of entity masking explored previously. They perform entity masking and a discriminative loss identifying which tokens were replaced focused on entity replacements.

[79] General domain NLP; ERICA presents a mechanism for leveraging contrastive learning and distant supervision to incorporate external knowledge into a PLM for improving language understanding. ERICA augments MLM with two per-token tasks to ensure the per-token representations within a document reflect the structure of the KG. First, ERICA ensures that the pooled representations of head and tail entities are similar when conditioned on a relation (which is prepended to the document prior to embedding). Second, ERICA ensures that relation embeddings (defined as concatenated head, tail per-token entity embeddings) are similar within and across documents. As both tasks are done on per-token embeddings and never at a per-sample level, this approach induces minimal constraints on the per-sample latent space.

[77] General domain NLP; Know-BERT integrates per-token entity information into an MLM pre-training scheme by performing unconstrained attention over a per-entity knowledge graph (only on pre-identified candidate entity spans), alongside any available entity linking supervision information via direct Named Entity Linking. This has similarities with [130] and [23].

[130] Biomedical domain NLP; KeBioLM integrates a per-token KG into a biomedical language model by augmenting token entity representations with attention lookups into a biomedical KG (regardless of whether the attended entities match a given entity mention in the source text, though they do only apply this on recognized entities). To ensure this attention is meaningful, they perform named entity linking and recognition as auxiliary PT objectives, leveraging the same KG embeddings used during the attention calculation. In doing so, the method incentivizes per-token representations to be similar to their associated entity representations, thus ensuring that the entities are reflected in the attention over the KG. KG embeddings are initialized using Trans-E [74]. Their usage of automatically attending over entities within their language model (without explicit constraints on those matches) is motivated by [23]'s work in [23] and has similarities to Know-BERT [77].

[118] General domain NLP; LUKE performs pre-training using MLM and an entity-specific masking/recognition scheme that is a slight variation on the traditional entity-specific masking [103] proposed. At FT time, they have other knowledge-specific integrations, including specialized query matrices in KQV attention based on attending to either traditional tokens or entities. However, at PT time, LUKE's only modulation over a ROBERTA [65] baseline is an entity masking task.

[101] General domain NLP; COLAKE performs a priori entity linking on the source text, then replaces per-token mentions with entity embeddings, and appends to the input text subgraphs from a (relational) knowledge graph, including both neighboring mentions and relations in the augmented input text block. This input is then encoded via a transformer that limits attention flow between tokens of different types and trains the entire ensemble with masked language, entity, and relation modelling.

[117] General domain NLP; In this paper, traditional masked language modelling is augmented with an entity-replacement-detection task. Named entity recognition and linking are performed before pre-training, and entity replacements are constrained to be the same type as the true entity.

[58] Knowledge Graph Completion; LP-BERT constructs a specialized pre-training corpus consisting of entity-relation statements from a knowledge graph. This is used in a pre-training context

under three pre-training tasks: masked language modelling, masked entity modelling, and masked relationship modelling. All three are per-token, and no per-sample tasks are used at pre-training time.

[100] Multilingual Language Models; UD-PrLM examines multilingual pre-training, and aims to improve it by incorporating universal dependency parse trees into the model. They incorporate a per-token task to align tokens with identified dependency parse tree components, alongside masked language modelling.

G.4 Language modelling, Per-token KG Integration, & Supervised Classification

[104, 102] General domain NLP; ERNIE 2.0 & 3.0 augments traditional MLM with entity-specific masking (e.g., masking the word "Mozart" from the sentence "Mozart was a musician") as well as a multi-task per-sample task, largely motivated at classifying a block of text based on internal text cohesion (predict the true order of the sentences within an input sample & identify whether the sentences within the input sample are spatial neighbors, come from the same document, or come from different documents). ERNIE 3.0 additionally augments pre-training with a per-token relation-embedding task using cloze-filling as a vehicle to perform relation extraction on pre-specified per-token KGs.

[139] General domain NLP; ERNIE (no relation to [103, 104]) uses both architectural and objective-function changes to inject per-token knowledge into PT. Specifically, they separately embed all named entities in a sample using the architecture to join contextualized entity embeddings alongside the embeddings of tokens, realizing that entity in the span and performing entity-specific masking. In addition, they simultaneously perform standard MLM and next-sentence prediction in the manner of BERT [17].

[3] General domain NLP; MG-BERT introduces a GCNN layer after BERT token, aggregating token embeddings together over a unified graph consisting both of co-occurrence relationships and knowledge graph relationships.

[128] General domain NLP; JAKET embeds entities by extracting per-token representations of entity texts inside per-entity descriptions, then produces updated KG embeddings via a graph attention network [106]. Those embeddings are then fed into a language model alongside per-token embeddings corresponding to those entities. The entire model is trained according to an MLM objective, plus entity category prediction and relation prediction (only on the entity embeddings extracted from entity descriptions and fed through the GCNN—*not* on the raw entities within the contextualized text).

[34] Biomedical NLP; BERT-MK introduces a transformer-based subgraph summarization network that produces entity embeddings for dynamically chosen subgraphs of a given knowledge graph. This network is trained via a contrastive triplet-validity objective. These are then fused with per-token embeddings in free-text based on apriori entity-token matching (*i.e.*, named entity recognition and linking must be performed first and separately before using this model).

[99] General domain NLP; Coke is similar to ERNIE [139], JAKET [128], and BERT-MK [34] in that it aggregates entity information by leveraging a GCNN over a restricted dynamic context KG based on token-entity mentions then integrates those augmented embeddings into the per-token embeddings of a BERT-style pretrained model (similar to JAKET and BERT-MK), but also leverages the denoising entity autoencoder task of ERNIE [139]. In addition, in the variant of COKE derived from the BERT model, COKE also employs the next-sentence prediction task introduced in BERT [17].

[135] Medical domain NLP; SMedBERT leverages a complex, multi-faceted loss including MLM, Sentence-order prediction SOP (as introduced in, e.g., ALBERT [52]), and includes per-token KG information by aggregating token embeddings across KG embeddings (produced via trans-H [114]) corresponding to matching entities and the neighbors of matching entities in the KG. They also include relation and entity masking variations to ensure the PT model

learns per-token information corresponding to the KG. This method bares similarity to Coke [99] and JAKET [128]. However, unlike Coke and JAKET, SMedBERT realizes the entity/neighbor matching via a geometric objective, which results in an explicit per-token knowledge graph alignment.

[129] General domain NLP; Dict-BERT focuses on augmenting BERT by concatenating definitions of rare words via a per-token KG integration. They add two additional tasks atop the traditional MLM task. First, a task maximizing the mutual information between a masked rare word (treated as a named entity) and its definition (represented as the per-token embedding of the first mention of the entity in the concatenated definition). Second, a task discriminating valid rare word definition per-sequence embeddings from non rare-word definition embeddings via a classification objective.

[44] Sentiment Analysis; SentiLARE integrates sentiment analysis and labels into pre-training by including word polarity signals during masked language modelling and embedding and augmenting pre-training with a supervised sentence sentiment prediction. Word polarities are determined via an external knowledge base integrated at the per-token level.

[140] Dialogue Modelling; SPIDER augments traditional MLM and NSP pre-training with two tasks specific to dialogue modelling: first, utterance order prediction, in which individual utterances (which are nested within a larger sample) are shuffled and the true order is predicted, and a geometric task ensuring that subject, verb, object triples from the utterances obey a geometric relationship inspired by KG embedding methods.

G.5 Language modelling & Graph link-prediction realized as single-task classification

These methods all employ some variant of a graph link-prediction task over their data. However, they all realize this link prediction task not by enforcing any relationship between independent sample embeddings but rather by concatenating samples corresponding to linked (or unlinked, for negative samples) pairs of vertices in the source graph, then framing the learning problem as a binary or multi-class classification problem over the (now concatenated) single output whole sample embedding. In doing so, they transform the task from one that implies a deep geometric constraint over the output latent space to one that only enforces an intra-sample objective and imposes only a shallow geometric constraint on the per-sample latent space.

[17] General domain NLP; Masked language model plus the binary classification of whether the input text block is sequentially consistent, with samples chosen via true positive pairs vs. randomly joined sentences. This can be seen as a link prediction task over a graph consisting of independent, disconnected "sticks", with each stick corresponding to sentences in the documents in the corpus, in sequential order.

[52] General domain NLP; Masked language model plus the binary classification of whether the input text block is sequentially consistent, with samples chosen via true positive pairs vs. reordered positive sentence pairs. This can be seen as a link prediction task over a directed graph consisting of independent, disconnected "sticks", with each stick corresponding to sentences in the documents in the corpus, in sequential order, with edge direction indicating sequential ordering.

[124] General domain NLP; Masked language model plus the classification of whether the input text block contains sentences from either (1) random documents, (2) a sequentially consistent pair within a single document, or (3) within a pair of sentences within two linked documents according to a document linking graph G. This can be seen as a link prediction/edge classification task over a graph whose nodes are text blocks in the corpus, with two distinct edge modalities. First, to capture sequential consistency within a document, one edge type produces a set of independent, disconnected "sticks", with each stick corresponding to sentences in the documents in the corpus, in sequential order. Second, to capture the document linking graph G, sentences in a document D_i are all linked to all sentences in a document D_j if and only if documents i and j are linked in G.

[51] General domain NLP; While this model incorporates an interesting per-token syntatic knowledge distillation procedure, at a per-token level it merely leverages BERT's NSP loss [17].

G.6 Language modelling & Single-task Classification

[73] Protein sequences; Masked language model plus the multi-class classification of to which protein family an input sequence belongs. Uses non-standard whole-sequence embedding procedure (no `[CLS]` token).

[111] General domain NLP; StructBERT includes masked language modelling, a token permutation language modelling task, and an extended version of the NSP/SOP task at a per-sample level.

G.7 Language modelling & Multi-task Classification

[63] General domain NLP; Masked language model plus multi-task classification across various NLP tasks.

[37] Graph data; This model uses a masked imputation task similar to a masked language model and a highly multi-task supervised whole-graph level prediction. On this non-NLP domain, [37] finds that the multi-task whole-graph level task is essential for performance.

[70] EHR Timeseries data; This model uses a masked imputation task similar to a masked language model over time series data and a multi-task supervised whole-sequence prediction task. On this non-NLP domain, [70] finds the multi-task whole-sequence level task essential for performance.

G.8 Language modelling & whole-sample graph-based contrastive objectives

[112] General domain NLP; KEPLER augments traditional MLM on text samples with a constraint ensuring the (per-sample) embeddings of entity descriptions pulled from pre-specified knowledge graphs (KGs) reflect geometric constraints, leveraging the [105] geometric constraints. As we will see in our theoretical analyses, these constraints are much more restrictive on the latent space geometry and thus imply a greater encoding of domain knowledge in the model. Note that JAKET [128] also leverages entity descriptions in its per-token encoding. However, these descriptions are (1) extracted via per-token embeddings, using the first mention of the token, not whole-sample embeddings, and (2) integrated back into the original text in a per-token manner, not optimized over directly via geometric constraints as in KEPLER.

[21] Molecules; CK-GNN designs a pre-training scheme for molecular graphs in which a molecular GNN is trained to produce molecule embeddings that obey the similarity structure of a 1-NN graph in a cluster-limited molecular fingerprint space (using the Dice similarity coefficient). Unlike the NLP approaches, this method has no intra-sample (*i.e.*, per-token, where here "token" refers to individual atoms within the molecular graph) pre-training task.

[40] Multi-lingual NLP; Much like KEPLER, XLM-K augments traditional MLM with two tasks that constrain the geometry of the per-sample latent space via a (now multi-lingual) graph of entity descriptions linked to sentences containing said entities. Like KEPLER, as the graph connections here are defined only for entity descriptions and not all free-text, the latent space regularization is only over a limited slice of the space.

[28] General domain NLP/IR; WebFormer designs a pre-training scheme leveraging the structure of DOM trees in HTML pages to impose multiple per-sample and per-sample/per-token hybrid constraints that encourage individual samples to be (a) close to noised versions of themselves based on reordering or masking and (b) to be close to representations of their parent/child nodes in the DOM tree, thus imposing a structural penalty geometrically. By mixing per-sample and per-token tasks, WebFormer even more closely entangles the per-sample and per-token latent spaces in their model, and this approach bears closer study in other contexts.

G.9 Language modelling & whole-sample augmentation/noising based contrastive objectives

[50] General domain NLP; InfoWord incorporates an objective alongside masked language modelling which pushes the whole-sample embedding of a sentence to have high mutual information with various sub-contexts within that sentence and low mutual information with sub-contexts of other sentences.

[26] General domain NLP; DeCLUTR optimizes for masked language modelling alongside a contrastive objective comparing anchor spans to positive spans chosen from within individual samples, contrasted against spans from other samples. This is considered "whole-sample" rather than a per-token contrastive loss as the embeddings of the spans (which can be quite long) are produced via a canonicalized pooling operation used for sentence embeddings.

[116] General domain NLP; CLEAR optimizes for masked language modelling alongside a contrastive objective powered by per-sentence noising strategies, including word or span deletion, reordering, and synonym substitution.

[71] General domain NLP; COCO-LM builds on other discriminative language modelling variants such as ELECTRA [12] by adding two additional tasks. First, a true language modelling task atop the auxiliary-model-driven corrupted input text. Second, a contrastive objective pushing corrupted sentences towards their un-corrupted originals and those derived from distinct sentences farther apart.

[8] General domain NLP; Semantic re-tuning via contrastive tension adds a pre-training objective onto language model pre-training. This is done to encourage the final per-sample representations of a single sentence embedded via two otherwise independently trained models to be similar and those of different sentences to be distinct.

[22, 127, 126, 80, 134] Networks; KCL,GraphCL, JOAO, MICRO-Graph and GCC use augmentation-based contrastive learning pre-training methods for network datasets. KCL is notable as it is (1) specialized for molecular graphs and (2) uses a knowledge-derived augmentation strategy that constructs a knowledge enriched version of an input molecular graph as its "augmentation policy." MICRO-Graph is also notable as its contrastive objective compares a graph to dynamically clustered "motif" subgraphs from within said graph as positive pairs.

[96] General domain NLP; GLM integrates a per-token KG through traditional entity masking (albeit with an improved selection mechanism) and a per-sample contrastive objective that uses the entity knowledge graph to generate distractor negative samples for the contrastive learning task.

[67] General domain NLP & Computer Vision; CAPT proposes a noising based contrastive learning loss *in substitution for* the masked language modelling loss of BERT. They employ no per-token pre-training task.

[141] Protein Sequences/Structures; GearNet introduces a vehicle for pre-training not over protein sequences, but rather over protein structures, realized as graphs. They combine intra-sample/per-amino-acid tasks, including prediction of masked node features and prediction of geometric relationships between nodes as implied by the protein graphs, and a per-sample noising based contrastive objective.

G.10 Language modelling & multi-modal or multi-lingual contrastive objectives

Note that by viewing multiple data modalities as "augmentations" of the data samples, one can realize these methods (in general) as examples of augmentation-based contrastive learning objectives, such as those used in [25]. However, as these methods are common, we highlight them explicitly here.

[11] General domain NLP; InfoXLM focuses on multi-lingual pre-training, and leverages per-token tasks. This includes multi-lingual masked language modelling and translation language modelling (*i.e.*, variations on a traditional masked language modelling task). It also incorporates a cross-lingual per-sample contrastive objective that aligns the geometry of the

latent spaces across distinct languages. One important nuance is that they use different layer depths to define the latent space for their cross-lingual contrastive objective vs. their per-token objectives, which is not natively describable in our framework. In addition, as each monolingual corpus lacks any rich, independent per-sample task, any individual monolingual latent space cannot be guaranteed to have any rich structural constraints.

G.11 Language modelling alone with relationally-concatenated samples

These methods concatenate samples together before processing them with a pre-training encoder based on inter-sample relations. This is an orthogonal direction to adding greater per-sample dependencies to pre-training methods than our framework but warrants commentary nonetheless.

[85] Protein sequences; MSA transformers extend protein-sequence language models such that they do not take in as input a single sequence but rather an entire multiple-sequence alignment (MSA) profile. These profiles consist of many sequences corresponding to evolutionary homologs of the same protein. This concatenated input is processed via a sparsified form of axial self-attention, which enables cross-attention between the various aligned sequences. They impose no per-sequence tasks by default in this architecture.

[53] General domain NLP; This theoretical analysis shows that transformers cannot model dependencies between sentences that never appear in the same example during pre-training. To combat this, they propose concatenating samples via inter-sample relations (in particular, via a kNN method) at pre-training time, enabling a greater diversity of cross-attention contexts during pre-training vs. fine-tuning. Thus, while they only use language modelling during pre-training, they speculate that their sample-augmentation procedure helps the model better reason about per-sample information through per-token tasks.

[6] General domain NLP; CDLM proposes to concatenate multiple related documents (leveraging categorical information to cluster documents) together into a single sample prior to performing traditional masked language modelling. To limit the model's complexity, attention is restricted to intra-document for unmasked tokens but allowed to be global for masked tokens.

[30] General domain NLP; REALM uses a latent variable model to learn a relevance score between input text spans and documents in an auxiliary document base. The top-k documents, according to this relevance score, are then concatenated to the input prior to solving the masked language modelling task used during pre-training. In this way, the model learns to join relevant documents from an external knowledge base in accordance with which documents would most improve the masked language modelling objective. In addition, by learning this relevance score, the model introduces an implicit whole-sample structural constraint on the latent space according to the unsupervised clustering induced by relevance assignment.

G.12 Autoencoding & Unsupervised Clustering

[54] General domain NLP; MARGE deviates significantly from the norm by not employing any form of language modelling or other forms of a per-token pre-training task. Instead, it employs only a per-sample contextualized autoencoding objective and an unsupervised per-sample retrieval step (to provide context for said autoencoding). While this approach does provide a deeper form of a per-sample structural constraint than many other approaches, it is also implicit and has no mechanism for injecting domain knowledge. MARGE is also tested solely on downstream tasks at the per-sample level, so it is unclear if this method would offer reduced benefits for per-token downstream tasks.

G.13 Methods orthogonal to our framework

[64] KG-BART is a text-generation model that leverages per-token knowledge after a text-encoder to enrich the generated text with information from a textual knowledge graph (in a per-token

manner). It is neither used for general pre-training nor does it leverage any additional per-sample constraints.

[122] Text-based Knowledge Graphs; This work produces embeddings of nodes in KGs by combining transformer-based text encodings with graph convolutional network KG embedding methods, leveraging link prediction as the pre-training task. Entity descriptions / textual features represent the individual nodes. Link prediction can be seen as inducing a geometric constraint via the connectivity of the knowledge graph on whole-sample embeddings. However, given that relationships are used in encoding the data as well, GraphFormer cannot be used in a context where KG links may not be observed at FT time. It should be seen not as a general text PT method but as an advanced KG embedding mechanism, so it does not directly fall under our framework.

[1] KeLM (unrelated to KELM [66]) is a method for converting a free-text KG into textual nodes so language modelling can be used over that corpus and is orthogonal to the methods of pre-training.

[87] This paper is a method for populating a KG from free-text via BERT. It has no bearing on incorporating structure or knowledge into PT and is irrelevant to our framework.

[133] This paper presents a method to drop redundant triples from a knowledge graph and a regularization technique to limit the impact of added irrelevant knowledge to per-token knowledge-enhanced PT methods such as ERNIE [139].

[123] Knowledge Graph Completion; KG-BERT is a method for knowledge graph completion in which textual representations of entities and relations in KGs are embedded by fine-tuning a pre-trained BERT style transformer for link prediction over a given KG. As this is only for knowledge graph completion, it is orthogonal to our study of pre-trained models in general.

[109] Knowledge Graph Completion; Much like KG-BERT, SimKGC is a method for knowledge graph completion that fine-tunes a BERT model via a contrastive loss over a fixed knowledge graph for link prediction. Though their methodology overlaps with ours in that both use variants of contrastive losses and SimKGC explores more complex negative sampling strategies, the two methods are still very different. Ours is focused on general pre-training and uses a single encoder and a unified latent space. In contrast, SimKGC is only examined for KG completion and encodes head and tail entities via separate encoders.

[115] Event Extraction (EE); CLEVE designs a pre-training method specifically for event extraction. Their pre-training method includes a text-encoder which includes a *cross-event* contrastive loss pushing *individual tokens* from the same "event" closer together than those from different events, which bears a surface similarity to our approach. In addition, they add a graph encoder over the semantic structure of events. Their methodology is focused solely on EE, which is orthogonal to our more general PT framework.

[47] General domain NLP and Computer Vision; ViLT is a method for pre-training aligned text-image pairs. It leverages masked language modelling, an image-text matching binary classification objective, and a contrastive objective comparing image and text representations. This multi-modal contrastive objective is very similar (insofar as it relates to our framework) to those works that perform multi-lingual or other multi-modal contrastive methods. In ViLT, however, the transformer architecture processes images and text jointly in a single encoder, so it is not well suited for use on only images or only text. This, combined with its focus on computer vision, renders it orthogonal to our framework.

[57] General domain NLP and Computer Vision; StructuralLM proposes a new method of pre-training for scanned documents that takes advantage of the structure of the document w.r.t. images and text simultaneously. As their focus is on cross-modal pre-training of text and image alignment, it is orthogonal to our work.

[20] General domain NLP and Computer Vision; This paper proposes a framework for simultaneous (and continuous) discovery of edges in a multi-modal knowledge graph and the leveraging of that knowledge graph to inform representation learning. However, it is not suitable for

our framework for two reasons. First, like ViLT, it is focused on image-text alignment pre-training. Second, when producing node (*e.g.*, images or text snippets) representations, it requires connectivity information in the associated multi-modal knowledge graph. In contrast, our methods take as input only elements from $\mathcal{X}$.

[61] Named Entity Linking; SapBERT is a method for aligning the output of a pre-trained language model with a per-token knowledge graph through a metric learning loss applied at a per-sample level but only over entity names (not even entity descriptions). As it applies this as a secondary, post-PT stage, and this method only optimizes for alignment between entity names and a static KG, it is not a general PT framework. It is thus orthogonal to our efforts here.

[68] Information Retrieval; HARP is a method for specializing pre-training towards ad-hoc query information retrieval. They introduce four retrieval-specific pre-training tasks leveraging hyperlinks in Wikipedia articles in addition to traditional masked language modelling. Rather than using the raw text of the hyperlinks or the per-sample representations of text spans containing hyperlinks, both of which are explored in [7], these authors use attention weights to extract various "queries" from the underlying text and match those against possible destination pages via contrastive losses. This, therefore, does not impose a constraint on the latent space over the original pre-training dataset $\mathcal{X}$ (but instead introduces a new latent space consisting of query spans) and is further specialized exclusively for ad-hoc retrieval tasks.

[41] Node Embedding for Heterogeneous Graphs; CPT-HG is a contrastive pre-training framework to embed nodes in a heterogeneous network. Unlike in our setting, where the pre-training graph G_{PT} is *only used as an implicit input to derive the loss function*, in CPT-HG the graph (with entire edge connectivity information) *is* the input to the problem. Thus, node embeddings will rely on connectivity information, which is not permissible in our pre-training context. So, this method is orthogonal to our study here.

[9] Expert Matching; CODE is a method specifically and exclusively designed to discover appropriate experts in an employment/contracting setting and is thus orthogonal to our framework, which is focused on more general pre-training.

G.14 Methods that only change things at FT time

[72] Biomedical domain NLP; MOP does not change anything at PT time but trains sub-KG adapters on entity recognition tasks prior to FT to infuse entity knowledge into the PT method. It is a per-token pre-training method.

[62] General domain NLP; K-BERT, at PT time, is actually equivalent to BERT [17]. However, it does do other interesting things at FT time, including augmenting the sentence flow with injected per-token knowledge graphs and limiting self-attention to only flow along links supported by the original sentence or the injected knowledge. However, as this is only true at FT time, it is equivalent to BERT at PT time.

[86] General domain NLP; This model, at PT time, is equivalent to BERT [17]. Like [62]. However, it specializes in a fine-tuning procedure for sentence information retrieval tasks, similar to how PT is adapted in this framework.

[121] General domain NLP; ConSERT adds an auxiliary specialization stage after pre-training to fine-tune sentence representations. This new stage imposes a SimCLR [10] style data-augmentation/noise-invariance based contrastive learning objective, using adversarial perturbations, token shuffling, token/feature/span erasure, and dropout noising methods.

[138] General domain NLP; IS-BERT does not modify anything from traditional BERT at pre-training time. However, they add a second PT stage to optimize sentence representations alone using an auxiliary feature extractor in the form of various CNNs applied atop BERT token representations. The final sentence representation is trained to maximize mutual information with various sub-contexts within the sentence but low mutual information with

other sentences. In this second pre-training stage, there is no language modelling performed. As this approach only adapts an auxiliary featurizer to produce sentence encodings and is not intended for general transfer learning, it is inappropriate for our framework. A similar work that integrates both components during pre-training, and thus is relevant in our work is [50] and is discussed above.

[66] General domain NLP; KELM does not modify PT objective but instead enhances a model at FT time by injecting per-token knowledge via a GNN module atop the pre-trained LM embeddings via a unified text-entity graph. It is similar to KBERT [62] in this way but resolves other issues with that approach relating to knowledge ambiguity and by supporting multi-hop reasoning, again over the per-token embeddings.

[19] General domain NLP; KI-BERT augments BERT with KG-specific information via joint token-entity embeddings and information fusion but does this only at FT time.

[119] General domain NLP; K-XLNet introduces a secondary FT stage in which knowledge injectors throughout an XL-Net architecture are further trained to leverage knowledge (encoded via free-text entity descriptions) that is injected into input sentences alongside matched tokens. It does not modify the XL-Net PT stage at all.

[110] General domain NLP; K-Adapter proposes to pre-train various knowledge adapters that can be used alongside a pre-trained language model at a fine-tuning time. Thus, while there is a pre-training process for the adapters, this process does not modulate the original pre-trained language model. In addition, both adapters pre-trained in this work are based on per-token knowledge graphs; one leverages concatenated entity embeddings to perform relation classification, and another predicts which token in the sentence is the "head" in a dependency parse tree, so no per-sample constraints are applied.

[78] General domain NLP; E-BERT injects per-token knowledge into BERT by first aligning embeddings of a knowledge graph with the input word piece embedding space of a (fixed, pre-trained) BERT model, then using various strategies to input them alongside their source mentions in FT text. They do no additional pre-training, so this model only affects the model at FT time.

[27] General domain NLP; [27] augment LMPT methods with an additional, pre-FT procedure in which the model is further trained using a supervised, per-sample metric learning task leveraging FT labels directly to form the classes used for metric learning. They do not materially change the task-independent PT procedure, though their FT metric learning procedure does induce some structure at the per-sample level.

[137] QA; GreaseLM is a method for fusing information from knowledge graphs into pre-trained language models. It shares many similarities with methods that do this for pre-training purposes, such as JAKET [128], CokeBERT [99], SMedBERT [135], and Bert-MK [34]. However, unlike these methods, it only employs these techniques at the fine-tuning time, for question answering tasks specifically. As it is not focused on general pre-training, it is outside our scope.

[45] Language modelling; kNN language models improve the text generation powers of language models by augmenting traditional decoding with a nearest-neighbor lookup operation over a text datastore leveraging the embeddings of a token's leftward context by the language model to judge nearest neighbors. However, it involves no additional language model training and can only be applied at the fine-tuning time to aid in text generation, and is thus out of our scope.

[46] Sentence embedding; NT-Xent proposes a secondary specialization stage after pre-training only for generating sentence embeddings. To do this, they employ a contrastive objective contrasting the final CLS embeddings of an updating, specialized BERT model against a pooled aggregate of the per-token embeddings across all layers of the pre-trained BERT model used to initialize the specialized sentence embedding model.

[56, 98, 38] Sentence Embedding; These methods propose to use unsupervised per-sample smoothing operations (a normalizing flow network in [56] and a mean/covariance standardization whitening operation in [98, 38]) on the per-sample embeddings after pre-training in order to produce higher quality per-sample embeddings.

[25] General domain NLP; SimCSE extends traditional MLM by imposing a second pre-training stage for optimizing sentence embeddings. In this stage, SimCSE optimizes the transformer such that the whole-sample embeddings satisfy either a supervised or unsupervised contrastive learning objective. In the supervised case, this is based on labeled sentence pairs according to a Natural Language Inference (NLI) task, with entailment pairs being treated as positives and contradiction pairs as hard negatives. In the unsupervised case, this is based solely on applying multiple dropout masks to the same sentence to generate positive pairs. Any two distinct sentence inputs are treated as negative samples. This extra pre-training stage is applied to a relatively small number of samples (10^6) relative to the entire PT cohort, which may help prevent catastrophic forgetting of the original pre-training objective.

[14] Academic NLP; SPECTER extends traditional language model pre-training by imposing a second pre-training stage for optimizing document embeddings (realized as `[CLS]` token embeddings of concatenated academic paper titles and abstracts). This stage uses a triplet-based geometric loss to ensure that these per-sample embeddings reflect the structure of a pre-specified citation network. This is a form of an explicit, structural constraint; however, they do not ever test fully fine-tuning the SPECTER model in their paper and only compare it against other, frozen pre-trained language models. This is likely to have a significant impact on model comparisons. Similar to SimCSE [25], this extra pre-training stage is applied to a small number of samples (146K documents) to help prevent catastrophic forgetting of the original pre-training objective.

[7] General domain NLP; This paper introduces a second pre-training stage after multi-lingual masked language modelling. In this second stage, hyperlinks in the source text (drawn from Wikipedia) are matched via single-task classification to a curated set of destination URL categories, collapsing all URLs pointing to the same Wikipedia page across languages into one. They do this classification in several ways, including incorporating the per-sample representation of the text span rather than merely the hyperlink token representations themselves (likely motivated by the likelihood of only a single hyperlink being present in the source text). We can realize this task as instances of several other common paradigms: (1) Single-task classification applied to the per-sample representation, (2) link prediction in a graph linking cross-lingual Wikipedia pages together, or (3) as an example of named entity recognition. This second stage is only allowed to modify the last two layers of the transformer architecture, which may be a vehicle to prevent catastrophic forgetting.

[120] Sentiment Analysis; SAKG-BERT augments a pre-trained language model with a sentiment-analysis knowledge graph at the fine-tuning time only by concatenating relevant relationships from the KG based on sentiment-laden terms appearing in the review to the raw input text. They do not otherwise change the pre-training or fine-tuning process.