nature machine intelligence

Article

https://doi.org/10.1038/s42256-023-00647-z

Structure-inducing pre-training

Received: 3 August 2022

Accepted: 23 March 2023

Published online: 1 June 2023

Check for updates

Matthew B. A. McDermott[1,2], Brendan Yap[1], Peter Szolovits[1] & Marinka Zitnik[2,3,4]

Language model pre-training and the derived general-purpose methods have reshaped machine learning research. However, there remains considerable uncertainty regarding why pre-training improves the performance of downstream tasks. This challenge is pronounced when using language model pre-training in domains outside of natural language. Here we investigate this problem by analysing how pre-training methods impose relational structure in induced per-sample latent spaces—that is, what constraints do pre-training methods impose on the distance or geometry between the pre-trained embeddings of samples. A comprehensive review of pre-training methods reveals that this question remains open, despite theoretical analyses showing the importance of understanding this form of induced structure. Based on this review, we introduce a pre-training framework that enables a granular and comprehensive understanding of how relational structure can be induced. We present a theoretical analysis of the framework from the first principles and establish a connection between the relational inductive bias of pre-training and fine-tuning performance. Empirical studies spanning three data modalities and ten fine-tuning tasks confirm theoretical analyses, inform the design of novel pre-training methods and establish consistent improvements over a compelling suite of methods.

The pre-training (PT)/fine-tuning (FT) learning paradigm (also known as transfer learning) has had a tremendous impact on natural language processing (NLP) and related domains[1–3]. PT/FT methods have produced models capable of providing free-text answers to natural language questions[4], predicting properties of proteins from sequences[5] and enabling reaction synthesis prediction from molecular simplified molecular-input line-entry system (SMILES) strings[6], among other advancements.

In NLP or NLP-derived PT/FT, for a given pre-training data modality $\mathcal{X}$, we are given a dataset $\mathbf{X} \in \mathcal{X}^{N_{\text{PT}}}$ of size $N_{\text{PT}} \in \mathbb{Z}$ and pre-train an encoder $f_{\vec{\theta}} : \mathcal{X} \to \mathcal{Z}$ parametrized by $\vec{\theta}$, which maps $\mathcal{X}$ into a latent space $\mathcal{Z}$. This encoder $f_{\vec{\theta}}$ is then transferred for use in various FT tasks (which are not known during PT). We evaluate PT/FT systems via the performance of $f_{\vec{\theta}}$ on said FT tasks.

In this Article, we are concerned primarily with the efficacy of PT/FT for downstream tasks that operate at a per-sample level. For example, in NLP, evaluating the sentiment of a full restaurant review is a per-sample task, in contrast to identifying a named entity token within a sentence, which is an intra-sample, per-token task. One aspect of PT that drives such eventual FT performance is the induced geometry of the pre-trained, per-sample latent space $\mathcal{Z}$ (formally defined in Methods). For example, it is well documented that the sentence embeddings produced by pre-trained language models in NLP can be non-smooth and anisotropic, which harms downstream task performance[7] (note that our use of the term language model refers to methods designed to produce embeddings or enable FT off of pre-trained language models, not to autoregressive language models for generation). In other domains, such as biomedical modalities, where per-sample tasks are even more prevalent than intra-sample tasks compared with NLP, the importance of this geometry only increases. Despite this importance, research into mechanisms to induce explicit, deep structural constraints in $\mathcal{Z}$ is limited. For example, many methods ignore the geometry

[1]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA. [2]Department of Biomedical Informatics, Harvard Medical School, Boston, MA, USA. [3]Broad Institute of MIT and Harvard, Cambridge, MA, USA. [4]Harvard Data Science Initiative, Cambridge, MA, USA. e-mail: marinka@hms.harvard.edu

of $\mathcal{Z}$ by imposing no PT loss over the whole-sample embeddings[3,8,9]. Other methods impose either only shallow constraints, such as through an auxiliary classification PT objective[1,10,11], or deeper structural constraints, but in an implicit manner, such as through data augmentation-based[12–17] or noising-based[18,19] contrastive losses. While such methods can be powerful and have been successful in many areas, we argue that the lack of a clear framework to design PT methods that impose structural constraints on $\mathcal{Z}$ that are simultaneously explicit (similar to supervised classification losses) and deep (similar to noising-based and augmentation-based contrastive losses) is a substantial weakness.

On the basis of this observation, we develop a framework under which the PT objective is subdivided into two components: first, a language model imputation or denoising objective that leverages intra-sample relationships, and second, a loss term driven to regularize the geometry of the per-sample latent space $\mathcal{Z}$ to reflect the connectivity patterns of a user-specified graph G_{PT}. By relying on graphs to capture the structure we wish to induce in $\mathcal{Z}$, this framework allows us to specify PT methods that induce deep structure in an explicit manner, filling exactly the gap identified above. In addition, this paradigm can capture diverse relationships, such as those motivated by external knowledge (for example, ref. 20), self-supervised constraints (for example, refs. 21,22) or distances between samples in an alternative modality (for example, ref. 23). Moreover, this PT framework is simultaneously specific to allow us to make theoretical guarantees about how different PT graphs impact FT performance, general enough to encompass a variety of PT methods and sufficiently expressive to motivate new PT methods that have not been previously studied. In addition to theoretical analysis, we demonstrate empirically that defining new methods according to our framework, using explicit forms of real-world structure, yields significant benefits over competitive PT baselines across three modalities and ten FT tasks.

Our work advances PT/FT research through three contributions. First, through a comprehensive review and detailed commentary, we show that existing PT methods do not induce structural constraints over $\mathcal{Z}$ that are simultaneously deep and explicit. Second, we establish a framework for describing PT methods, which provides a mechanism to design PT methods that explicitly induce deep structural constraints in $\mathcal{Z}$ by a user-specified PT graph G_{PT}. We further support this framework with theoretical results quantifying how the graph's structure relates to FT task performance. Crucially, this formalization in our new PT paradigm offers insight into when PT does or does not add value over supervised learning alone. Third, we show that structure-inducing PT methods through our framework perform at or above the level of existing PT methods across three data modalities and ten FT tasks.

Results

General PT problem formulation

Given a dataset $\mathbf{X}_{\text{PT}} \in \mathcal{X}^{N_{\text{PT}}}$, a PT method aims to learn an encoder $f_\theta : \mathcal{X} \to \mathcal{Z}$ such that f_θ can be transferred to FT tasks that are unknown at PT time. While we can leverage additional information at PT time to inform the training of f_θ (for example, PT-specific labels $\mathbf{Y}_{\text{PT}}$), the encoder f_θ must take only samples from $\mathcal{X}$ as inputs so that it can be used for FT. PT methods typically solve this problem by training f_θ to minimize a PT loss $\mathcal{L}_{\text{PT}}$ over $\mathbf{X}_{\text{PT}}$. For example, in the model Bidirectional Encoder Representations from Transformers (BERT), $\mathcal{X}$ consists of free-text samples, f_θ is a transformer model and $\mathcal{L}_{\text{PT}}$ consists of both a masked language modelling per-token loss and the next-sentence-prediction (NSP) per-sample loss[1].

Our definition of PT ignores secondary applications of the PT objective; for example, autoregressive language models (for example, Generative Pre-trained Transformer (GPT)-3 (ref. [3])) are often used for their generative use directly and not as commonly used to acquire embeddings or in transfer learning. Therefore, we are primarily interested in PT methods derived from NLP PT methods. This area is of particular interest because methods have been successful within NLP[1,3,24], have motivated a large number of derived methods in non-language, biomedical modalities[25–28] and are not yet fully technically understood[7,29,30].

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 through the following definitions.

Definition 1 explicit versus implicit structural constraints. *A PT objective $\mathcal{L}_{\text{PT}}$ imposes a structural constraint that is explicit (versus implicit) to the degree that it (as f_θ approaches optimality) permits us to reason directly about the relationship (in particular, the distance) between any two samples $\mathbf{z}_i$ and $\mathbf{z}_j$ in the latent space $\mathcal{Z}$, where subscripts i and j are merely used to differentiate between these two samples in $\mathcal{Z}$.*

Definition 2 deep versus shallow structural constraints. *A PT objective $\mathcal{L}_{\text{PT}}$ imposes a structural constraint that is deep (versus shallow) based on how much information (for example, how many dimensions) would be required to fully satisfy the constraint.*

For example, consider a classification PT loss with labels in the set $\mathcal{Y}$, with sample i having label $y_i \in \mathcal{Y}$, and using a logit layer that maps the induced representation of sample i to a predicted score: $\mathbf{z}_i \mapsto \tilde{y_i}$. This method produces an explicit structural constraint because, near optimality, we can infer that the relative (cosine) distance between two samples $\mathbf{z}_i$ and $\mathbf{z}_j$ 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 $\mathbf{p}_c \in \mathcal{Z}$, then compress all samples $\mathbf{z}_i$ near their class prototype $\mathbf{p}_{y_i}$. Moreover, this distance-based constraint can be accomplished in a very-low-dimensional space $\mathcal{Z}$ (for example, we can distribute each $\mathbf{p}_c$ uniformly about a two-dimensional unit circle, then compress all $\mathbf{z}_i$ to appear at a minimal cosine distance from their class prototypes), illustrating that this constraint is very shallow.

In contrast, consider a contrastive method that asserts that $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$ should be close to $\mathbf{z}'_i = f_\theta(\tilde{\mathbf{x}}_i)$, where $\tilde{\vec{x}}_i$ is a perturbed version of $\vec{x}_i$ under some noising or augmentation procedure $\mathbf{x}_i \mapsto \tilde{\mathbf{x}}_i$, but simultaneously far from other samples $\mathbf{z}_j$. 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 $\mathbf{z}_i$ and $\mathbf{z}_j$ is constrained. However, it imposes a deeper constraint than the classification objective because the implicit connections between samples induced by the noising procedure reflect relationships that cannot necessarily be captured in a low-dimensional space (dependent on dataset size and density).

Existing PT method constraints

To show that existing methods broadly do not provide means to impose structural constraints that are simultaneously deep and explicit, we survey over 90 existing PT methods based on how their objective functions constrain the $\mathcal{Z}$ (Extended Data Fig. 1 and Supplementary Information). For full details on our review findings, see Methods. Throughout all examined methods, we find that deep, explicit structural constraints are rarely employed. Instead, most methods either (1) impose no per-sample PT objectives at all (for example, text-generation models, which are often not used for embeddings at all but rather for prompting or generative applications[3,8,9,31]), (2) use explicit, but shallow, supervised PT objectives (for example, BERT's NSP objective, A Lite BERT's (ALBERT's) sentence-order prediction (SOP) objective or various multi-task objectives[1,10,11]), or (3) use implicit, but deep, unsupervised or self-supervised contrastive PT objectives (for example, contrastive sentence embedding los ses[12,13,18,19,32] or other noising-based or augmentation-based approaches[14–17]).

https://doi.org/10.1038/s42256-023-00647-z

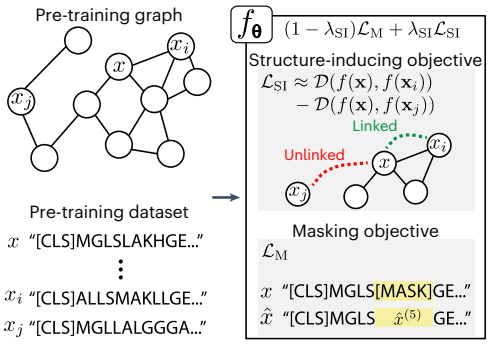


Fig. 1 | Our PT framework. We re-cast the PT formulation by taking a PT graph G_{PT} as an auxiliary input. G_{PT} is used to define a structure-inducing objective $\mathcal{L}_{SI}$, which pushes a PT encoder f_θ to embed samples such that samples are close in the latent space if and only if they are linked in G_{PT}.

Across all surveyed methods, we find that only four methods impose simultaneously explicit and deep constraints: Knowledge Embedding and Pre-trained LanguagE Representation (KEPLER)[33], Contrastive Knowledge-aware GNN (CK-GNN)[23], XLM-K[34] and WebFormer[35]. All four can be described as some form of per-sample graph alignment, in which an external, PT knowledge graph G_{PT} or connectivity algorithm is employed over a subset of PT samples, and the output embeddings of pairs of samples $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$ and $\mathbf{z}_j = f_\theta(\mathbf{x}_j)$ are constrained to reflect their relationships in the PT graph. This form of constraint is explicit, as the graph G_{PT} contains explicit 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.

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, no constraints are implied on the general domain free-text samples in $\mathcal{X}$ alone[33,34]. In CK-GNN, the graph connectivity is derived from a cluster-restricted one-nearest-neighbour graph in an alternative modality's distance space, which may offer a limited higher-order structure. Unlike the NLP approaches, this method has no intra-sample (for example, per-token) PT task[23]. 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. Furthermore, WebFormer is a specialized model specifically for processing web content (text and HTML elements), so this approach cannot be directly generalized to other domains[35]. Moreover, these methods explore only the particular contexts of their models. They offer no general framework for realizing these deep, explicit per-sample constraints in other contexts and do not explore any theory on how these constraints relate to performance for FT tasks[23,33–35].

Overall, our review of PT methods establishes unequivocally that PT 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 consensus on how to constrain the $\mathcal{Z}$ explicitly and deeply. These findings motivate our framework, which offers insight into realizing deep, explicit structural constraints in PT models across diverse contexts and provides theoretical guidance on how structural constraints relate to FT performance. As we show in our results, inducing deep, explicit constraints through our framework will induce significant benefits over existing PT methodologies across three diverse biomedical domains.

Structure-inducing PT

Our PT problem framework includes two small but important differences from the standard formulation (Fig. 1).

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

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

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

$\mathcal{L}_M$ is a traditional, intra-sample objective (for example, a language model), and $\mathcal{L}_{SI}$ is a new, structure-inducing objective designed to regularize the per-sample latent space geometry by the relationships (edges) in G_{PT}. Under our framework, $\mathcal{L}_{SI}$ is only allowable for G_{PT}, f_θ and $\mathcal{Z}$ if it permits some stable optima at which point a radius nearest-neighbour connectivity algorithm under some distance function in $\mathcal{Z}$ will recover G_{PT} (formal constraint is in Methods). Note that this constraint strikes a connection between our framework and the wealth of existing research focused on graph representation learning[36–41]. 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 modelling 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 PT setting where f_θ must take in only inputs from $\mathcal{X}$ directly.

As the loss term added $\mathcal{L}_{SI}$ is explicitly designed to induce the structure of G_{PT} in $\mathcal{Z}$, we call methods (in particular methods leveraging deep, explicit structural constraints) trained under our framework structure-inducing pre-training (SIPT) methods. Many existing PT approaches can be re-realized as SIPT methods, including classification-based PT objectives such as NSP or SOP, contrastive methods, or existing graph alignment methods (Methods). Although SIPT is designed to make it easier to induce deep, explicit structural constraints, it is also flexible enough to capture implicit or shallow structural constraints.

Theoretical analyses

Under our framework, one can link the structure of the PT graph G_{PT} to eventual FT task performance. In particular, as an SIPT embedder f over graph G_{PT} approaches optimality under the loss $\mathcal{L}_{SI}$, it produces an embedding space such that nearest-neighbour performance for any downstream task is lower bounded by the performance that could be obtained via the nearest-neighbour algorithm over graph G_{PT} (Theorem 1). This fact directly connects the geometry of the graph G_{PT} with the eventual FT performance of an 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 $\mathbf{X}_{PT}$ be a PT dataset, let G_{PT} be a PT graph and let f_{θ^*} be an encoder pre-trained under a PT objective permissible under our framing that realizes an $\mathcal{L}_{SI}$ value no more than ℓ^*. Then, under embedder f, the nearest-neighbour accuracy for an FT task y converges as dataset size increases to at least the local consistency* (Supplementary Definition 3) *of y over G_{PT}.*

We establish two corollaries of Theorem 1 that illustrate the importance of choosing graphs G_{PT} that impose deep structural constraints.

Table 1 | A summary of our datasets, tasks and benchmarks

	Proteins	Abstracts	Networks
Data modality ($\mathbf{x}_i$ is a…)	Protein sequence	Biomedical paper abstract	Protein–protein interaction network ego-graph
PT dataset	Tree of life[20]	Microsoft Academic Graph[21,22]	Ref. [26]
$(\mathbf{x}_i, \mathbf{x}_j) \in G_{PT}$	$\mathbf{x}_i$ interacts with $\mathbf{x}_j$	$\mathbf{x}_i$'s paper cites $\mathbf{x}_j$'s paper	$\mathbf{x}_i$'s central protein agrees on all but nine Gene Ontology labels with $\mathbf{x}_j$'s central protein.
Per-token baseline	TAPE[5]	SciBERT[53]	Attribute masking[26]
Per-sample baseline	PLUS[52]	BioLinkBERT[56]	Multi-task learning[26]
FT dataset	TAPE[5]	SciBERT[53]	Ref. [26]

For example, for the Proteins domain, our PT dataset is the set of protein sequences contained in the tree-of-life dataset[20], proteins are linked in our PT graph GPT if and only if they interact according to the tree-of-life graph. In addition, we compare the FT tasks in the TAPE benchmark against the raw, per-token baseline publicly available in the TAPE model[5] and the per-sample baseline published in the PLUS PT model[52].

Table 2 | Mean (± standard deviation) relative reduction of error (defined to be ([baseline error]−[G_{PT} model error])/[baseline error]) of models trained under our framework versus published per-token or per-sample baselines

Domain	Task	Versus per-token PT		Versus per-sample	
		Relative reduction of error	Δ	Relative reduction of error	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%**[a]	↑	NA	
Abstracts	PF	0.3%±0.2	~	**0.8%±0.3**	↑
	SC	2.4%±4.1	~	−1.1%±5.5	~
	AA	**17.7%±6.5**	↑	11.6%±16.2	~
	SRE	**6.7%±0.4**	↑	−3.6%±10.1	~
Networks		7.8%±5.2	~	**5.1%±2.7**	↑

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 (↑ and bolded), no significant change (~) or a statistically significant decrease (↓ and bolded). Statistical significance is assessed via a *t*-test at significance level $P < 0.1$. [a]Per-sample analysis and variance estimates for CP were infeasible due to the computational cost of this task. FT tasks are described in Table 3.

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

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

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

Then, as the 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-neighbour accuracy of an SIPT embedder) will likewise tend to one.*

Informally, these corollaries establish that when a shallow structural constraint is used (for example, 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-neighbour graph over an arbitrary manifold $\mathcal{M}$, then an SIPT model permits a theoretical guarantee for FT performance that approaches unity as the PT dataset size grows for any FT task that is smooth over $\mathcal{M}$.

This theoretical analysis shows that we can directly connect the structure induced in $\mathcal{Z}$ to downstream FT performance. As such, new PT methods that leverage graphs G_{PT} with deeper structural constraints can markedly improve performance, as we will demonstrate on real-world datasets in our experiments. Complete proofs for all theoretical results and semi-synthetic experiments validating our theoretical findings in practice are in Methods.

Datasets and tasks

We examine three data modalities for our experiments: 'Proteins', containing protein sequences; 'Abstracts', containing free-text biomedical abstracts; and 'Networks', containing subgraphs of protein–protein interaction (PPI) networks.

In each data modality, we use different PT datasets and leverage different kinds of PT graphs G_{PT}, test on publicly available benchmarks for FT tasks and compare our SIPT methods with compelling baselines spanning both per-sample and per-token methods (Tables 1–3). Further details on these aspects are in Methods.

$\mathcal{L}_{SI}$ and training procedures

As discussed in the definition of our framework, an SIPT method differs from a standard PT method by (1) the choice of graph G_{PT} (Table 1) and (2) the design of the structure-inducing loss $\mathcal{L}_{SI}$. To define $\mathcal{L}_{SI}$ in our experiments, we leverage ideas from structure-preserving metric learning[42–44]. Structure-preserving metric learning 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[45] and a multi-similarity loss[46], from supervised metric learning to the graph-based, structure-preserving context of $\mathcal{L}_{SI}$ terms in SIPT.

In addition to these losses, in the Abstracts and Proteins domains, we use a warm-start procedure to initialize PT 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 introducing our $\mathcal{L}_{SI}$ term. Second, we 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 set-up, including formal statements of our contrastive and multi-similarity losses, are in Methods. Note that, as is standard in PT applications, for each PT algorithm and data modality, we pre-train a single model on

Table 3 | FT tasks

FT dataset	FT task		Description	Metric
	Name	Abbreviation		
TAPE[5]	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 three-dimensional conformation	Precision @ $L/5$
SciBERT[53]	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 relation extraction	SRE	Per-sentence relation extraction	Macro-F1
Networks[26]			Multi-label binary classification into 40 Gene Ontology terms	Macro-AUROC

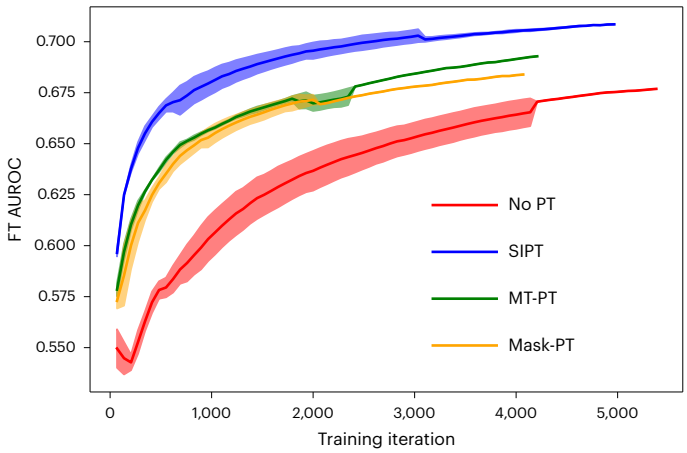


Fig. 2 | FT performance over Networks. Mean ± standard deviation FT AUROC as a function of FT iteration for the Networks dataset. Differences in variance scale result from different runs triggering early stop at different iterations. The SIPT method converges faster and performs better than intra-sample (masked node modelling) or per-sample (multi-task classification) PT. MT-PT indicates using traditional, supervised, multi-task pre-training alone. Mask-PT represents performing mask-imputation pre-training alone, whereas SIPT indicates the combination of the two approaches through our SIPT framework.

the PT dataset, then fine-tune that one pre-trained model on each FT task independently; in other words, in no setting do we need to pre-train a separate model per FT task.

SIPT matches or outperforms all baselines

To analyse our experiments, we compute the relative reduction of error of the best-performing SIPT model versus the per-token or per-sample baselines across all FT tasks (Table 2). In 10 out of 15 cases, SIPT improves over existing methods; in no case does it do worse than either baseline. In some cases, the gains in performance are significant, with improvements of approximately 17% (0.05 macro-F1 raw change) on ACL-ARC (AA), 6% on SciERC relation extraction (SRE) (0.01 macro-F1 absolute change) and 4% on remote homology (RH; 2% absolute accuracy change). SIPT models further establish a new state-of-the-art performance on AA and RH and match state-of-the-art performance on fluorescence (FL), stability (ST) and paper field (PF). See Table 3 and Supplementary Information for details on these tasks, and recall that the F1 metric is the harmonic mean of precision and recall.

Figure 2 shows how performance evolves over FT iterations for the Networks dataset to determine whether the improvements observed at the final converged values are present throughout training. We see that SIPT methods converge faster to better performance than both baselines. Raw results across all settings are presented in Extended Data Tables 3 and 4.

SIPT performance gains are robust

SIPT performance gains persist across all three data modalities and all different G_{PT} types. This shows that explicitly regularizing the per-sample latent space geometry offers value across NLP, non-language sequences and non-sequential domains. Furthermore, leveraging graphs, including those defined by external knowledge, by self-supervised signals in the data directly, and by nearest-neighbour methods over multi-task label spaces, is beneficial. Furthermore, these improvements exist compared with standard language modelling approaches and against existing methods that impose per-sample PT objectives, including single- and multi-task classification objectives.

Gains are attributable to SIPT loss $\mathcal{L}_{SI}$

As outlined in Methods, our experimental design permits us to determine how much of the observed gains in Table 2 are due to the SIPT 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 structure-inducing components introduced by our framework. Full ablation study results can be found in Extended Data Tables 3 and 4.

Discussion

Despite the breadth of research into PT methods, methods for imposing explicit and deep structural constraints over the per-sample, PT

latent space $\mathcal{Z}$ are under-explored (Extended Data Fig. 1). Our theoretical and empirical analyses show that this deficit matters. In particular, we define a PT framework, SIPT, under which the PT loss is subdivided into two components: one that is designed to capture intra-sample (for example, per-token) relationships and one that is intended to constrain the per-sample latent space to capture relationships between samples given by a user-specified PT graph G_{PT}. Under our framework, we show theoretically and via experiments that the structure induced in $\mathcal{Z}$ can be directly connected to eventual FT performance. Empirically, we show that SIPT methods leveraging a variety of PT graphs can consistently outperform 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 PT graphs—for example, 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? For example, 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-neighbour relationships—for example, by leveraging higher-order topological considerations or by matching a distance function rather than a discrete graph? In addition, further exploring the structure-inducing objective's impact on the underlying models' internal mechanisms, as explored via explainable artificial intelligence techniques, would be an exciting avenue for future work. We anticipate that further analyses of these and other questions will lead to new PT methods and enable PT to be successful across diverse domains.

Methods

Structure-inducing losses

We use a multi-similarity loss[46], parameterized by positive pair weight, w_+, negative pair weight, w_-, and fixed hyperparameter, t, given below:

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

We also leverage a contrastive loss modelled after the version in ref. 45. For this loss, we assume we are given the following mappings: 'pos', which maps $\mathbf{x}$ into a positive node (that is, linked to $\mathbf{x}$ in G_{PT}), and 'neg', which maps $\mathbf{x}$ into a negative node (that is, not linked to $\mathbf{x}$ in G_{PT}). The union of a seed minibatch B of points $\mathbf{X}_{\text{B}}$ 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:

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

The Proteins dataset and FT tasks

We use a dataset of ~1.5 million protein sequences from the Stanford tree-of-life dataset[20] (https://snap.stanford.edu/tree-of-life/data.html). The associated GitHub repository for this resource lists a Massachusetts Institute of Technology (MIT) license.

Two proteins are linked in G_{PT} for this dataset 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.

For FT, we use the Tasks Assessing Protein Embeddings (TAPE) FT benchmark tasks[5], 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) and 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 three-dimensional conformation (metric: precision at $L/5$ where L is protein length). All of these tasks are from publicly available datasets that can be obtained directly on TAPE's GitHub (https://github.com/songlab-cal/tape#data), which lists no licences for these datasets though the overall GitHub is released under a BSD 3-Clause 'New' or 'Revised' License. RH tasks a model to predict a protein fold category at a per-sequence level. This task's dataset contains 12,312/736/718 train/validation/test proteins and is originally sourced from ref. 47. SS 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/validation/test proteins and is sourced from ref. 48. ST 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/validation/test proteins, originally sourced from ref. 49. FL requires a model to predict how brightly a protein will fluoresce. This task's dataset contains 21,446/5,362/27,217 train/validation/test proteins and is originally sourced from ref. 50. Finally, CP requires a model to predict whether any given pair of amino acids from a protein are less than 8 Å apart or not. This task's dataset is sourced from ProteinNet[51].

In these experiments, we compare against the published TAPE model[5], which uses a language modeling task alone as our per-token comparison point, and the Protein sequence representations Learned Using Structural information (PLUS)[52] model, which optimizes for LM and supervised classification jointly, for our per-sample comparison point.

The Abstracts dataset and FT tasks

We use a dataset of ~650,000 free-text scientific article abstracts from the Microsoft Academic Graph (MAG) dataset[21,22]. The Abstracts PT data (the MAG dataset) is licensed with an Open Data Commons Attribution License (ODC-By) v1.0 license.

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

Here, we use a subset of the FT tasks used in the SciBERT paper[53], 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 and AA tasks both predict an 'intent' label for citations, and SRE is a relation extraction task. All FT datasets can be obtained from the SciBERT GitHub (https://github.com/allenai/scibert), which lists no dataset-specific licences but is released with an Apache-2.0 license. The PF task 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/validation/test sentences. Although the original dataset is derived from the MAG[21], it was formulated into this task format by SciBERT directly[53]. The SC task challenges 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/validation/test sentences and is originally sourced from ref. 54. The AA task requires 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/validation/test sentences and is originally sourced from ref. 55.

We compare against the published SciBERT model[53] as our per-token comparison and the BioLinkBERT model[56] as our per-sample comparison. BioLinkBERT augments language modelling with a classification task to predict whether the input text consists of two sentences from the same document, linked documents (where linkage

is determined via a citation graph) or unlinked documents. In this way, it uses similar information as used to build our PT graph but via a single-task classification loss rather than the more general structure-inducing losses we use here. Recently, more successful base language models have been proposed beyond the SciBERT model (such as PubMedBERT[57]) and switching to using those to initialize our SIPT models in the warm-start procedures would probably further improve performance across all models. However, given the computational expense of model PT, we retain the use of SciBERT for our initialization model (and accordingly for our corresponding per-token baseline) and leave the investigation of PubMedBERT for future work.

The Networks dataset and FT tasks

We use a dataset of ~70,000 PPI ego networks here, sourced from ref. 26. Each sample here describes a single protein, realized as a biological network (that is, an attributed graph) corresponding to the ego network about that protein (that is, 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.

This dataset is labelled with the presence or absence of any of 4,000 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 nine. This is an alternative-representation nearest-neighbour graph.

We study only one FT task in this setting, which is the multi-label binary classification of the 40 Gene Ontology term annotations (metric: macro area under the receiver operating characteristic curve (AUROC)) used in ref. 26. We use the PT set for FT training and evaluate the model on a held-out random 10% split.

We compare against both attribute-masking[26] and multi-task supervised PT.

Experimental set-up

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 PT for only a small number of epochs under the SIPT loss subdivision. We assess both multi-similarity and contrastive $\mathcal{L}_{SI}$ 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.

Ablation analyses

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 $\lambda_{SI} = 0$, 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 relevant datasets. For the Networks dataset, no other 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.

Selection of λ_{SI} model parameter

For the Proteins and Abstracts datasets, 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 $G_{PT} = (V, E)$. 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 label ranking average precision, normalized discounted cumulative gain, average precision and mean reciprocal rank, 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 FT task and can be determined solely based on the PT data. The final results for these experiments are shown in Extended Data Table 5 for the proteins dataset and Extended Data Table 6 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.

Model architecture and other model parameters

The architectures of our encoders for the Proteins and Abstracts domains are entirely determined from our source models in TAPE[5] and SciBERT[53]. In particular, for proteins and scientific articles, we use a 12-layer transformer with a hidden size of 768, an intermediate size of 3,072 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 as the prediction head, taking as input the output of the final layer's [CLS] token as a whole-sequence embedding. We also tested either PT for a single or four additional epochs based on validation set performance. We 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[26] for the mask model runs, save that for computational efficiency, scale the batch size up as high as possible, then proportionally scale up the learning rate to account for the larger batch size. This corresponds to a batch size of 1,024, a learning rate of 0.01, a graph convolutional neural network (GCNN) with a Graph Isomorphism Network (GIN) encoder, embedding dimensions of 300, 5 layers, 10% dropout, mean pooling and a node feature combination strategy (JK) of 'last'.

FT 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 ref. 58, although 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 of epochs without observing improvement in the validation set performance) was employed, and batch size was set as large as possible considering 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 5×10^{-6} and 5×10^{-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[26] 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 Extended Data Table 1 for proteins and Extended Data Table 2 for scientific articles.

Implementation and compute environments

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 various NVIDIA graphics processing units. Proteins and Abstracts PT runs were performed on a Power 9 system, each run using 4 NVIDIA 32 GB V100 graphics processing units with InfiniBand at half precision.

Systematic review of PT methods

Papers were selected via a manual search of the NLP and NLP-derived

PT methods (that is, methods focused primarily on other domains or multi-modal domains were excluded) via Google Scholar and by crawling through references of papers already included. Citation counts for each work were obtained via Google Scholar on 2 August 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 publication or (2) the first submission date to the arXiv or bioRxiv platforms, as referenced via an exact title match. A manual review was done to classify how PT 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 74 were suitable for inclusion in numerical review results (Extended Data Fig. 1). Supplementary Information summarizes and categorizes all methods considered (and reasons for exclusions are given). Note that our framework focuses on NLP-derived PT methods, but we do not examine generative PT methodology focused on high-dimensional continuous distributions, such as diffusion models[59]. However, these methods have succeeded in other domains, such as computer vision.

Data availability

Our synthetic datasets and pointers to real-world datasets are publicly available at https://github.com/mmcdermott/structure_inducing_pre-training.

Code availability

Python implementation of the methodology developed and used in the study is available via the project website at https://zitniklab.hms.harvard.edu/projects/SIPT. The code to reproduce results, documentation, and usage examples are at https://github.com/mmcdermott/structure_inducing_pre-training.

References

1. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:Human Language Technologies, Volume 1 (Long and Short Papers)* 4171–4186 (Association for Computational Linguistics, 2019).
2. Deng, J. et al. Imagenet: a large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* 248–255 (IEEE, 2009).
3. Brown, T. B. et al. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* **33**,1877–1901 (NIPS, 2020).
4. Sanh, V. et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations* (2022).
5. Rao, R. et al. Evaluating protein transfer learning with TAPE. In *Advances in Neural Information Processing Systems Vol 32* (eds Wallach, H. et al.) (Curran Associates, 2019).
6. Schwaller, P., Hoover, B., Reymond, Jean-Louis, Strobelt, H. & Laino, T. Extraction of organic chemistry grammar from unsupervised learning of chemical reactions. *Sci. Adv.* **7**, eabe4166 (2021).
7. Li, B. et al. On the sentence embeddings from pre-trained language models. In *Proc. 2020 Conference on Empirical Methods in Natural Language Processing* 9119–9130 (Association for Computational Linguistics, 2020).
8. Liu, Y. et al. RoBERTa: a robustly optimized BERT pretraining approach. Preprint at https://arxiv.org/abs/1907.11692 (2019).
9. Radford, A., Narasimhan, K., Salimans, T. & Sutskever, I. Improving language understanding by generative pre-training. (2018).
10. Lan, Z. et al. ALBERT: a lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations* (ICLR, 2019).
11. Liu, X., He, P., Chen, W. & Gao, J. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (eds Korhonen, A. et al.) 4487–4496 (ACL, 2019).
12. Giorgi, J., Nitski, O., Wang, B. & Bader, G. DeCLUTR: deep contrastive learning for unsupervised textual representations. In *Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* Vol. 1, 879–895 (Association for Computational Linguistics, 2021).
13. Kong, L. et al. A mutual information maximization perspective of language representation learning. In *International Conference on Learning Representations* (2020).
14. Tetko, I. V., Karpov, P., Bruno, E., Kimber, T. B. & Godin, G. Augmentation is what you need! In *International Conference on Artificial Neural Networks* 831–835 (Springer, 2019).
15. Karpov, P., Godin, G. & Tetko, I. V. Transformer-CNN: Swiss knife for QSAR modeling and interpretation. *J. Cheminform.* **12**, 1–12 (2020).
16. Tetko, I. V., Karpov, P., Van Deursen, R. & Godin, G. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nat. Commun.* **11**, 1–11 (2020).
17. Schwaller, P. et al. Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS Cent. Sci.* **5**, 1572–1583 (2019).
18. Wu, Z. et al. CLEAR: contrastive learning for sentence representation. Preprint at https://arxiv.org/abs/2012.15466 (2020).
19. Meng, Y. et al. COCO-LM: correcting and contrasting text sequences for language model pretraining. In *Adv. Neural Inf. Process. Syst.* (eds Ranzato, M. et al.) **34**, 23102–23114 (Curran Associates, 2021).
20. Zitnik, M., Sosič, R., Feldman, M. W. & Leskovec, J. Evolution of resilience in protein interactomes across the tree of life. *Proc. Natl Acad. Sci. USA* **116**, 4426–4433 (2019).
21. Wang, K. et al. A review of microsoft academic services for science of science studies. *Front. Big Data* **2** (2019).
22. Hu, W. et al. Open graph benchmark: datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems* **33**, 22118–22133 (NEURIPS, 2020).
23. Fang, Y. et al. Knowledge-aware contrastive molecular graph learning. Preprint at https://arxiv.org/abs/2103.13047 (2021).
24. Sanh, V. et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations* (2021).
25. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl Acad. Sci. USA* **118**, e2016239118 (2021).
26. Hu, W. et al. Strategies for pre-training graph neural networks. In *ICLR* (2020).
27. McDermott, M. B. A. et al. A comprehensive ehr timeseries pre-training benchmark. In *Proceedings of the Conference on Health, Inference, and Learning, CHIL '21* 257–278 (ACM, 2021).
28. Rao, R. M. et al. MSA transformer. In *Proc. 38th International Conference on Machine Learning, Proc. Machine Learning Research* Vol. 139 (eds Meila, M. & Zhang, T.) 8844–8856 (PMLR, 2021).
29. Saunshi, N., Plevrakis, O., Arora, S., Khodak, M. & Khandeparkar, H. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of the 36th International Conference on Machine Learning, volume 97* (eds Chaudhuri, K. & Salakhutdinov, R.) 5628–5637 (PMLR, 2019).
30. Levine, Y. et al. The inductive bias of in-context learning: rethinking pretraining example design. In *International Conference on Learning Representations* (2022).

31. Radford, A. et al. Language models are unsupervised multitask learners. *OpenAI* **1**, 9 (2019).
32. Ribeiro, D. N. & Forbus, K. Combining analogy with language models for knowledge extraction. In *3rd Conference on Automated Knowledge Base Construction* (2021).
33. Wang, X. et al. KEPLER: a unified model for knowledge embedding and pre-trained language representation. *Trans. Assoc. Comput. Linguist.* **9**, 176–194 (2021).
34. Jiang, X., Liang, Y., Chen, W. & Duan, N. XLM-K: improving cross-lingual language model pre-training with multilingual knowledge. In *Proc. AAAI Conference on Artificial Intelligence* Vol. 36, 10840–10848 (2022).
35. Guo, Y. et al. Webformer: pre-training with web pages for information retrieval. In *Proc. 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* 1502–1512 (ACM, 2022).
36. Gao, H. & Huang, H. Deep attributed network embedding. In *Procedings of the 27th International Joint Conference on Artificial Intelligence* 3364–3370 (AAAI Press, 2018).
37. Cui, G., Zhou, J., Yang, C. & Liu, Z. Adaptive graph encoder for attributed graph embedding. In *Proc. 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 976–985 (ACM, 2020).
38. Li, Y., Sha, C., Huang, X. & Zhang, Y. Community detection in attributed graphs: an embedding approach. In *Proc. AAAI Conference on Artificial Intelligence* Vol. 32 (2018).
39. Li, M. M., Huang, K. & Zitnik, M. Representation learning for networks in biology and medicine: advancements, challenges, and opportunities. Preprint at https://arxiv.org/abs/2104.04883 (2021).
40. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations* (OpenReview.net, 2017).
41. Hamilton, W., Ying, Z. & Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems, volume 30* (eds. Guyon, I. et al.) 1025–1035 (2017).
42. Vert, J.-P. & Yamanishi, Y. Supervised graph inference. In *Advances in Neural Information Processing Systems, volume 17* (eds Saul, L. et al.) (MIT Press, 2004).
43. Shaw, B. & Jebara, T. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09* (ACM, 2009).
44. Shaw, B., Huang, B. & Jebara, T. Learning a distance metric from a network. In *Advances in Neural Information Processing Systems, volume 24* (eds Shawe-Taylor, J. et al.) (Curran Associates, 2011).
45. Hadsell, R., Chopra, S. & LeCun, Y. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2*. 1735–1742 (2006).
46. Wang, X., Han, X., Huang, W., Dong, D. & Scott, M. R. Multi-similarity loss with general pair weighting for deep metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019* 5022–5030 (Computer Vision Foundation/IEEE, 2019).
47. Hou, J., Adhikari, B. & Cheng, J. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics* **34**, 1295–1303 (2018).
48. Klausen, M. S. et al. NetSurfP-2.0: improved prediction of protein structural features by integrated deep learning. *Proteins* **87**, 520–527 (2019).
49. Rocklin, G. J. et al. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science* **357**, 168–175 (2017).
50. Sarkisyan, K. S. et al. Local fitness landscape of the green fluorescent protein. *Nature* **533**, 397–401 (2016).
51. AlQuraishi, M. ProteinNet: a standardized data set for machine learning of protein structure. *BMC Bioinform.* **20**, 1–10 (2019).
52. Min, S., Park, S., Kim, S., Choi, H.-S. & Yoon, S. Pre-training of deep bidirectional protein sequence representations with structural information. *IEEE Access* **9**, 123912–123926 (2021).
53. Beltagy, I., Lo, K. & Cohan, A. SciBERT: a pretrained language model for scientific text. 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)* 3615–3620 (ACL, 2019).
54. Cohan, A., Ammar, W., van Zuylen, M. & Cady, F. Structural scaffolds for citation intent classification in scientific publications. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* 3586–3596 (ACL, 2019).
55. Jurgens, D., Kumar, S., Hoover, R., McFarland, D. & Jurafsky, D. Measuring the evolution of a scientific field through citation frames. Trans. *Assoc. Comput. Linguist.* **6**, 391–406 (2018).
56. Yasunaga, M., Leskovec, J. & Liang, P. LinkBERT: pretraining language models with document links. In *Proc. 60th Annual Meeting of the Association for Computational Linguistics* Vol. 1, 8003–8016 (Association for Computational Linguistics, 2022).
57. Gu, Y. et al. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Computi. Healthc.* **3**, 1–23 (2021).
58. McDermott, M., Yap, B., Hsu, H., Jin, D. & Szolovits, P. Adversarial contrastive pre-training for protein sequences. Preprint at https://arxiv.org/abs/2102.00466 (2021).
59. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical text-conditional image generation with clip latents. Preprint at https://arxiv.org/abs/2204.06125 (2022).

Acknowledgements

M.B.A.M. was partly supported by a National Institutes of Health (NIH) grant LM013337 and a collaborative research agreement with IBM, as well as by a Harvard Medical School Department of Biomedical Informatics Berkowitz Postdoctoral Fellowship. B.Y. was supported by a Massachusetts Institute of Technology (MIT) Undergraduate Research Opportunity fund. M.Z. gratefully acknowledges the support by NIH R01HD108794, US Air Force Contract No. FA8702-15-D-0001, and awards from Harvard Data Science Initiative, Amazon Faculty Research, Google Research Scholar Program, Bayer Early Excellence in Science, AstraZeneca Research, and Roche Alliance with Distinguished Scientists. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funders.

Author contributions

M.B.A.M. and B.Y. collated datasets, wrote modelling code and ran experiments. M.B.A.M. compiled the final results and completed the review of existing pre-training studies. M.B.A.M., P.S. and M.Z. conceived the study and shaped the framing of the work. P.S. and M.Z. offered insight and guidance throughout the project. M.B.A.M. and M.Z. wrote the final paper, and M.B.A.M., B.Y., P.S. and M.Z. contributed edits to drafts.

Competing interests

The authors declare no competing interests.

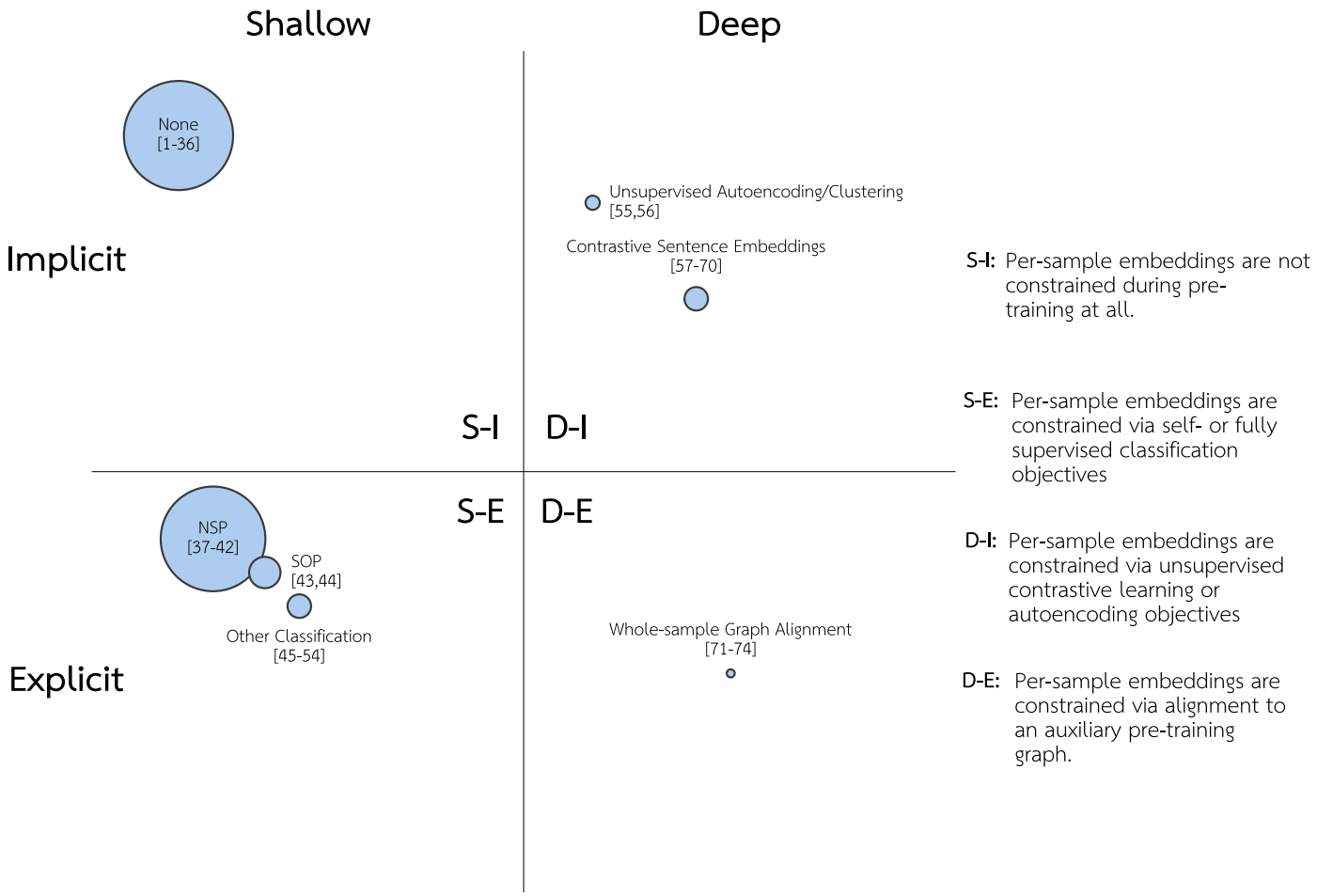
Additional information

Extended data is available for this paper at https://doi.org/10.1038/s42256-023-00647-z.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-023-00647-z.

Correspondence and requests for materials should be addressed to Marinka Zitnik.

https://doi.org/10.1038/s42256-023-00647-z



Extended Data Fig. 1 | Existing Pre-training (PT) Methods. A summary of 74 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 *implicit* 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[1]. "SOP" refers to "Sentence-order Prediction," the per-sample PT task introduced in ALBERT[10]. Note that over 90 studies in total were considered in our review, but only 74 met the inclusion criteria to be included in this figure. These methods are described in more detail in the Supplementary Information.

https://doi.org/10.1038/s42256-023-00647-z

Extended Data Table 1 | Final hyperparameters for our PROTEINS domain

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

https://doi.org/10.1038/s42256-023-00647-z

Extended Data Table 2 | Final hyperparameters for our ABSTRACTS dataset

Task	Number of epochs	LR
Paper Field	2	5e-5
ACL-ARC	4/5	5e-5
SciCite	3/2	1e-5

Final hyperparameters for our ABSTRACTS dataset. All models used a batch size of 32 and no early stopping to match the original SciBERT paper[53]. LR, learning rate. A / B = [LM PT Hyperparameter] / [SIPT Hyperparameter].

https://doi.org/10.1038/s42256-023-00647-z

Extended Data Table 3 | Results for the Proteins Domain

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

Results of the TAPE Transformer[5], the PLUS Transformer[52] (*: our measurements), our LM PT baseline, and two SIPT variants ("-C" indicates the contrastive loss, "-M" the multisimilarity loss). Higher is better, and best-performing results per task are bolded.

https://doi.org/10.1038/s42256-023-00647-z

Extended Data Table 4 | Results for the Abstracts Domain

Model	PF	SC	AA	SRE
SciBERT	**0.66**	0.85	0.71	0.80
BioLinkBERT	**0.66±0.0**	**0.86±0.01**	0.73±0.04	**0.82±0.02**
LM PT	**0.66±0.0**	0.85±0.01	0.70±0.05	0.80±0.01
SIPT-C	**0.66±0.0**	**0.86±0.01**	**0.76±0.02**	0.81±0.00
SIPT-M	**0.66±0.0**	0.85±0.00	0.73±0.05	N/A

Results of the original SciBERT[53] model, our own LM PT baseline, and two SIPT variants ("-C" indicates the contrastive loss, "-M" the multisimilarity loss). Higher is better, and best-performing results per task are bolded.

https://doi.org/10.1038/s42256-023-00647-z

Extended Data Table 5 | PT Link Retrieval Performance for the Proteins Domain

Method	λ_{SI}	LRAP	nDCG	AP	MRR
Random Baseline	N/A	0.88%	27.1%	0.88%	0.003
TAPE[5]	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

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.*, $\lambda_{SI} > 0$) leads to improved performance.

https://doi.org/10.1038/s42256-023-00647-z

Extended Data Table 6 | PT Link Retrieval Performance for the Abstracts Domain

Method	λ_{SI}	LRAP	nDCG	AP	MRR
Random Baseline	N/A	0.89%	26.0%	0.27%	0.016
SciBERT[53]	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[59]	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

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., $\lambda_{SI} > 0$) leads to improved performance.