

Rethinking Relational Encoding in Language Model Pre-Training for General Sequences

Matthew B. A. McDermott¹ Brendan Yap¹ Pete Szolovits¹ Marinka Zitnik²

Abstract

Language model pre-training (LMPT) has achieved remarkable results in natural language understanding. However, LMPT is much less successful in non-natural language domains like protein sequences, revealing a crucial discrepancy between the various sequential domains. Here, we posit that while LMPT can effectively model *per-token* relations, it fails at modeling *per-sequence* relations in non-natural language domains. To this end, we develop a framework that couples LMPT with deep structure-preserving metric learning to produce richer embeddings than can be obtained from LMPT alone. We examine new and existing pre-training models in this framework and theoretically analyze the framework overall. We also design experiments on a variety of synthetic datasets and new graph-augmented datasets of proteins and scientific abstracts. Our approach offers notable performance improvements on downstream tasks, including prediction of protein remote homology and classification of citation intent.

1. Introduction

Pre-trained language models² have revolutionized natural language processing (NLP) for short text spans (Devlin et al., 2019; Peters et al., 2018). This has prompted researchers to attempt to adapt these frameworks to other domains, including protein sequences (Rao et al., 2019), drug structures (Hu et al., 2020b), and tabular data (Yoon et al., 2020). Successes in these domains would be highly impactful across a broad range of scientific fields. However, in many of these domains, LMPT has realized less significant gains than in NLP, with examples in protein and tabular domains often failing to outperform non-PT approaches (Shanehsazzadeh

et al., 2020; Yoon et al., 2020).

We will let “per-token” and “per-sequence” refer to tasks that form predictions at the level of individual tokens (or short spans of tokens) and entire sequences (or pairs of sequences), respectively. We posit that while LMPT is quite effective across domains at pre-training for per-token tasks, it performs suboptimally on non-NLP domains overall because it is much less effective at modelling per-sequence tasks on these domains. This, we argue, is because many per-sequence tasks in NLP may be naturally reformulated as LM tasks and thus LMPT will naturally learn relevant per-sequence signals when pre-training over a sufficiently large dataset. In contrast, within other domains, this reformulation is usually not possible. Thus, LMPT will perform worse at per-sequence modelling even after extensive pre-training. Figure 1 shows this argument visually, and offers concrete examples of how per-sequence tasks can be reformulated as LM tasks in NLP, but not for protein sequences. The existing literature on LMPT also supports this explanation. For example, it is known that LMPT offers compelling few-shot performance across a variety of NLP tasks (Brown et al., 2020), and studies have leveraged the ability to reformulate per-sequence tasks as LM tasks directly in order to enhance pre-training (Raffel et al., 2019). See the Appendix for more detailed commentary on this point.

This argument suggests that we need to augment LMPT with additional tasks that are better able to capture per-sequence relationships. Existing approaches to do so typically simply add additional supervised tasks into the PT process (Liu et al., 2019; Min et al., 2020). However, this approach is often limited, as not only will we rarely have large, fully-labeled datasets suitable for PT, but including additional supervised PT tasks risks negative transfer to fine-tuning (FT) tasks, a phenomenon commonly observed in multi-task learning (Ruder, 2017; Wu et al., 2020). Another related paradigm within NLP is the use of external knowledge graphs during PT (Shen et al., 2020; Liu et al., 2020). However, these efforts almost exclusively focus on how to inject knowledge into LMPT at a *per-token/per-entity* level rather than a *per-sequence* level, similar to NLP example 1.2 in Figure 1. While important, such work is orthogonal to our effort given its per-entity, not per-sequence, focus.

¹Massachusetts Institute of Technology, Cambridge, MA, USA

²Harvard University, Cambridge, MA, US. Correspondence to: Matthew B. A. McDermott <mmd@mit.edu>.

²LM, masked (MLM) or traditional language model; PT, pre-training; LMPT, language model pre-training.

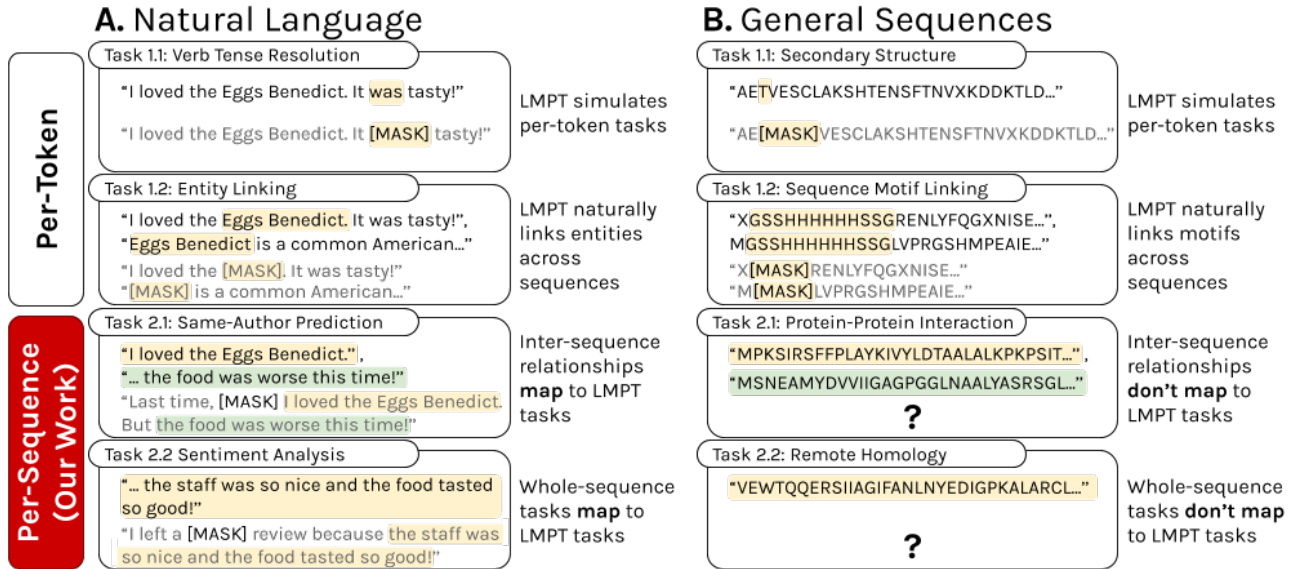


Figure 1. Examples indicating how LMPT succeeds or fails at capturing different kinds of per-token and per-sequence relationships across (A) NLP and (B) general sequences, illustrated here with protein sequences. Each example task (rounded box) displays (above) a possible sequence associated with a particular fine-tuning task, and (below) the realization, if such exists, of this task as an LMPT task, in both cases with task-relevant spans highlighted. Our framework is primarily designed to augment *per-sequence* processing. Protein sequences and tasks are sourced from (Rao et al., 2019; AlQuraishi, 2019).

Present work. We introduce a new pre-training framework called *Structure-Preserving Pre-Training (SPPT)*³ which leverages PT graphs to ensure that PT models reflect per-sequence relationships. The insight behind our approach is to jointly optimize for both the traditional LM task (to capture per-token information) and a new, deep-learning variant of Structure-Preserving Metric Learning (SPML) (to capture per-sequence information), which tasks a model to represent sequences via similar embeddings if and only if they are linked in the PT graph (Vert & Yamanishi, 2004; Shaw & Jebara, 2009; Shaw et al., 2011). See Figure 2 for a visual overview of our framework. Remarkably, SPPT enables us to theoretically relate the structure of the PT graph to direct FT task suitability. Further, SPPT generalizes a number of existing PT frameworks across several domains.

We test SPPT on synthetic and real domains. On synthetic data, we show that SPPT induces embeddings that are reflective of particular graph structures, just as our forthcoming theoretical analysis predicts, and we examine its performance as a function of the amount of relational content encoded in the graph. On real data, we show on seven downstream tasks spanning protein sequences and scientific articles (protein-protein interaction graphs (Zitnik et al., 2019) and academic citation graphs (Wang et al., 2019a)) that

³Code is available at https://github.com/mmcdermott/structure_preserving_pre-training.

SPPT leads to FT task performance that either matches or exceeds LMPT results alone. Most notably, we see that SPPT induces a gain of up to 4% in accuracy over the TAPE (Rao et al., 2019) pre-trained transformer on remote homology prediction, achieving a new state-of-the-art performance, and a gain of up to 5% macro F1 over SciBERT (Beltagy et al., 2019) on citation intent classification.

In sum, our key contributions are:

- We introduce SPPT, a principled, self-supervised pre-training framework that leverages relational graphs during PT but requires no additional input for FT. SPPT generalizes many self- and weakly-supervised PT frameworks, including BERT, multi-task, and metric learning PT.
- We show that SPPT provides unique insights into how the topology of the PT graph affects the FT performance.
- SPPT surpasses a number of baseline methods on multiple domains and challenging downstream tasks, and it dominates comparable approaches that use LMPT alone.

2. Related Work

Natural language pre-training. Pre-training has been extensively explored within NLP. Building on early success (Peters et al., 2018; Devlin et al., 2019), researchers have explored purely autoregressive approaches (Brown et al., 2020), hybrid approaches (Yang et al., 2019), approaches

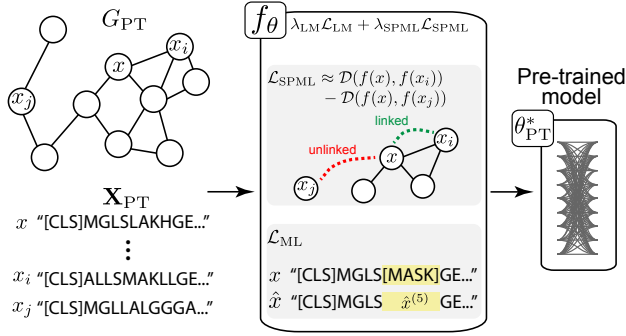


Figure 2. The Structure-Preserving Pre-Training (SPPT) framework incorporates both LMPT and Structure-Preserving Metric Learning (SPML) for per-token and per-sequence modeling, respectively. The full SPML loss is defined in Section 3.1.

that leverage external knowledge graphs to inject knowledge at a per-token/per-entity level (Sun et al., 2019; Liu et al., 2020; Shen et al., 2020), and approaches that improve efficiency through specialized learning architectures (Clark et al., 2019; Chen, 2020).

Language model pre-training for biomedical sequences.

Across drugs (Hu et al., 2020b; Wang et al., 2019b), tabular (Yoon et al., 2020), time series (McDermott et al., 2020), and protein data (Rao et al., 2019), many non-NLP domains have benefited from LMPT. However, gains here have been much more limited than in NLP. Additionally, many of the successful approaches in these domains already leverage partially supervised pre-training objectives (Hu et al., 2020b; McDermott et al., 2020; Yoon et al., 2020; Min et al., 2020; Filipavicius et al., 2020), which agrees with our hypothesis that LMPT must be augmented with additional tasks to model per-sequence relationships.

Pre-training theory. Retrospective analyses have found that pre-trained LMs reflect properties consistent with domain knowledge (Coenen et al., 2019; Tenney et al., 2019; Rao et al., 2020). Prospectively, Lee et al. suggests that autoregressive PT will transfer positively to downstream tasks that obey certain conditional independence relationships with the PT learning strategy (Lee et al., 2020). Other than these works, we are not aware of works that offer practical, theory-driven guidance on to what degree LM PT will be helpful for a given set of FT tasks.

Metric learning. Supervised metric learning (Kulis et al., 2012) aims to learn an embedder f that minimizes the distance between samples that have the same label while maximizing it between those with different labels (Suárez et al., 2020). In contrast to linear metric learning, where the learned distance is the squared Euclidean distance after applying a linear transformation to instances globally (Gold-

berger et al., 2004; Weinberger & Saul, 2009; Davis et al., 2007), alternative methods utilize kernels (Kulis et al., 2006), globally consistent local distance functions (Weinberger & Saul, 2008), deep methods (Roth et al., 2020; Schroff et al., 2015), and adversarial frameworks (Duan et al., 2018; Zheng et al., 2019). Metric learning loss functions have been successfully used as regularization to improve robustness of deep models (Mao et al., 2019). Recent studies have also examined ways to unify existing metric learning losses (Wang et al., 2019c).

A sub-field within metric learning that we use in SPPT is *Structure-Preserving Metric Learning* (SPML) (Vert & Yamaniishi, 2004; Shaw & Jebara, 2009; Shaw et al., 2011). Given an input graph $G = (V, E)$, classical SPML is concerned with learning a (linear) embedding of V such that the graph G can be recovered in the embedding space via a connectivity algorithm (Shaw et al., 2011). To the best of our knowledge, there has been minimal investigation of any generalizations to or improvements on SPML in the context of the existing progress regarding deep metric learning.

3. Structure-Preserving Pre-Training

We use the notation $\mathbf{X}_{PT} \in \mathbb{R}^{N_{PT} \times d_f}$ to refer to a pre-training dataset of $N_{PT} \in \mathbb{N}$ sequences (samples), each of *feature* dimension $d_f \in \mathbb{N}$. We use \mathbf{X}_{PT} to refer to both the tensor and set of sequences $\{\mathbf{x}_i | 1 \leq i \leq N_{PT}\}$. Let $d_e \in \mathbb{N}$ be our *embedding* dimension and $\theta \in \Theta$ be a parameter vector. Finally, we use notation $G = (V, E)$ to represent a graph with vertices V and edges E .

Problem Definition. Given PT dataset $\mathbf{X}_{PT} \in \mathbb{R}^{N_{PT} \times d_f}$ and $G_{PT} = (V, E)$, $\mathbf{X}_{PT} \subseteq V$, we aim to learn an encoder $f_\theta : \mathbb{R}^{d_f} \rightarrow \mathbb{R}^{d_e}$ that embeds sequences in \mathbf{X}_{PT} into d_e -dimensional vectors in order to maximize the performance on downstream tasks $(\mathbf{X}_{FT}, \mathbf{Y}_{FT}) \in \mathbb{R}^{N_{FT} \times d_f} \times \mathbb{R}^{N_{FT} \times d_y}$ that are unknown at pre-training time.

We presume f_θ permits extracting both a representation of the sequence at a *per-token* representation as well as at a global, *whole-sequence* level—for example, the BERT encoder produces both contextual embeddings of each token in the sentence and the embedding of the auxiliary [CLS] token, which is commonly used as a whole-sequence embedding (Devlin et al., 2019). Note that neither does f_θ take the graph G_{PT} as a direct input nor are we guaranteed to have access to an analogous graph G_{FT} at fine-tuning time.

Structure-Preserving Pre-Training (SPPT). In SPPT framework, we learn encoder f_θ by jointly solving two tasks: a traditional LM task (with loss \mathcal{L}_{LM}) and a Structure-Preserving Metric Learning (SPML) task (with loss \mathcal{L}_{SPML}), which pushes f 's whole-sequence embeddings to reflect the relationships encoded in G_{PT} . We will learn these two tasks jointly via a convex combination of their losses with

normalized weights λ_{SPML} and $\lambda_{\text{LM}} = 1 - \lambda_{\text{SPML}}$:

$$\theta_{\text{PT}}^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \lambda_{\text{LM}} \mathcal{L}_{\text{LM}} + \lambda_{\text{SPML}} \mathcal{L}_{\text{SPML}}|_{\theta}$$

In the remainder of this section, we will detail the SPML task, as this is the novel component of our PT framework. Details for the LM component (which are all standard) can be found in the Appendix.

3.1. Deep Structure-Preserving Metric Learning

Problem Definition. Let \mathcal{D} be a distance metric in \mathbb{R}^{d_e} . Structure-Preserving Metric Learning (SPML) aims to learn embeddings that minimize $\mathcal{D}(f_{\theta}(\mathbf{x}_i), f_{\theta}(\mathbf{x}_j))$ for linked pairs $(\mathbf{x}_i, \mathbf{x}_j) \in E$ while maximizing $\mathcal{D}(f_{\theta}(\mathbf{x}_r), f_{\theta}(\mathbf{x}_s))$ for pairs $(\mathbf{x}_r, \mathbf{x}_s) \notin E$.

Note that this contrasts with the traditional deep metric learning problem over a labeled dataset \mathbf{X}, \mathbf{y} in which we must learn f such that $f(\mathbf{x}_i), f(\mathbf{x}_j)$ are similar if and only if $y_i = y_j$. In the Appendix, we show in detail how to adapt a significant amount of the existing literature regarding traditional deep metric learning to this distinct, under-explored structure-preserving setting, including a derivation of Wang et al.’s General Pair Weighting framework for SPML.

SPML loss $\mathcal{L}_{\text{SPML}}$. We extend two traditional metric learning losses in this work for the structure-preserving context: A simple contrastive loss (Hadsell et al., 2006), presented fully in the Appendix, and the Multi-Similarity loss (Wang et al., 2019c), which we define here. Let $G = (\mathbf{X}, E)$ be the graph over which we are learning. The Multi-Similarity loss is specified by positive and negative weights w_+ , w_- and a threshold t :

$$\begin{aligned} \mathcal{L}_{\text{SPML}}^{(\text{MS})} = & \frac{1}{Nw_+} \log \left(1 + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in E} e^{-w_+(\langle f(\mathbf{x}_i), f(\mathbf{x}_j) \rangle - t)} \right) \\ & + \frac{1}{Nw_-} \log \left(1 + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \notin E} e^{w_-(\langle f(\mathbf{x}_i), f(\mathbf{x}_j) \rangle - t)} \right) \end{aligned}$$

SPML negative sampling strategies. Typically, traditional metric learning methods only leverage *intra-batch* negative sampling algorithms, and furthermore use only the induced embedding-space similarity between samples (based on the current iteration of the embedder) to inform sampling. In the SPML context, we take advantage of graph structure to further inform negative sampling with a tiered strategy, in which batches are first sampled based on *source-graph* distance, and only then are sequences within a batch sampled based on their embedding-space distance.

Concretely, in our best performing experiments on protein sequences, we first sample batches such that negative samples for each node are a subset of proteins from the same

species (each of which is a separate component of the overall graph), then use the SPML Multi-Similarity loss, which absorbs intra-batch sampling by aggregating over all pairs of nodes. On scientific articles, where the majority of nodes are in a single large connected-component, we leverage random negative sampling at both levels, and leverage a contrastive loss. Further discussion of these losses is in the Appendix.

4. New Theoretical Properties

In this section, we will explore some of the theoretical properties of the SPPT framework induced by the inclusion of the SPML task. First, let us define that an encoder f achieves *SPML-optimal embeddings* at radius $r \in \mathbb{R}^+$ if $\mathcal{D}(f(\mathbf{x}_i), f(\mathbf{x}_j)) < r \iff (\mathbf{x}_i, \mathbf{x}_j) \in E$. In the remainder of this section, we will explore when we can and cannot expect there to exist optimal solutions to the SPML problem, as well as how optimal SPML solutions reflect the topology of the graph G_{PT} —and thus how they relate to FT task performance.

4.1. When Does G_{PT} Permit Optimal Embeddings?

There are a number of well-known results regarding which graphs permit optimal node embeddings (Shaw et al., 2011; Vert & Yamanishi, 2004). For example, as shown in the Appendix, an arbitrary graph $G = (V, E)$ requires at least $d_e = \Theta(|V|)$ dimensions to produce an embedding such that k - or radius-nearest neighbor connectivity algorithms under the Euclidean distance metric will recover G .

Using $d_e = \Theta(|V|)$ is clearly infeasible. For a more practical setting, we will examine two types of graphs which arise naturally in practice and permit much smaller optimal embeddings: disconnected cliques, which correspond to traditional metric learning problems, and manifold nearest-neighbor graphs. We also explore both of these graphs via synthetic experiments in Section 5.

1) Disconnected cliques. Disconnected cliques allow us to capture supervised metric-learning problems in a structure-preserving manner, and yield SPML-optimal embeddings for all $d_e \in \mathbb{N}$ by first assigning each clique i a unique point $\mathbf{c}_i \in \mathbb{R}^{d_e}$, then mapping each node in clique i to \mathbf{c}_i directly.

2) Manifolds. Suppose our data lives approximately on a suitably defined manifold \mathcal{M} of dimension $d_m \in \mathbb{N}$, and that G_{PT} approximates a radius-nearest neighbor graph on this manifold. Then it is known that G_{PT} permits an SPML-optimal embedding in no more than $2d_m$ dimensions, regardless of $|V|$. See the Appendix for more details.

4.2. Properties of SPML at Optimality

Note that by definition we can use an SPML-optimal embedding to directly recover G_{PT} by forming a nearest-neighbor

graph with cutoff r in the output space $f(V)$.

Now, suppose we have a task of interest defined by a *true* labeling function $y : \mathbf{X} \mapsto \mathcal{Y}$. Given the graph $G_{PT} = (\mathbf{X}_{PT}, E)$, let us define the neighborhood function $N(\mathbf{x}) = \{\mathbf{x}' \in \mathbf{X}_{PT} | (\mathbf{x}, \mathbf{x}') \in E\}$ and the (estimated) labeling function $MC_{G_{PT}, y}(\mathbf{x}) = \operatorname{argmax}_{l \in \mathcal{Y}} \sum_{\mathbf{x}' \in N(\mathbf{x})} \mathbb{1}_{y(\mathbf{x}') = l}$. Then, we can define the *homophily* of the task y over the graph G_{PT} by $\mathbb{P}(y(\mathbf{x}) = MC_{G_{PT}, y}(\mathbf{x}))$ (Zhu et al., 2020; Huang & Zitnik, 2020; Zhang et al., 2016).

Next, let us presume f is an SPML-optimal embedder with cutoff r . Then, as it recovers the graph G_{PT} , a radius nearest-neighbor classifier in the embedding space \mathbb{R}^{d_e} defined via $\hat{y}(\mathbf{x}) = \operatorname{argmax}_{l \in \mathcal{Y}} \sum_{\mathbf{x}_i | \mathcal{D}(\mathbf{x}, \mathbf{x}_i) < r} \mathbb{1}_{y(\mathbf{x}_i) = l}$ will have accuracy given precisely by the homophily of the task y over G_{PT} , by definition. Thus, we have shown that we can directly link the input graph structure—in particular, as measured through homophily—to FT task suitability under SPPT at optimality. Note that other forms of graph dependence (e.g., structural equivalence rather than homophily) are not captured as cleanly by SPML.

4.3. Expressing Pre-training Models from the Literature

The SPPT framework generalizes many pre-training strategies, several of which we outline concretely below. Note that here our interest is in equivalence in pre-training *objective*, rather than the prescribed pre-training *algorithm*.

4.3.1. SUPERVISED PRE-TRAINING

1) Link prediction. Link prediction is trivially an example of an SPPT task as SPML is itself directly a form of link prediction, through a classifier based on induced embedding space distance. Setting $\lambda_{SPML} = 1$ realizes the entire SPPT framework as a link prediction task.

2) Traditional metric learning pre-training. We can realize any supervised metric learning task as an SPPT task with $\lambda_{SPML} = 1$ via a graph of disconnected cliques corresponding to class labels.

3) Supervised classification pre-training. As before, we form a graph composed of disconnected cliques with edges linking only those nodes that share the same label. This realizes classification as link prediction, and thus by our prior argument, SPPT with $\lambda_{SPML} = 1$ captures this task.

4.3.2. LANGUAGE MODEL PRE-TRAINING

4) Traditional LMPT. Any pre-training method that does not entail any cross-sample tasks can be realized as a task within SPPT by setting $\lambda_{SPML} = 0$ and $\lambda_{LM} = 1$ and then directly using the LM PT method.

5) Joint LM and supervised PT. As we can realize any supervised learning objective from Section 4.3.1 as an SPPT

task, any joint LM and supervised learning task is also a valid SPPT task, now with $\lambda_{LM} > 0$ and $\lambda_{SPML} > 0$. This includes models such as MT-DNN (Liu et al., 2019) (which includes a supervised, multi-task learning component), BERT (Devlin et al., 2019) (which includes a link prediction task on a graph of linear chains represent sequences of text spans), and PLUS (Min et al., 2020) (which includes a protein-family prediction task).

6) ALBERT (Lan et al., 2019). ALBERT’s supervised learning task is a *directed* edge vs. anti-edge prediction task, which therefore does not naively fit into the SPML framework, which as presented here only deals with *undirected* graphs. However, simple extensions of SPML to accommodate directed graphs or graphs with multiple edge types would support ALBERT.

5. Results on Datasets with Synthetic Graphs

We create novel datasets with synthetic graphs to investigate three key questions regarding SPPT. First, 5.1: Does SPPT yield high-performance embeddings across diverse graph types? Second, 5.2: Does SPPT retain performance when graphs suffer from mild levels of noise? Finally, third, 5.3: Does SPPT learn embeddings that are selectively informative of tasks in accordance with their particular graph topology in line with our predictions in Section 4.2?

Shared experimental procedure. Separately for each experiment, we construct a PT graph and dataset of free-text sentences⁴ labeled with LDA-produced topics. In all cases, we pre-train both LMPT and SPPT ($\lambda_{SPML} = 0.1$) models using a shallow transformer encoder f and character-level tokenizer. After PT, FT is performed in a zero-shot manner using 3-nearest-neighbor euclidean-distance classifiers in the output embedding space. As our interest in these experiments is not to establish SPPT generalizability (which is demonstrated by its strong performance in Section 6 on real data) but rather to show that SPPT learns to model the graph G_{PT} in the expected manner, FT is performed over the PT sequences directly in a zero-shot manner. Final evaluation uses the AUROC of the 3-NN classifiers.

5.1. SPPT: Diverse Graph Types

Motivation. In Section 4, we argued that SPPT would be particularly well-suited to component-wise cliques and manifold nearest-neighbor graphs. Here, we test this claim using synthetic graphs of both of these types.

Datasets. To construct graphs that are component-wise cliques, we sample a set of sentences with approximately equal topic distributions, then add edges between any two

⁴Source: <https://www.kaggle.com/mikeortman/wikipedia-sentences>

Graph Type	Graph	LMPT	SPPT
Cliques	N/A	0.62	0.94
Manifold	Plane	0.51	0.94
	Möbius	0.55	0.89
	Sphere	0.51	0.86
	Torus	0.54	0.77
Mechanistic	Homophily	0.94	0.99
	Motif	0.73	0.88
	Structural	0.90	0.79

Table 1. Comparisons between zero-shot k -NN FT AUROC (higher is better) of LMPT models and SPPT models over various graphs with various forms of structural alignment.

sentences with the same topic. For manifolds, we restrict our attention to two-dimensional simplicial complexes, formed by stitching together induced topic simplicies composed of triples of topics, with sentences localized onto those topic simplicies corresponding to the three topics which capture the most probability mass of the entire topic distribution for that sample. With localized sentences, G_{PT} can be formed by producing a radius r nearest-neighbor graph over the simplicial manifold using geodesic distances. We use manifolds corresponding topologically to a plane, Möbius strip, hollow sphere, and a hollow torus.

Results. Table 1 shows the performance of SPPT vs. LMPT across all graphs we test in this experiment. As hypothesized, we see uniform improvements of SPPT over LMPT across all clique and manifold datasets.

5.2. SPPT: Noise Rate and Relational Content

Motivation. In Section 4, we claimed that SPPT should yield embeddings reflective of the underlying structure of G_{PT} . In this experiment, we test this claim by examining the zero-shot FT performance of a model across increasing levels of noise—and thus decreasing relational information—in the input graph G_{PT} .

Datasets. We use clique graphs from Section 5.1, perturbed by randomly adding edges with probability $p \in [0, 0.5]$.

Results. Figure 3 compares SPPT performance to LMPT performance as we increase the noise rate from 0 to 50% on an otherwise high-homophily component-wise clique graph. As expected, we can see that for all values of noise up to 15%, SPPT retains a strong improvement over LMPT, and that even at 50% noise, SPPT is still quite competitive with LMPT, indicating that SPPT falls back gracefully to its LM component under conditions of extreme noise.

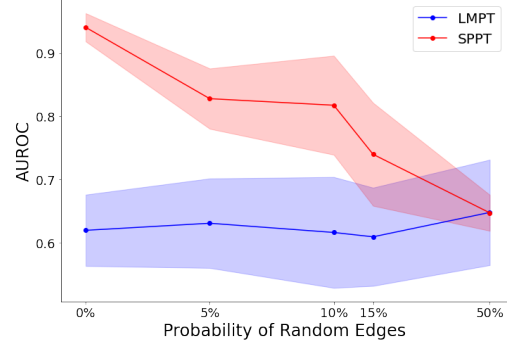


Figure 3. Zero-shot FT AUROC (y -axis) versus graph noise rate (x -axis) of an MLM model and an SPPT model. For noise rates up to 10% we see stark improvements of the SPPT over LMPT, and at 50% noise, the two approaches perform comparably.

5.3. SPPT: Homophily vs. Structural Equivalence

Motivation. In Section 4, we claimed that SPML-optimal embeddings will offer zero-shot radius nearest-neighbor FT performance that is proportional to the homophily of the FT task over G_{PT} . In this experiment, we test this claim with synthetic datasets containing tasks with varied topologies over G_{PT} ; in particular, we expect to see SPPT significantly outperform LMPT on tasks with high homophily, and to under-perform LMPT on tasks with no homophily (as in that case embeddings will be explicitly expected to reflect an opposing notion of structure).

Datasets. Graphs in this setting contain a backbone cycle of parametrized size with a number of random graph motifs affixed regularly along the cycle (Donnat et al., 2018). Sentences are assigned to nodes such that sentence topics align with one of three labeling schemes: for the *homophily* graph, we assign node labels based on a k -means clustering algorithm applied over indicator vectors describing a node’s immediate neighborhood; for the *motif* graph, we assign node labels based on which motif the node is a part of in the broader graph; and finally, for the *structural* graph, we assign node labels based on a k -means clustering algorithm applied over graphlet degree vectors which describe a node’s structural role in the graph. Further details on featurizations are in the Appendix. We expect the homophily dataset to have the highest level of homophily with topic labels, the motif dataset the next highest, and finally the structural dataset, which should have negligible homophily. These datasets allow us to test how SPPT performs on graphs with high-homophily and low-homophily FT tasks, as desired.

Results. In addition to our clique and manifold results, Table 1 shows the performance difference between SPPT and LMPT models across each of the three synthetic graphs in this experiment. As expected, on both the homophily and motif datasets, SPPT significantly outperforms LMPT, and

in the structural dataset, which by design shows high structural equivalence but no homophily, SPPT underperforms LMPT. Note this is in some sense a counterpoint to the prior results regarding cases with low homophily – there, we saw that when graphs degrade due to noise, SPPT and LMPT remain competitive, but here we see that when the graph is, by design, opposed to homophily, SPPT underperforms LMPT. Together, both results support that our method is explicitly capturing graph structure (and in particular capturing homophily).

6. Results on Datasets with Real-world Graphs

Next, we demonstrate on two domains (protein biology in Section 6.1 and scientific articles in Section 6.2) that SPPT can improve fine-tuning performance on real data and graphs. In both domains, we create new pre-training datasets that couple sequences with pre-training graphs.

Our primary experimental procedure to assess these claims will be to pre-train models under both traditional LMPT and SPPT, then fine-tune both systems on a selection of appropriate, established benchmark tasks. Results on these fine-tuning tasks will be our final comparison metrics. In order to minimize the required run-time of pre-training experiments and to enable direct comparisons of the value SPPT adds during pre-training, we do not pre-train our models from scratch here but instead initialize our pre-training models from existing, public models and continue pre-training for a smaller number of epochs under SPPT. In order to select the precise weighting (λ_{SPML} , λ_{LM}) of the SPPT loss terms, we performed pre-training under various values of λ_{SPML} , then selected the value of λ_{SPML} in order to yield maximal performance at link-retrieval over the pre-training graph. Across both domains, full details on model architectures and hyperparameter tuning are in the Appendix.

Following our hypothesis that LMPT is much less effective at modeling per-sequence relationships outside of the natural language (Section 1), we expect that SPPT will be most effective on the protein dataset. Similarly, we also expect that on the protein dataset, SPPT will more aggressively leverage the SPML (as measured from the weighting term λ_{SPML}) than it will on the scientific articles dataset.

6.1. Protein Sequences

Dataset and protein interaction graph. We use Stanford Tree-of-life dataset to source our pre-training protein sequences as our protein interaction graph (Zitnik et al., 2019). Tree-of-life is a multi-species (e.g., human, zebrafish, mouse) graph whose nodes are proteins and edges indicate protein-protein interactions. We restrict this graph to proteins that have some observed interactions (i.e., non-zero degree nodes), which yields a pre-training cohort with

1,450,633 proteins over 1,840 species, with 8,762,166 interactions. As this dataset is used for pre-training, not fine-tuning (and thus not for our final model comparisons), we employ no train/test split here, but instead leverage the entire graph and dataset for pre-training. We use the TAPE benchmark (Rao et al., 2019) as the source of fine-tuning data/tasks and for our initial PT model.

Fine-tuning tasks. We assess our model on four TAPE fine-tuning tasks (Rao et al., 2019). Briefly, with further commentary in the Appendix, 1) **Remote homology (RH)** is a per-sequence, multi-class classification task to predict a protein fold category (metric: accuracy); 2) **Secondary structure (Sec. Str.)** is a per-token, multi-class classification task to predict structural properties of each amino acid in a folded protein (metric: accuracy); and 3) **Stability (St.)** & 4) **Fluorescence (Fl.)** are per-sequence, continuous regression tasks to predict the protein’s stability and fluorescence intensity, respectively (metric: Spearman’s ρ).

Baselines. We consider the following three baselines. First, to assess the direct benefit of SPPT, we compare against both the originally published TAPE results (Rao et al., 2019) and results using our code and pre-training settings, but with $\lambda_{\text{SPML}} = 0$, so that only additional LMPT is performed. If we find that SPPT outperforms both of these, we will have shown that even with only minimal additional pre-training, SPPT offers improvements over a competitive, published system in a manner that is not attributable to additional LMPT alone, and therefore must rely on the novel components of SPPT. Finally, in addition to these baselines, we also compare against Min et al.’s PLUS system, which augments LMPT with an additional supervised PT task, in order to assess the benefits SPPT offers against competing approaches that also attempt to improve per-sequence modeling. Note that we restrict our attention to Transformer encoders here, and as such consider only Transformer results in our baseline comparisons as well.

Results. Table 2 shows the comparison of our model under the Multi-Similarity loss and contrastive loss against all baselines. We see that SPPT improves over all baselines on all tasks here except for Fluorescence, where we only match the TAPE Transformer. These improvements are particularly notable on the remote homology task, where we improve over TAPE by 5.6%, of which approximately 2.8% is directly attributable to SPPT. To the best of our knowledge, this result on Remote Homology achieves a new state-of-the-art (SOTA) result on this task, albeit only by 1%, and matches the SOTA on Fluorescence and Stability. Altogether, these results strongly support the claim that the SPPT system offers significant value in real-world pre-training scenarios.

Model	RH	Fl.	St.	Sec. Str.
TAPE	21%	0.68	0.73	73%
PLUS	19.8%±1.7*	0.63	0.76	73%
LMPT	23.8%±1.1	0.67±0.00	0.76±0.02	73.9%±0.0
SPPT	25.1%±0.6	0.68±0.00	0.77±0.01	73.9%±0.0
Contr.				
SPPT	26.6%±1.0	0.68±0.00	0.76±0.01	74.2%±0.1
Multi.				

Table 2. Results of the TAPE Transformer (Rao et al., 2019), the PLUS Transformer (Min et al., 2020) (*: our measurements), our LMPT baseline, and two SPPT variants. Higher is better.

6.2. Natural Language Sequences

Datasets and co-citation graph. Our pre-training is performed over the Microsoft Academic Graph (MAG) dataset (Wang et al., 2019a) retrieved from the Open Graph Benchmark (Hu et al., 2020a). This graph has nodes representing scientific articles (for which free-text abstracts are available), as well as auxiliary nodes corresponding to authors, topics, and publication venues, with edge types corresponding to authorship, topicality, publication, and citations. We restrict the dataset to only those “paper” nodes with associated free-text abstracts, and only consider “citation” edges. This yields a graph with 649,880 nodes/abstracts and 4,302,412 edges. Further details about the dataset are in the Appendix. We use the SciBERT benchmark (Beltagy et al., 2019) for our fine-tuning data/tasks and initial PT model.

Fine-tuning tasks. We assess our model over three SciBERT fine-tuning tasks: Paper field, SciCite, and ACL-ARC (Beltagy et al., 2019). All tasks are per-sentence, multi-class classification problems and will be evaluated in Macro-F1. Briefly, **Paper Field** task is to predict a paper’s area of study from its title, and the **SciCite** and **ACL-ARC** tasks both are to predict an “intent” label for sentences that cite other scientific works within academic articles, albeit over different datasets and with different classes.

Baselines. As with our protein-domain experiments, we first compare to the original SciBERT model and a model trained under our framework with $\lambda_{\text{SPML}} = 0$ so that only additional LMPT is performed. There are fewer options here than in the protein domain for a suitable external baseline or competing system. In particular, to the best of our knowledge, no work exists that leverages any form of supervised or semi-supervised pre-training methods at a per-sequence level over the SciBERT benchmark. We instead compare to Gururangan et al.’s DAPT system as an external baseline, as this model is both public and yields the best performance we can find out of any published (excluding preprint-only) model on the ACL-ARC task (chosen since it is where our method offers the largest improvement over SciBERT).

Model	Paper Field	SciCite	ACL-ARC
SciBERT	0.66	0.85	0.71
DAPT	N/A	N/A	0.75±0.03
LMPT	0.66±0.0	0.85±0.01	0.70±0.05
SPPT (Contrast.)	0.66±0.0	0.86±0.01	0.76±0.02
SPPT (Multisim.)	0.66±0.0	0.85±0.00	0.73±0.05

Table 3. Results of the original SciBERT (Beltagy et al., 2019) model, the DAPT model (Gururangan et al., 2020) and our own LMPT baseline against the SPPT framework. Higher is better. Gururangan et al. do not measure Paper Field or SciCite tasks.

Results. Table 3 shows the comparison of our SPPT framework against all baselines. We find that SPPT outperforms both SciBERT and LMPT on both ACL-ARC and SciCite, and matches the other two on the Paper Field task. Most notably, on the ACL-ARC task, SPPT outperforms SciBERT by over 0.04, and our own baseline by over 0.06 (both in Macro-F1, out of 1.0). However, SPPT’s improvement is much smaller compared to DAPT (Gururangan et al., 2020), which we outperform by 0.01.

6.3. Natural Language vs. Protein Sequences

Results in Section 6.1-6.2 also provide empirical evidence for our premise (Section 1) that LMPT models would benefit from additional support in modeling per-sequence tasks on non-NLP domains. We observe three points consistent with this hypothesis. First, we see that SPPT yields a smaller performance benefit overall on the NLP dataset relative to the protein dataset. Second, we also observe that the automatically selected λ_{SPML} parameter for NLP was a full order of magnitude smaller than it was for proteins (0.01 vs. 0.1). Third, we note that LMPT much more naturally captured the structure of the graph G_{PT} (via its link-retrieval performance) on NLP than it did on proteins. In NLP, SPPT only improved the average precision (AP) during retrieval by 9.6%, versus over 30% on proteins. Overall, these findings suggest that unaltered LMPT is much more able to model per-sequence tasks in NLP than over proteins. See the Appendix for further details.

7. Conclusion

We presented Structure-Preserving Pre-Training (SPPT), a pre-training framework that relies on deep structure-preserving metric learning to significantly improve per-sequence modeling of general sequences, a data modality incredibly important to a variety of applications. We demonstrated both theoretically and empirically that SPPT offers improvements over regular LMPT; showing that SPPT produces embeddings that naturally reflect the topology of the pre-training graph, as well as showing that on two real-world

domains SPPT is equal to or better than LMPT alone.

To further explore the SPPT framework, there are several notable future directions. Firstly, exploring various kinds of graphs, such as temporal or knowledge graphs, could offer significant improvements to PT in some contexts. Secondly, pre-training a model from scratch, rather than from an initialized LMPT model, could yield significant gains. Finally, repeating our analyses on other biological domains, such as on CRISPR sequences in genome editing (Barrangou & Doudna, 2016), is a natural direction for future work.

References

- AlQuraishi, M. ProteinNet: a standardized data set for machine learning of protein structure. *BMC Bioinformatics*, 20(1), 2019.
- Bao, H., Dong, L., Wei, F., Wang, W., Yang, N., Liu, X., Wang, Y., Gao, J., Piao, S., Zhou, M., et al. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *ICML*, pp. 642–652, 2020.
- Barrangou, R. and Doudna, J. A. Applications of CRISPR technologies in research and beyond. *Nature Biotechnology*, 34(9):933–941, 2016.
- Beltagy, I., Lo, K., and Cohan, A. SciBERT: A Pretrained Language Model for Scientific Text. In *EMNLP*, 2019.
- Brown et al. Language Models are Few-Shot Learners. *arXiv preprint arXiv: 2005.14165*, 2020.
- Chen, L. Variance-reduced Language Pretraining via a Mask Proposal Network. *arXiv: 2008.05333*, 2020.
- Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *ICLR*, 2019.
- Coenen, A., Reif, E., Yuan, A., Kim, B., Pearce, A., Viégas, F., and Wattenberg, M. Visualizing and Measuring the Geometry of BERT. In *NeurIPS*, 2019.
- Cohan, A., Ammar, W., van Zuylen, M., and Cady, F. 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.
- Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. Information-theoretic metric learning. In *ICML*, pp. 209–216, 2007.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019.
- Donnat, C., Zitnik, M., Hallac, D., and Leskovec, J. Learning structural node embeddings via diffusion wavelets. In *KDD*, pp. 1320–1329, 2018.
- Duan, Y., Zheng, W., Lin, X., Lu, J., and Zhou, J. Deep adversarial metric learning. In *CVPR*, pp. 2780–2789, 2018.
- Filipavicius, M., Manica, M., Cadow, J., and Martinez, M. R. Pre-training Protein Language Models with Label-Agnostic Binding Pairs Enhances Performance in Downstream Tasks. In *Machine Learning for Structural Biology Workshop at NeurIPS 2020*, 2020.
- Goldberger, J., Hinton, G. E., Roweis, S., and Salakhutdinov, R. R. Neighbourhood components analysis. *NIPS*, 17: 513–520, 2004.
- Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks. In *ACL*, 2020.
- Hadsell, R., Chopra, S., and LeCun, Y. Dimensionality Reduction by Learning an Invariant Mapping. In *CVPR*, 2006.
- Hou, J., Adhikari, B., and Cheng, J. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, (8), 2018.
- Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *arXiv: 2005.00687*, 2020a.
- Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. Strategies for Pre-training Graph Neural Networks. In *ICLR*, 2020b.
- Huang, K. and Zitnik, M. Graph meta learning via local subgraphs. In *NeurIPS*, 2020.
- Jurgens, D., Kumar, S., Hoover, R., McFarland, D., and Jurafsky, D. Measuring the Evolution of a Scientific Field through Citation Frames. *Transactions of the ACL*, 6, 2018.
- Klausen, M. S., Jespersen, M. C., Nielsen, H., Jensen, K. K., Jurtz, V. I., Sønderby, C. K., Sommer, M. O. A., Winther, O., Nielsen, M., Petersen, B., and Marcatili, P. NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins*, (6), 2019.
- Kulis, B., Sustik, M., and Dhillon, I. Learning low-rank kernel matrices. In *ICML*, pp. 505–512, 2006.
- Kulis, B. et al. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2012.

- Kupavskii, A. B. and Polyanskii, A. Proof of schur’s conjecture in \mathbb{R}^d . *Combinatorica*, 37(6):1181–1205, 2017.
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *ICLR*, 2019.
- Lee, J. D., Lei, Q., Saunshi, N., and Zhuo, J. Predicting What You Already Know Helps: Provable Self-Supervised Learning. *arXiv: 2008.01064*, 2020.
- Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., and Wang, P. K-bert: Enabling language representation with knowledge graph. In *AAAI Conference on Artificial Intelligence*, 2020.
- Liu, X., He, P., Chen, W., and Gao, J. Multi-Task Deep Neural Networks for Natural Language Understanding. In *ACL*, 2019.
- Mao, C., Zhong, Z., Yang, J., Vondrick, C., and Ray, B. Metric learning for adversarial robustness. *NeurIPS*, 2019.
- McDermott, M., Yap, B., Hsu, H., Jin, D., and Szolovits, P. Adversarial contrastive pre-training for protein sequences. *arXiv preprint arXiv:2102.00466*, 2021.
- McDermott, M. B. A., Nestor, B., Kim, E., Zhang, W., Goldenberg, A., Szolovits, P., and Ghassemi, M. A Comprehensive Evaluation of Multi-task Learning and Multi-task Pre-training on EHR Time-series Data. *arXiv: 2007.10185*, 2020.
- Min, S., Park, S., Kim, S., Choi, H.-S., and Yoon, S. Pre-Training of Deep Bidirectional Protein Sequence Representations with Structural Information. *arXiv: 1912.05625*, 2020.
- Mukherjee, A. Approximation Theorems and Whitney’s Embedding. In *Differential Topology*, pp. 43–67. Springer International Publishing, Cham, 2015. ISBN 978-3-319-19045-7. doi: 10.1007/978-3-319-19045-7_2.
- Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep Contextualized Word Representations. In *NAACL*, 2018.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv: 1910.10683*, 2019.
- Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. Evaluating Protein Transfer Learning with TAPE. In *NeurIPS*. Curran Associates, Inc., 2019.
- Rao, R., Meier, J., Sercu, T., Ovchinnikov, S., and Rives, A. Transformer protein language models are unsupervised structure learners. *bioRxiv preprint bioRxiv: 2020.12.15.422761*, 2020.
- Rocklin, G. J., Chidyausiku, T. M., Goreshnik, I., Ford, A., Houlston, S., Lemak, A., Carter, L., Ravichandran, R., Mulligan, V. K., Chevalier, A., Arrowsmith, C. H., and Baker, D. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347), 2017.
- Roth, K., Milbich, T., Sinha, S., Gupta, P., Ommer, B., and Cohen, J. P. Revisiting Training Strategies and Generalization Performance in Deep Metric Learning. In *ICML*, 2020.
- Ruder, S. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv: 1706.05098*, 2017.
- Sarkisyan et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603), 2016.
- Schroff, F., Kalenichenko, D., and Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, pp. 815–823, 2015.
- Shanehsazzadeh, A., Belanger, D., and Dohan, D. Is Transfer Learning Necessary for Protein Landscape Prediction? *arXiv: 2011.03443*, 2020.
- Shaw, B. and Jebara, T. Structure preserving embedding. In *ICML*, 2009.
- Shaw, B., Huang, B., and Jebara, T. Learning a Distance Metric from a Network. In *NeurIPS*, 2011.
- Shen, T., Mao, Y., He, P., Long, G., Trischler, A., and Chen, W. Exploiting structured knowledge in text via graph-guided representation learning. In *EMNLP*, Online, 2020.
- Suárez, J. L., García, S., and Herrera, F. A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges. *Neurocomputing*, 2020.
- Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., and Wang, H. ERNIE 2.0: A Continual Pre-training Framework for Language Understanding. In *AAAI Conference on Artificial Intelligence*, 2019.
- Tenney, I., Das, D., and Pavlick, E. BERT Rediscovered the Classical NLP Pipeline. In *ACL*, 2019.
- Vert, J.-P. and Yamanishi, Y. Supervised graph inference. In *NeurIPS*, 2004.

- Wang, K., Shen, Z., Huang, C., Wu, C.-H., Eide, D., Dong, Y., Qian, J., Kanakia, A., Chen, A., and Rogahn, R. A Review of Microsoft Academic Services for Science of Science Studies. *Frontiers in Big Data*, 2019a.
- Wang, S., Guo, Y., Wang, Y., Sun, H., and Huang, J. SMILES-BERT: Large Scale Unsupervised Pre-Training for Molecular Property Prediction. In *ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, 2019b.
- Wang, X., Han, X., Huang, W., Dong, D., and Scott, M. R. Multi-Similarity Loss With General Pair Weighting for Deep Metric Learning. In *CVPR*, 2019c.
- Weinberger, K. Q. and Saul, L. K. Fast solvers and efficient implementations for distance metric learning. In *ICML*, pp. 1160–1167, 2008.
- Weinberger, K. Q. and Saul, L. K. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(2), 2009.
- Whitney, H., Eells, J., and Toledo, D. *Collected Papers of Hassler Whitney*, volume 1. Nelson Thornes, 1992.
- Wu, S., Zhang, H. R., and Ré, C. Understanding and Improving Information Transfer in Multi-Task Learning. In *ICLR*, 2020.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *NeurIPS*, 2019.
- Yoon, J., Zhang, Y., Jordon, J., and van der Schaar, M. VIME: Extending the Success of Self- and Semi-supervised Learning to Tabular Domain. In *NeurIPS*, 2020.
- Zhang, D., Yin, J., Zhu, X., and Zhang, C. Homophily, structure, and content augmented network representation learning. In *ICDM*, 2016.
- Zheng, W., Chen, Z., Lu, J., and Zhou, J. Hardness-aware deep metric learning. In *CVPR*, pp. 72–81, 2019.
- Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Generalizing graph neural networks beyond homophily. In *NeurIPS*, 2020.
- Zitnik, M., Sosič, R., Feldman, M. W., and Leskovec, J. Evolution of resilience in protein interactomes across the tree of life. *Proceedings of the National Academy of Sciences*, 116(10), 2019.

A. Code and Data Availability

Our preliminary, anonymized code, synthetic datasets, and pointers to real-world datasets are available here: https://github.com/mmcdermott/structure_preserving_pre-training

B. LMPT on Natural Language vs. General Domains

As discussed in the introduction, many per-sequence NLP tasks can be re-framed as LM problems. In particular, suppose we have a prediction task $T : \mathbf{X} \rightarrow \mathbf{y}$, $\mathbf{y} \in \mathcal{Y}^n$ which operates at a per-sequence level, and has *free-text description* of the labels D (e.g., if our task is to predict review sentiment as in Bao et al., our description D could be “review sentiment”) and class descriptions C_y , $y \in \mathcal{Y}$ (e.g., if 1 indicates positive sentiment, then C_1 = “positive” and C_0 “negative”). With these free-text descriptions, any labeled pair (\mathbf{x}, \mathbf{y}) can be represented as a single NLP sequence via: “[\mathbf{x}] has D C_y]. For example: “[I loved the food!] has review sentiment positive].

C. General Pair Weighting in our SPPT

Recall that Wang et al.’s General Pair-Weighting (GPW) shows that any tuple-based supervised metric learning loss can be realized as below, for some value of weights w_{ij} :

$$\mathcal{F}_{\text{SPPT}} = \sum_{i=1}^m \left(\sum_{y_i \neq y_j}^m w_{ij} S_{ij} - \sum_{y_j = y_i}^m w_{ij} S_{ij} \right).$$

Here, we show that we can form an analogous SPML Pair-Weighting framework.

From Supervised- to Structure-Preserving Metric Learning. Given any supervised metric learning dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$, $\mathbf{y} \in \mathcal{Y}^n$, we can realize a graph-guided metric learning problem with identical optimal solution via the graph $G_{\mathbf{X}, \mathbf{y}} = (\mathbf{X}, \{(\mathbf{x}_i, \mathbf{x}_j) \in \mathbf{X} \times \mathbf{X} | y_i = y_j\})$. Note that this graph contains a series of disconnected cliques, with the set of cliques bijective with the set of class labels \mathcal{Y} .

Per the GPW (Wang et al., 2019c), any pair-based loss is dependent on \mathbf{y} only through the agreement statistics \mathbf{y} (e.g., $y_i = y_j$, $y_i \neq y_j$). Let \mathbf{A} be the adjacency matrix of $G_{\mathbf{X}, \mathbf{y}}$. Then, we can generalize these to reflect adjacency matrix operations $\mathbf{A}_{ij} = 1$ or $\mathbf{A}_{ij} = 0$, respectively, and in particular realize any supervised metric learning loss criteria as a “natural” structure-preserving analog.

Structure-Preserving Pair Weighting. In the manner discussed previously, we can therefore translate any pair-based supervised metric learning loss into a structure-preserving

loss. In particular, if we let \tilde{A} denote the ‘‘inverted’’ adjacency matrix, $\tilde{A} = \mathbf{1} - A$, and W be the matrix with elements w_{ij} , then the structure-preserving pair weighting is realized via:

$$\begin{aligned}\mathcal{F}_{\text{SPPT}} &= \sum_{i=1}^m \left(\sum_{y_i \neq y_j}^m w_{ij} S_{ij} - \sum_{y_i = y_j}^m w_{ij} S_{ij} \right) \\ &= \sum_{i=1}^m \left(\sum_{(x_i, x_j) \notin E}^m w_{ij} S_{ij} - \sum_{(x_i, x_j) \in E}^m w_{ij} S_{ij} \right) \\ &= \sum_{i=1}^m \left(\sum_{j=1}^m [\tilde{A} \odot W \odot S]_{ij} - [A \odot W \odot S]_{ij} \right) \\ &= \mathbf{1}^T \cdot (\tilde{A} \odot W \odot S - A \odot W \odot S) \cdot \mathbf{1}.\end{aligned}$$

This shows that we can realize a direct analogue of any pair- or tuple-based metric learning loss in the SPPT context in such a manner that under the restriction to graphs corresponding to class labels y , the SPPT loss is identical to that of the original metric-learning loss.

C.1. Contrastive Loss in SPML

Following Hadsell et al., we use contrastive loss in our Structure-Preserving Metric Learning (SPML). For this loss, we assume we are given the following mappings: ‘pos’, which maps x into a positive node (i.e., linked to x in graph G), and ‘neg’, which maps x into a negative node (i.e., not linked to x in graph G). The union of a set of points x and its images under pos and neg mappings then form our full minibatch.

Note this presentation is slightly different from Hadsell et al.’s presentation, but it is mathematically equivalent. This loss is specified by the positive and negative margin parameters μ_+ and μ_- as:

$$\begin{aligned}\mathcal{L}_{\text{SPML}}^{(\text{CL})} &= \frac{1}{N} \sum_{x_i \in X} \max(\mathcal{D}(x_i, \text{pos}(x_i)) - \mu_+, 0) \\ &\quad + \frac{1}{N} \sum_{x_i \in X} \max(\mu_- - \mathcal{D}(x_i, \text{neg}(x_i)), 0).\end{aligned}$$

D. Further Theoretical Analysis of SPML

Next, we derive theorems on the existence of optimal embeddings of sequences.

D.1. Preliminaries

Let $G = (V, E)$ be an arbitrary graph, $d_e \in \mathbb{N}$ be an embedding dimension, \mathcal{D} be the Euclidean distance. We will say that there exists an SPML-optimal embedding of G if there exists a positive radius $r \in \mathbb{R}^+$ and mapping $f : V \rightarrow \mathbb{R}^{d_e}$

such that $\mathcal{D}(f(v_1), f(v_2)) < c$ if and only if $(v_1, v_2) \in E$. Note that any SPML-optimal embedding also yields a margin parameter $m = \min_{(v_i, v_j) \notin E} (\mathcal{D}(f(v_i), f(v_j))) - \max_{(v_i, v_j) \in E} (\mathcal{D}(f(v_i), f(v_j)))$ where $m \in \mathbb{R}^+$ given G is finite and f is an optimal embedding.

Theorem 1. *Given arbitrary $G = (V, E)$, $d_e \in \mathbb{N}$, suppose that there exists an SPML-optimal embedding of G in \mathbb{R}^{d_e} . Then, for any $r \in \mathbb{R}^+$ or $m \in \mathbb{R}^+$, there exists an SPML-optimal embedding of G with radius r or with margin m .*

Proof. By the definition of an SPML optimal embedding, there exists some $r' \in \mathbb{R}^+$ and mapping $f' : \mathbb{R} \rightarrow d_e$ such that $\mathcal{D}(f'(v_1), f'(v_2)) < r'$ if and only if $(v_1, v_2) \in E$. Then, by the linearity (up to sign) of the norm operator, we can see that $f : v \mapsto \frac{r}{r'} f'(v)$ and $f : v \mapsto \frac{m}{m'} f'(v)$ will yield optimal embeddings with radius r or margin m , respectively. \square

As a result of Theorem 1, we can restrict our attention in subsequent sections to establishing the existence of any SPML optimal embedding, rather than on constructing a specific embedding.

D.2. Optimal Embeddings for Arbitrary Graphs

Lemma 1. *Let $n \in \mathbb{N}$, and let S consist of the vertices of any non-degenerate, n -dimensional simplex in \mathbb{R}^n , with edge lengths given by ℓ_{s_1, s_2} , $(s_1, s_2) \in S \times S$. There exists a strictly positive real number $\varepsilon_{s_1, s_2} \in \mathbb{R}^+$ and transformations h_{s_1, s_2}^+ , h_{s_1, s_2}^- of the simplex that preserves both the non-degeneracy of the simplex and the lengths of all edges ℓ_{s_i, s_j} ($(i, j) \neq (1, 2)$) except for ℓ_{s_1, s_2} , which is modified by ε_{s_1, s_2} , where h^+ extends the length and h^- contracts it.*

Proof. Let $S_{-1} = \{s \in S | s \neq s_1\}$. As S is a non-degenerate simplex in \mathbb{R}^n , $|S| = n + 1$, so $|S'| = n$. Thus, S' defines a hyperplane in \mathbb{R}^n . S_{-2} similarly defines a different hyperplane. These two hyperplanes intersect at the affine subspace $A \subset \mathbb{R}^n$ of dimension $n - 2$ spanned by $\{s \in S | s \notin \{s_1, s_2\}\}$. We can construct a function that realizes a rotation of S_{-2} about A , while preserving S_{-1} .

Note that all edges (s_i, s_j) , for $(i, j) \neq (1, 2)$ are contained entirely within either S_{-1} or S_{-2} . To see this, note that if neither i, j are equal to 1, 2, respectively, then $s_i, s_j \in S_{-1} \cap S_{-2}$. If $i = 1$, then $j \neq 2$ and $s_i, s_j \in S_{-2}$. If $j = 2$, then $i \neq 1$ and $s_i, s_j \in S_{-1}$.

Thus, rotating S_{-2} while preserving S_{-1} will preserve the lengths ℓ_{s_i, s_j} , for $(i, j) \neq (1, 2)$. As S is non-degenerate, $S_{-1} \nparallel S_{-2}$. This means there is some angle $\varphi > 0$ between them. Thus we can choose to rotate S_{-2} either by $\frac{\varphi}{2}$ or $-\frac{\varphi}{2}$ without yielding a degenerate simplex. Further, these will induce changes of opposite signs $d_+, d_- \in \mathbb{R}^+$

in the length ℓ_{s_1, s_2} . Set $\varepsilon_{s_1, s_2} = \min(d_+, d_-)$. By the intermediate value theorem, there exists some angle of rotation $\theta \in [-\frac{\varphi}{2}, \frac{\varphi}{2}]$ that both extends and contracts ℓ_{s_1, s_2} by ε . \square

Theorem 2. *Given $G = (V, E)$, $d_e \in \mathbb{N}$, $d_e \geq |V| - 1$, there exists an SPML optimal embedding of G .*

Proof. We will prove this by construction. Let f be the function that maps V to the vertices of the $|V|$ -vertex regular simplex in \mathbb{R}^{d_e} (as $d_e \geq |V| - 1$, this is always possible). Note that all points in $f(V)$ will thus be equidistant from one another, and a (simplex) edge will exist between all pairs of points (v_1, v_2) within $f(V)$. By Lemma 1, we can now choose to (iteratively, in arbitrary order), shrink all the simplex edges (v_1, v_2) corresponding to graph edges $(v_1, v_2) \in E$ by non-zero values ε_{v_1, v_2} . The mapping f then realizes an optimal embedding. \square

For our next several proofs, let us recall that for arbitrary dimension d , $B_r(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^d$ is the open ball of radius r about \mathbf{x} : $B_r(\mathbf{x}) = \{\mathbf{x}' \in \mathbb{R}^d | \mathcal{D}(\mathbf{x}, \mathbf{x}') < r\}$. Given a set $S \subseteq \mathbb{R}^d$, we will further use notation $B_r(S) = \bigcap_{s \in S} B_r(s)$. Recall additionally that for any set $S \subseteq \mathbb{R}^d$, $\text{diameter}(S) = \sup_{s_1, s_2 \in S \times S} \mathcal{D}(s_1, s_2)$. Note that for a closed set $S = \overline{S}$, this supremum will be obtained by some pair $s_1^*, s_2^* \in S \times S$. Typically, for all of these notations, we d will be inferred from context.

Let S_r be a regular simplex on $N + 1$ points in \mathbb{R}^N of side length $r \in \mathbb{R}$. Let $\Delta_r = B_r(S_r)$ denote the *Reuleaux simplex*—the intersection of radius r balls around the vertices of S_r (Kupavskii & Polyanskii, 2017). We will prove several lemmas regarding these Reuleaux simplices.

Lemma 2. Δ_r has diameter r .

Proof. Clearly, as $S_r \subseteq \overline{\Delta_r}$ has edge length r , $\text{diameter}(\overline{\Delta_r}) \geq r$. However, the boundary of Δ_r is spanned by either (1) portions of hyper-spheres of corresponding to radius r spheres surrounding simplex vertices or (2) simplex vertices themselves. The maximum distance among any two such points on this boundary is exactly r , obtained precisely with any point on a hypersphere boundary to its corresponding simplex vertex. Thus, $\text{diameter}(\overline{\Delta_r}) \leq r$. So, $\text{diameter}(\overline{\Delta_r}) = \text{diameter}(\Delta_r) = r$, as desired. \square

Lemma 3. *Let $S \subset \mathbb{R}^{N-1}$ be a set of $N + 1$ points such that for all $s, s' \in S$, $\mathcal{D}(s, s') > r$. Then $\text{diameter}(B_r(S)) \leq r$.*

Proof. Consider $f : \mathbb{R} \mapsto \mathbb{R}$ defined by

$$f : d \mapsto \inf_{S \subseteq \mathbb{R}^{N-1} | |S|=N, \sum_{s, s' \in S} \mathcal{D}(s, s')=d} \text{diameter}(B_r(S)).$$

Stated in words, f for any given input $d \in \mathbb{R}$, f returns the minimum possible induced ball-intersection diameter of any set of N points whose sum of pairwise distances is d .

Next, consider a set X that meets our stated criteria of N points. Note that if we form a new set X' by preserving all but one pairwise distance elements, and the one that is not preserved is shrunk slightly, then $\text{diameter}(B_r(X))$ can only stay the same or increase. This is because we can form such a transformation by rotating about all but two elements of X , and the remaining two elements grow strictly closer together. But, this implies that the function f is non-increasing with d .

Furthermore, we know that $\text{diameter}(\Delta_r)$ is given by precisely r , and that S_r obtains the minimal possible sum of pairwise distances for any set that satisfies our constraints. This, combined with the non-increasing nature of f , completes the proof. \square

Corollary 1. *The set $S \subset \mathbb{R}^{N-1}$ of size N such that all pairwise distances within S are strictly greater than $r \in \mathbb{R}$ that obtains minimal associated covered diameter $\text{diameter}(B_r(S))$ is the regular simplex S_r , with the associated Reuleaux Simplex Δ_r obtaining diameter r .*

Proof. This is a direct consequence of Lemmas 2 and 3. \square

Corollary 2. *It is impossible to place two sets of N points, $R = \{r_1, \dots, r_N\}$, $S = \{s_1, \dots, s_N\}$ in \mathbb{R}^{N-1} such that there exists some $r \in \mathbb{R}$ such that for all $i \neq j$, $\mathcal{D}(r_i, r_j), \mathcal{D}(s_i, s_j) > r$ and for all $1 \leq i, j \leq N$, $\mathcal{D}(r_i, s_j) \leq r$.*

Proof. Note that $\text{diameter}(S), \text{diameter}(R) > r$, but $\text{diameter}(B_r(S)), \text{diameter}(B_r(R)) \leq r$ based on Corollary 1.

Further, note that $S \subseteq B_r(R)$ (and vice versa). Further, recall that as $B_r(R)$ is the non-empty intersection of finitely many open sets, it is open. But, this immediately establishes that our claim, as thus placing the sets of points as desired would involve placing a set of diameter greater than r strictly inside an open set of diameter at most r , which is impossible. \square

Theorem 3. *Given embedding dimension $d_e \in \mathbb{N}$, there exists a graph $G = (V, E)$ with $2(d_e + 1)$ nodes that permits no optimal SPML embedding.*

Proof. Let $N = d_e + 1$. Then, let $G = (V, E)$ be the bipartite graph with $V = \{r_1, \dots, r_N, s_1, \dots, s_N\}$, and $E = \{(r_i, s_j) | 1 \leq i, j \leq N\}$.

By Corollary 1, it is impossible to arrange the subsets $R = \{r_1, \dots, r_N\}$ and $S = \{s_1, \dots, s_N\}$ in the desired fashion. \square

D.3. Optimal Embeddings for Manifolds

Definition D.1. Let \mathcal{M} be a manifold of dimension N embedded in \mathbb{R}^{2N} , where such an embedding is guaranteed to exist by the Whitney Embedding Theorem (Whitney et al., 1992; Mukherjee, 2015). Note that for any point $x \in \mathcal{M}$, with its corresponding embedding $\pi(x) \in \mathbb{R}^{2N}$, there exists a radius $r_x \in \mathbb{R}^+$ such that the ball of radius r around $\pi(x)$ in \mathbb{R}^{2N} intersection $B_{r_x}(\pi(x)) \cap \pi(\mathcal{M})$ is isomorphic both to some ball around x in \mathcal{M} and to a ball of radius r' in the copy of \mathbb{R}^N isomorphic to the neighborhood of x on \mathcal{M} . Let us call the manifold \mathcal{M} valid if $\inf_{x \in \mathcal{M}}(r_x) > 0$.

Theorem 4. Let X be a set of points distributed along a valid, d_m dimensional manifold \mathcal{M} . Let $G : (V, E)$ be a nearest neighbor graph over X on manifold \mathcal{M} —e.g., $|V| = |X|$ and there exists some isomorphism $m : V \rightarrow X$, such that $(v_1, v_2) \in E$ if and only if there exists some geodesic along the manifold \mathcal{M} between v_1 and v_2 and that the length of this geodesic ($\mathcal{D}_{\mathcal{M}}(m(v_1), m(v_2))$) is less than r . Then, there exists an SPML-optimal embedding of G with as few as $d_e = 2d_m$ dimensions.

Proof. This is a direct consequence of the definition (above) of a valid manifold. As $r^* = \inf_{x \in \mathcal{M}} r_x > 0$, we know that we can select some strictly positive $c \leq r^*$ such that all neighborhoods within c of any point in the embedding of the manifold only intersect the manifold in a manner that preserves geodesic distance. This establishes the claim. \square

E. Further Details on Synthetic Graph Construction for Task Topology Relationships

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 as described in the main body, with a base cycle and motifs distributed along that cycle evenly.

Node Featurization Nodes in this graph are then assigned internal features based on three notions of graph topology. For the “Homophily” 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 homophily, 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 SPPT system, as it would not be useful if SPPT could only capture data in contexts where the graph was perfectly reflected by the node features themselves.

F. Further Details on Fine-Tuning Tasks

F.1. Proteins

The tasks in the TAPE benchmark (Rao et al., 2019) on which we test are described more fully below:

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 (Hou et al., 2018).

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 (Klausen et al., 2019).

Stability This is a per-sequence, continuous regression problem, evaluated using 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 (Rocklin et al., 2017).

Fluorescence This is a per-sequence, continuous regression problem, evaluated using 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 (Sarkisyan et al., 2016).

F.2. Scientific Articles

The tasks in the SciBERT benchmark (Beltagy et al., 2019) 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).

Paper Field This problem tasks 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 original derived from the MAG (Wang et al., 2019a), it was to the best of our knowledge formulated into this task format by SciBERT directly (Beltagy et al., 2019).

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 (Cohan et al., 2019).

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 (Jurgens et al., 2018).

G. Further Details on Architecture and Hyperparameters

Synthetic Experiments. The Cliques and Mechanistic experiments in this domain 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 be as small as possible while still permitting reasonable learning dynamics.

Real Data Experiments. The architectures of our encoders are fully determined from our source models in TAPE (Rao et al., 2019) and SciBERT (Beltagy et al., 2019). In particular, for proteins and scientific articles, we use a 12-layer Transformer with a hidden size of 768, 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 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 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.

Fine-tuning hyperparameters (learning rate, batch size, and 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 LMPT model in (McDermott et al., 2021), though additional limited hyperparameter searches were performed to validate that these choices were

Table 4. Final hyperparameters for our protein dataset. All tasks used 200 total epochs and performed early stopping after 25 epochs of no validation set improvement. LR, learning rate.

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 scientific articles dataset. All models used a batch size of 32 and no early stopping, to match the original SciBERT paper (Beltagy et al., 2019). LR, learning rate. A / B = [LMPT Hyperparameter] / [SPPT Hyperparameter].

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

adequate. As the original source for these hyperparameters was an LMPT model, any bias here should be *against* SPPT, 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 given the limitations of the underlying machine. For the PLUS reproduction, we additionally compared hyperparameters analogous to the reported PLUS hyperparameters for other tasks as well as analogous to our hyperparameters for other tasks and used those that performed best on the validation set. For scientific articles, we performed grid search to optimize downstream task performance on the validation set, with learning rate varying between 5e-6 and 5e-5 and number of epochs between 2 and 5. The same grid search was used in original SciBERT system. We additionally match the SciBERT benchmark by applying dropout of 0.1, using the Adam optimizer with linear warm-up and decay, a batch size of 32, and no early stopping.

Final hyperparameters for each downstream task are shown in Tables 4 for proteins and 5 for scientific articles.

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

H. Further Commentary on Link Retrieval Results

To choose the optimal value of λ_{SPML} for use at FT 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_{SPML} 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 Table 6 for the proteins dataset and Table 7 for scientific articles.

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 (Rao et al., 2019) for proteins and SciBERT (Beltagy et al., 2019) for scientific articles.

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

CS RoBERTa (*for scientific articles only*) Nodes are embedded via the Gururangan et al.’s DAPT CS RoBERTa model, which is another LMPT model over scientific abstracts which performed very well on ACL-ARC, the task on which SPPT does best in scientific articles.

SPPT (*for various values of λ_{SPML}*). Nodes are represented via SPPT PT models at the specified weighting. For proteins, all SPPT models are initialized from TAPE, but for scientific articles, we test against both initializing from SciBERT and from CS RoBERTa (as both are just different, domain-specific LMPT 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 SPPT improves retrieval performance over the baselines for proteins much more than it does for scientific articles. This is, admittedly, largely due to Gururangan et al.’s CS RoBERTa model’s surprisingly good performance without any modifications, however as we also compare

SPPT 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 SPPT will offer more significant advantages on non-natural language domains.

Table 6. PT set link-retrieval performance for a random baseline, the raw TAPE model, and SPPT for various weighting parameters λ_{SPML} 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 SPPT framework that yield better results than the strongest baseline, providing evidence that incorporating sequence-level relational information into PT (i.e., $\lambda_{\text{SPML}} > 0$) leads to improved performance.

Method	λ_{SPML}	LRAP	nDCG	AP	MRR
Random Baseline	N/A	0.88%	27.1%	0.88%	0.003
TAPE (Rao et al., 2019)	N/A	8.50%	34.9%	2.41%	0.226
LMPT Baseline	0	8.92%	38.0%	2.33%	0.238
SPPT (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 7. PT set link-retrieval performance for a random baseline, the raw SciBERT model, and SPPT for various weighting parameters λ_{SPML} 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 SPPT framework that yield better results than the strongest baseline, providing evidence that incorporating sequence-level relational information into PT (i.e., $\lambda_{\text{SPML}} > 0$) leads to improved performance.

Method	λ_{SPML}	LRAP	nDCG	AP	MRR
Random Baseline	N/A	0.89%	26.0%	0.27%	0.016
SciBERT (Beltagy et al., 2019)	N/A	17.22%	52.8%	5.16%	0.272
LMPT Baseline (SciBERT initialized)	0	16.79%	35.4%	5.00%	0.271
DAPT CS RoBERTa (Gururangan et al., 2020)	N/A	32.56%	50.3%	12.86%	0.459
LMPT Baseline (CS RoBERTa initialized)	0	30.58%	48.3%	12.36%	0.438
SPPT (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
SPPT (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