How Useful is Continued Pre-Training for Generative Unsupervised Domain Adaptation?

Rheeya Uppaal[1] **Yixuan Li**[1] **Junjie Hu**[1,2]
[1]Department of Computer Sciences,
[2]Department of Biostatistics and Medical Informatics
University of Wisconsin-Madison
{uppaal, jhu, sharonli}@cs.wisc.edu

Abstract

Recent breakthroughs in scale have enabled the emergence of powerful generative language models, and the ability to fine-tune these models on various tasks by casting them into prompts or instructions. In this landscape, the problem of Unsupervised Domain Adaptation (UDA), or the problem of leveraging knowledge from a labeled source domain to an unlabeled target domain, has been left behind, with recent UDA methods still addressing discriminative classification. In particular, two popular UDA approaches, involving Continued Pre-Training (CPT) and learning domain invariant representations, have been under-explored in the generative setting, signaling a gap. In this work, we evaluate the utility of CPT for generative UDA. We first perform an empirical evaluation to measure the trade-offs between CPT and strong methods promoting domain invariance. We further evaluate how well the benefits of CPT extend to different architectures, tuning methods and data regimes. We then motivate the use of CPT by studying to what degree it benefits classification performance on the target domain. Finally, we attempt to understand the mechanism behind which CPT improves classification performance on the unlabeled target domain. Our findings suggest that the model implicitly learns the downstream task while predicting masked words informative to that task. Our work connects the body of UDA research with that of instruction tuning, enabling an initial step towards a wider applicability of modern language models. Our code is available at https://github.com/Uppaal/cpt-generative-uda.

1 Introduction

Recent advancements in the pre-training of language models have enabled the widespread use of powerful generative models, which can be leveraged across multiple domains with no training (Brown et al., 2020; Scao et al., 2022; Touvron et al., 2023, *inter alia*). Despite these advancements, these autoregressive models are still fragile under certain kinds of data distribution shifts, making their applications across these domains challenging (Ribeiro et al., 2020; Bajaj et al., 2021; Chuang et al., 2023; Uppaal et al., 2024, *inter alia*). This is addressed, in part, by the concept of instruction tuning with templates (Zhang et al., 2023; Sanh et al., 2022; Ouyang et al., 2022; Wang et al., 2022; Wei et al., 2022), enabling the learning of new tasks without any randomly initialized parameters.

The problem of unsupervised domain adaptation (UDA) leverages learned knowledge from a labeled source domain to an unlabeled target domain (Pan and Yang, 2010; Ganin and Lempitsky, 2015; Long et al., 2015, *inter alia*). It is useful for adaptation to unlabeled domains with high labeling costs, where supervised instruction tuning does not suffice. Despite the pervasive need for models to generalize to such domains, recent UDA methods still address discriminative classification, barring the application of these approaches to recent generative models. In particular, Continued Pre-Training and Domain Invariance-based methods, two widely popular classes of UDA approaches (Ramponi and Plank, 2020), are completely unexplored for UDA in the generative setting.

The Continued Pre-Training (CPT) approach involves extended pre-training on a domain or task, followed by supervised training on the downstream task (Gururangan et al., 2020). This approach has been widely used for adaptation to labeled domains (Gao et al., 2021; Kim et al., 2021; Hung et al., 2023, *inter alia*), and in the UDA setup for unlabeled domains (Han and Eisenstein, 2019; Zhang et al., 2021b; Karouzos et al., 2021; Pfeiffer et al., 2020; Parović et al., 2023). Invariance-based approaches attempt to learn representations that are invariant across domains (Tzeng et al., 2014; Ganin et al., 2016; Wu and Shi, 2022; Guo et al., 2022), with the notion that when the learned representa-

Proceedings of the 9th Workshop on Representation Learning for NLP (RepL4NLP-2024), pages 99–117

tions from both domains cannot be distinguished by a classifier and the classifier performs well on the source domain, it will also exhibit strong performance on the target domain. These two classes of methods introduce a trade-off: invariance-based methods suffer from instability issues (Han and Eisenstein, 2019; Kashyap et al., 2021), while continued pre-training requires a larger computational budget. But how would this trade-off extrapolate to the generative setting? For example, invariance-based methods are well motivated in discriminative tasks, where there is a clear decision boundary; however, the same does not hold for generative tasks.

To address these gaps, we introduce the setting of Generative UDA, where an autoregressive model is trained to leverage knowledge from a labeled source domain to an unlabeled target domain, *using only next word prediction as its objective*. We formalize the use of CPT for this setting in Section 2, and then attempt to explore and understand the behaviour of CPT for Generative UDA. We begin by performing an empirical analysis on 40 real-world domain pairs to explore the tradeoff between continued pre-training and invariance-based approaches, and find vanilla CPT to be competitive with and significantly more stable than a state of the art invariance-based approach (Section 3). We then stress test CPT, by applying it across varying model architectures and scales, tuning approaches and data regimes. We find that CPT is robust across these settings, unlike our invariance-based approach (Section 4).

With recent language models being trained across vast corpora which may include domains similar to the target domain, the requirement for continued pre-training may be raised to question. In Section 5, we show that continued pre-training is indeed essential for strong downstream performance on the target domain, and this performance rapidly degrades with limited target domain exposure. Finally, we attempt to shed light on how masking plays a role in improving classification accuracy on the unlabeled target domain in Section 6. We find that the model may implicitly learn the downstream task as it predicts masked words that are informative to the downstream task.

Our work attempts to connect the body of UDA research with recent trends in language modeling, by providing a set of insights into the behaviour of the popular class of continued pre-training approaches, in the Generative UDA setting. We hope this enables an initial step towards a wider applicability of modern language models.

2 Continued Pre-Training for Generative UDA

2.1 Preliminaries: The UDA Problem

We consider a text classification task, where $\mathcal{X}$ is the input space of all text sentences and $\mathcal{Y} = \{1, ...K\}$ is the label space. In the UDA problem, we have access to a source labeled dataset $\mathcal{D}_{\text{src}} = \{(x_i, y_i)\}_{i=1}^{N}$ consisting of samples from a joint distribution P_{src}, and a target unlabeled dataset $\mathcal{D}_{\text{tgt}} = \{x_j\}_{j=1}^{M}$ sampling from a target input distribution $P_{\text{tgt}}^{\mathcal{X}}$. We further denote $P_{\text{src}}^{\mathcal{X}}$ as the marginal distribution of P_{src} on $\mathcal{X}$, where $P_{\text{src}}^{\mathcal{X}} \neq P_{\text{tgt}}^{\mathcal{X}}$. The goal of UDA is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that minimizes the error rate $\mathbb{E}_{x \sim P_{\text{tgt}}^{\mathcal{X}}} \mathbb{1}[f(x) \neq y]$.

2.2 CPT for UDA as a Sequence of Prompt Based Tasks

We now formalize the extension of continued pre-training to the setting of generative UDA. We use the traditional two-phase training pipeline from Gururangan et al. (2020)[1]. The first phase uses templates to cast the source and target domain sequences into an autoregressive pre-training task.[2] The second phase applies supervised instruction-tuning of the model on source-labeled data.

Task 1: Autoregressive Continued Pre-training We reuse the input sequences from the source-labeled dataset $\mathcal{D}_{\text{src}}$ as the source-unlabeled dataset, denoted as $\mathcal{D}_{\text{src}}^{\mathcal{X}}$. Next, similar to Raffel et al. (2020); Song et al. (2019), for an unlabeled sequence $x \in \mathcal{D}_{\text{src}}^{\mathcal{X}}$ and $\mathcal{D}_{\text{src}}$, we use a prompt template to convert the sequence x to an input-output sequence pair, i.e., $\mathbb{M}(x) = (\tilde{x}, \tilde{y})$. For masked language modeling (MLM), an instruction is prepended to a randomly masked sequence x to create $\tilde{x}$. The output sequence $\tilde{y}$ is a concatenation of masked words from x. For example, given $x =$*"The movie was so cool! Two hours of fun."*, we construct $\tilde{x} =$*"Fill in the blanks: "The _ cool!*

[1]While we use the two-phase multi-task training pipeline (sequential) in our main experiments, in Appendix I, we show that an equivalent single-phase multi-task training pipeline (joint) results in similar performance.

[2]We investigate mask language modeling for T5 models and switch to causal language modeling for decoder-only models with a few simple template changes. We compare both in Section 4.

Two hours _", and $\tilde{y}$ = *"<sep> movie was so <sep> of fun. <sep>"*. For causal language modeling (CLM), $\tilde{x} = x$.

Given $(\tilde{x}, \tilde{y})$, we train an autoregressive LM parameterized by θ to minimize the negative log-likelihood loss averaged over output words and the total loss over a corpus $\mathcal{D} = \mathcal{D}^{\mathcal{X}}_{\text{src}} \cup \mathcal{D}_{\text{tgt}}$.

$$\ell(\tilde{x}, \tilde{y}; \theta) = -\frac{1}{|\tilde{y}|} \sum_{t} \log P_\theta(\tilde{y}_t | \tilde{x}, \tilde{y}_{1:t-1}) \quad (1)$$

$$\mathcal{L}_{\text{CPT}}(\mathcal{D}; \theta) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \ell(\mathbb{M}(x); \theta)$$

Task 2: Source Supervised Instruction-tuning In the second phase, we use labeled data from the source domain to train the model on the downstream classification task. Similar to the first phase, we use prompts[3] to generate input-output sequence pairs: $\mathbb{C}(x, y) = (\tilde{x}, \tilde{y}) \; \forall \; (x, y) \in \mathcal{D}_{\text{src}}$. For example, for sentiment classification, if x =*"I like this movie."*, $y = 1 \Rightarrow \tilde{x}$ = *"[x] Is this sentence positive or negative?"*, $\tilde{y}$ = *"Positive"*.

Given the augmented sequence pair $(\tilde{x}, \tilde{y})$ and the model trained after the first phase, we compute the same negative log-likelihood loss $\ell(\tilde{x}, \tilde{y}; \theta)$ in Eq. (1). Finally, we define the total loss on the source-labeled dataset in the second phase as:

$$\mathcal{L}_{\text{CLS}}(\mathcal{D}_{\text{src}}; \theta) = \frac{1}{|\mathcal{D}_{\text{src}}|} \sum_{(x,y) \in \mathcal{D}_{\text{src}}} l(\mathbb{C}(x, y); \theta) \quad (2)$$

After training, we follow the practice of Liu et al. (2022) to convert a label string $\tilde{y}$ to its corresponding label y at test time for evaluation.

3 Evaluating the Efficacy of Continued Pre-training for Generative UDA

3.1 Experimental Setup

Datasets We use the MNLI and Amazon Review classification datasets, which are widely used UDA benchmarks (Malik et al., 2023; Karouzos et al., 2021; Guo et al., 2020). The MNLI corpus (Williams et al., 2018) contains sentence pairs across five genres: Travel (T), Fiction (F), Government (G), Slate (S), and Telephone (Te). The task classifies every sentence pair as entailment, neutral, or contradiction. The Multi-Domain Sentiment Analysis Dataset (Blitzer et al., 2007) contains binary sentiment reviews for different types of Amazon products. We use reviews from the Apparel (A), Baby (B), Books (Bo), Cameras (C), and Movies (M) domains. We evaluate a total of 40 pairs of source and target domains, across the two datasets. Appendix A contains more details about the datasets.

Models and Tuning Methods Our main experiments use the T5v1.1 base model and (IA)^3 (Liu et al., 2022) PEFT method. T5v1.1 is an improved version of the original T5 model (Raffel et al., 2020), and unlike the original T5 model, it is not trained on any supervised datasets. We then extend our evaluation to different model architectures (T0, GPT-2), tuning methods (full fine-tuning, adapters) and data regimes (Section 4).

Training Each training phase is 30,000 steps long for MNLI and 15,000 steps for the Amazon dataset. We use Adam, a batch size of 8, and a learning rate of 0.003. We set the maximum sequence length to 256 tokens. We use length normalization during evaluation, as proposed by Liu et al. (2022). For each experiment, we report the mean and standard deviation across 3 runs. More details can be found in Appendix B.

Baselines Since our goal is to study the behaviour of CPT for generative UDA, we compare it with a simple supervised baseline, and a state-of-the-art invariance-based approach.

- **Src+Tgt** (All labeled): We fine-tune the model using labeled data from both the source and target domains. This serves as an upper bound on target domain performance.
- **UDAPTER**: Malik et al. (2023) propose an invariance-based method that measures the multi-kernel maximum mean discrepancy (MK-MMD) (Gretton et al., 2012; Bousmalis et al., 2016) between source and target embeddings from each transformer layer and sums them to obtain an aggregate loss $\mathcal{L}_{\text{div}}$. The final loss is the weighted sum of $\mathcal{L}_{\text{div}}$ and the classification loss, i.e., $\mathcal{L} = \lambda \; \mathcal{L}_{\text{cls}} + (1 - \lambda) \; \mathcal{L}_{\text{div}}$, where λ gradually changes from 0 to 1 during training. Here, we use the embeddings from a model as it is being instruction tuned on the downstream classification task. Their method achieves state-of-the-art performance, and outperforms popular UDA approaches like DANN (Ganin et al., 2016) and DSN (Bousmalis et al., 2016); thus we only compare CPT with this approach.

[3]Prompt templates were selected from the Public Pool of Prompts (Bach et al., 2022).

3.2 Continued Pre-training is Competitive with Domain-Invariance Methods

Performance We compare CPT with other baselines over 40 domain pairs of the MNLI and Amazon Review datasets, and report the average accuracies over all pairs in Table 1. We see that CPT is competitive to UDAPTER. (Appendix C contains detailed results over 40 pairs and significance tests to check for competitiveness.) Interestingly, a visualization of sentence embeddings in Figure 8 (Appendix C) suggests that representations learned through CPT are not domain invariant. In addition to the MMD based method of Malik et al. (2023), we also compare CPT on one domain pair with other methods that promote domain invariance (DANN (Ganin et al., 2016), CORAL (Sun et al., 2017)) and weight interpolation (Ilharco et al., 2022) in Appendix C, further confirming the competitiveness of CPT.

Dataset	Src+Tgt	UDAPTER	CPT
Amazon	92.66 (0.45)	89.02 (2.17)	**89.34** (0.48)
MNLI	78.14 (0.25)	70.19 (1.71)	**74.12** (0.68)

Table 1: Avg. target-domain classification accuracy and standard deviation over 3 runs.

Stability CPT performs more stably than UDAPTER, with the invariance-based method often reporting a variance of over 20% across runs (Table 6 in Appendix C). For example, for the MNLI pair Fiction (F) → Government (G), minimizing UDAPTER yields a variance of 23.4% across runs. This observation is consistent with existing findings (Kashyap et al., 2021; Han and Eisenstein, 2019) that minimizing divergence measures like MMD, when combined with auxiliary task-specific loss functions, result in training instabilities and vanishing gradients. We discuss this in more detail in Appendix J.

4 How General are the Benefits of CPT?

Instruction tuning for large models is often performed through parameter-efficient fine-tuning (PEFT) on limited data. This tuning also applies to models of different scales and architectures (decoder-only and encoder-decoder). In this section, we evaluate the utility of CPT across these factors, using the A→M domain pair from the Amazon Reviews dataset.

CPT Helps Decoder-only Models. We extend our analysis from MLM with encoder-decoder language models, to causal language modeling (CLM) with decoder-only language models, using GPT-2 (medium) (Radford et al., 2019). We perform CLM in the first training phase by simply training the model for next-word prediction given the original sequence. Table 2 shows that CPT provides strong improvements on the target domain in comparison to the invariance-based baseline.

Method	Accuracy
Src+Tgt	79.8 (0.3)
UDAPTER	66.0 (1.1)
CPT	**75.8** (0.4)

Table 2: Performance of CPT with causal language modeling for the decoder-only GPT-2 model. CPT significantly outperforms the invariance-based method.

CPT Outweighs Invariance-based Methods for Instruction-tuned Models. We evaluate the performance of CPT over T5v1.1 XL (3B parameters) and the instruction-tuned T0 (3B parameters) (Sanh et al., 2022). Figure 1 (Table 8 of Appendix D) shows a wider gap between UDAPTER and CPT with higher model capacity, and this gap is further increased with instruction tuning. We hypothesize that this gap is due to the vast difference between the objectives of domain invariance and instruction tuning.

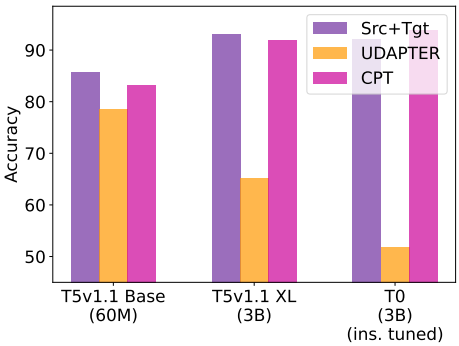

Figure 1: The performance gap between CPT and UDAPTER increases with larger models, from T5v1.1 Base (60M parameters) to T5v1.1 XL (3B parameters), and further increases with instruction tuning (T0 3B).

CPT Benefits are Consistent across Tuning Approaches. PEFT approaches have been shown to introduce resilience to domain shift (Fu et al., 2023). To isolate this effect from the CPT framework, we use T5v1.1 to evaluate CPT in a full fine-tuning setup. Additionally, we compare CPT with

two PEFT approaches[4] : Adapters (Houlsby et al., 2019) and (IA)^3 (Liu et al., 2022). We see in Figure 2 (Table 9 in Appendix E) that CPT continues to perform stronger than the domain invariance-based UDAPTER method.

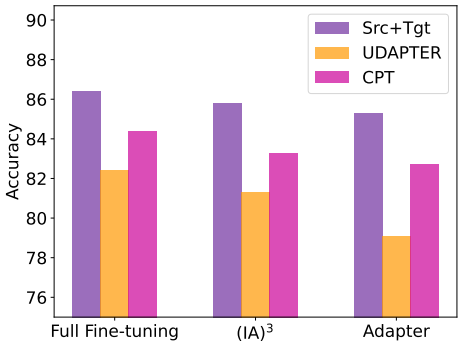

Figure 2: Performance of CPT across different tuning approaches with the T5v1.1 base model. CPT remains more powerful than UDAPTER across all tuning approaches.

CPT Outperforms Invariance-based Methods in Low-data Regimes. In this low-data experiment, we assume access to k labeled source-domain examples. For CPT, we assume access to the full unlabeled dataset in both domains for the first training phase, and k-shot access to labeled source-domain examples for the second phase of supervised training. For a fair comparison, we also introduce a two-phase version of the UDAPTER pipeline—the first phase minimizes MMD between unlabeled source and target domain embeddings (full data access), while the second phase optimizes supervised training on the source domain (k-shot). Figure 3 (Table 10 in Appendix F) showcases CPT clearly outperforming both variants of UDAPTER, across three different models for $k = 256$. Furthermore, Figure 4 (Table 11 in Appendix F) shows CPT providing consistent improvements in as low as 32-shots, unlike the unstable invariance-based approach.

5 To What Extent is Target-Domain Exposure Beneficial?

Given the vast distributions language models are pre-trained on, a natural assumption might be that the model has already been exposed to a domain similar to the target domain during pre-training. This would mean that the model could simply extrapolate the learned downstream task from the source to the target domain, questioning the need for CPT.

[4]We choose Adapters because He et al. (2022) present a unified view of PEFT approaches which shows that the operations applied by Adapters are very similar to those of Prefix Tuning (Li and Liang, 2021) and LoRA (Hu et al., 2022). We choose (IA)^3 since it is a state-of-the-art PEFT approach that uses a fraction of the learnable parameters of Adapters (More in Appendix E).

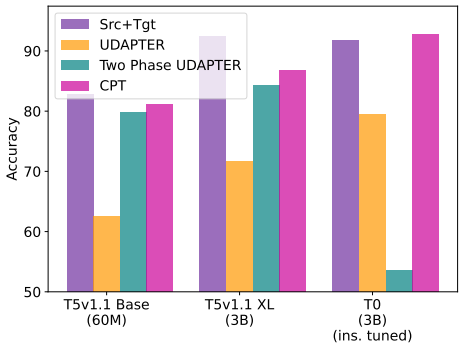

Figure 3: Performance of CPT across different models, in a 256-shot learning setup. Unlike both variants of UDAPTER, CPT is stable and provides consistent benefits across models.

In this section, we establish that exposure to the target domain *is* helpful, even when similar domains may have been encountered during pre-training. Table 3 evaluates CPT on the A→M domain pair with the T5v1.1 model. We note that the performance on the target domain is strongly impacted by the presence of target-domain data during the first phase of training.

Phase 1 Data	**Accuracy**	
	Source	Target
Source Only	93.3 (0.1)	76.5 (0.2)
Target Only	92.9 (0.4)	82.3 (0.7)
Source + Target	**93.5** (0.4)	**83.3** (0.5)

Table 3: Comparison of CPT with varying data exposure during the first phase of training. Performance on the target domain strongly benefits more from exposure to target domain, and is boosted further with exposure to the source domain.

For a more fine-grained analysis, we investigate the impact of degree of exposure to the target domain, by varying the masking rates during the first phase of training. While masking 15% of a sequence is considered standard for random masking, previous work has shown that BERT-sized models (Devlin et al., 2019) can learn from as high as 80% masking rates during pre-training followed by adaptation to a labeled task (Wettig et al., 2023). The source-domain performance shown in Figure 5 (Table 12 in Appendix) matches this trend. However,

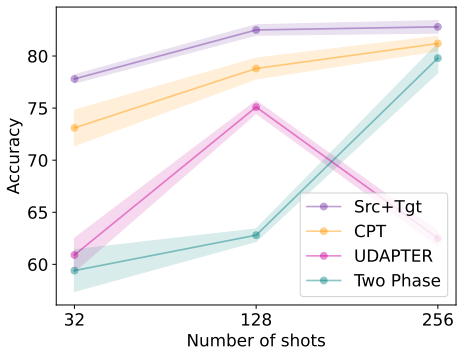

Figure 4: Few-shot performance of CPT, for varying k. Unlike both variants of UDAPTER, CPT is stable and provides consistent benefits across number of shots.

high masking rates effectively reduce the exposure of the model to target data, strongly deteriorating the performance on the target domain[5]. We hypothesize that since the model never sees any labeled data of the target domain, it heavily depends on the signal it gets from the unlabeled data through masking.

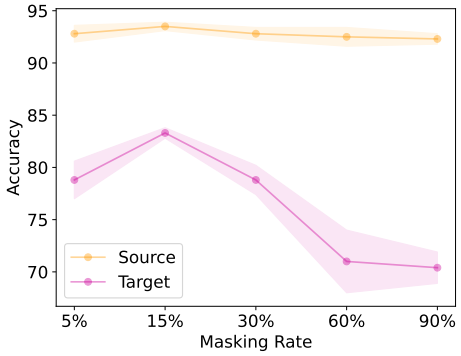

Figure 5: Impact of Masking Rate on CPT. With high masking rates, the performance on the source domain is largely maintained, but the performance on the target domain rapidly deteriorates.

6 Why does Word Prediction Aid Classification for Generative UDA?

In this section, we examine why predicting masked words of the source and target domains through CPT boosts sentence classification on the unlabeled target domain for generative UDA. We hypothesize that by having to predict masked words that are informative to the downstream task during pre-training, the model implicitly learns information about the downstream task. For example, given the masked sentence, *"I really _ the movie, it was a fascinating watch."*, the masked word is indicative of the downstream task, in this case sentiment analysis. The model can only predict this masked word (which would be a positive word like "*loved*" or "*enjoyed*") by using other words informative to the task ("*fascinating*"). Through this process, the model is essentially learning features which are useful to the downstream task, despite having no direct supervision.

To test this hypothesis, we quantize the "informativeness" of each word to a classification task: an informative word is highly correlated with any of the labels in the downstream task.[6] Specifically, we follow Gururangan et al. (2018) and use pointwise mutual information (PMI) (Fano, 1961) of the word with respect to the class label:

$$\text{PMI}(\text{word}, \text{class}) = \log \frac{p(\text{word}, \text{class})}{p(\text{word})p(\text{class})},$$

where we count the frequency of a word-class pair on $\mathcal{D}_{\text{src}}$ to estimate $p(\text{word}, \text{class})$, and similarly count a word and a class individually on $\mathcal{D}_{\text{src}}$ to estimate $p(\text{word})$ and $p(\text{class})$.

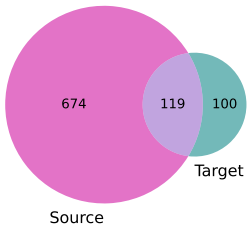

Figure 6: Vocabulary overlap between label-informative words of the source and target domains. The numbers in the Venn diagram indicate the number of words in both sets.

We use two sets of words from a dataset: those with the top $k\%$ (informative) and bottom $k\%$ (uninformative) PMI with any inference label ($k = 15$). We also filter out low-frequency words from the selection.[7] We compute these sets for the source and target domains individually, assuming access to target labels. We use the T5v1.1 model on the A→M pair for our analysis.

[5]With masking rates under the optimal value of 15%, the semantic and background features learned through model prediction of masked words is limited, hurting performance on the target domain.

[6]These informative words are similar to pivot features (Blitzer et al., 2006; Ziser and Reichart, 2018; Ben-David et al., 2020, *inter alia*), with the exception that they are chosen based on information from the source domain only.

[7]Any word that occurs less than 10 times in the entire training corpus is considered to be low frequency.

How Masking helps Learn Classification We first confirm that label-informative words indeed impact the classification performance of the CPT model. We do this by masking informative words from a sentence at inference. Figure 7 (a) shows us that the performance of the model on the source domain is not impacted by masking uninformative words, but drops on masking informative words. However, how do we know how much of this bias towards label informative words was learned during the continued pre-training phase, rather than during supervised fine-tuning? To attempt to disentangle the impact of the training phases, we train the model through selective masking (informative or uninformative) in the first phase of training, and minimize the impact of the second phase by making it a few-shot task. Figure 7 (c) shows us that the model performs best on classification when trained to predict label informative words during masking, indicating that the model does indeed learn features relevant to the downstream task during the first phase of training.

The Interplay between CPT and Classification for Generative UDA We now extend this analysis to the target domain to understand how CPT plays a role in learning features from the unlabeled domain. Figure 7 (d) shows us that informative masking outperforms uninformative masking by a significant gap, once again signaling that the masking process helps the model implicitly learn the downstream task. However, unlike with the source domain, random masking results in the strongest performance. This is due to the domain mismatch: the informative words for the source and target domains are not identical (Figure 6), and the supervised training on the source domain adds a bias towards source-informative words. The mixture of these two sets of words are best predicted through random masking, explaining its strong performance.

This phenomenon also draws the observation that random masking is preferred to selective masking for generative UDA, contrary to single domain settings where informative masking is more useful (Levine et al., 2021; Gu et al., 2020).

7 Discussion

The Computational Trade-off of CPT Our results in Section 3.2 show that continued pre-training and methods promoting domain invariance are competitive with each other. Continued pre-training suffers from the computational drawback of requiring an additional phase of training. Conversely, invariance-based methods are difficult to optimize, possibly requiring more runs to achieve a stable optimum, and having a higher amortized computational cost. Inspired by Karouzos et al. (2021), we introduce a simple single phase variant of continued pre-training which is *equivalent* in performance to its two phase variant, nullifying the additional computational overhead of the approach (more details in Appendix I).

UDA in the Age of LLMs Recent breakthroughs in scale have showcased that large language models (LLMs) are highly powerful, and can perform various downstream tasks with limited or no training. This may raise question on the relevance of the UDA problem as a whole — does a model even require expensive adaptation to a domain it may have already been exposed to during pre-training? In addition to our analysis in Section 5, we argue that the requirement to adapt small models to unseen domains still holds in specific cases. Small supervised models have been shown to be comparable with, or even outperform, zero-shot general-purpose LLMs on various downstream tasks (Huang et al., 2023; Zhu et al., 2023; Tang et al., 2024), serving as lightweight and customizeable (through fine-tuning) alternatives. Safety critical domains like healthcare and finance would benefit more from these models than a generalist LLM. Our study does not address how to better adapt to domains, rather we investigate ways a model may adapt to data unseen during pre-training. This is a question that holds for current LLMs, and will continue to hold as long as models are unable to access infinite data during pre-training.

8 Related Work

UDA through Promoting Domain Invariance A major class of approaches in Model-centric UDA methods (Ramponi and Plank, 2020) aims to minimize $\mathcal{H}\Delta\mathcal{H}$ divergence (Ben-David et al., 2010) between the source and target domain features, through adversarial training (Tzeng et al., 2014; Ganin et al., 2016; Tzeng et al., 2017; Guo et al., 2022, *inter alia*) or through minimizing measures of domain similarity (Bousmalis et al., 2016; Ge et al., 2023). Malik et al. (2023) have shown the minimization of MMD to outperform other invariance-based methods. However, past work has shown that domain-invariance is a weak constraint

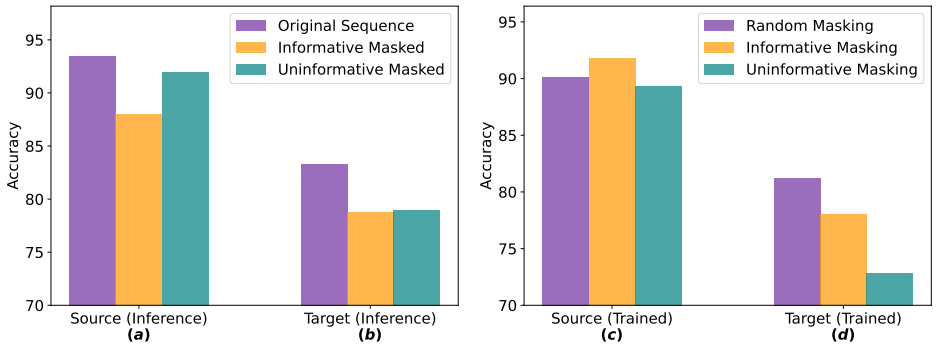

Figure 7: The impact of selective masking on classification performance of a CPT trained model. **Left**: Masking label informative words during inference degrades classification performance, compared to the original unmasked sequence. However, the removal of uninformative words does not impact the model on the source domain. **Right**: The impact of continued pre-training can be partially disentangled from that of supervised training by making the supervised training phase few-shot and training the model to mask informative or uninformative words during the continued pre-training phase. Informative masking is most beneficial for the source domain, indicating that the model learns task relevant features during masking. On the target domain, informative masking still captures some knowledge about the downstream task, however, the supervised training phase adds a bias towards source label informative words. Thus, random masking is most powerful.

for adaptation (Zhao et al., 2019; Karouzos et al., 2021), could introduce domain-specific hyperparameters (Trung et al., 2022), and is also prone to instability issues (Han and Eisenstein, 2019; Sun et al., 2019; Wilson and Cook, 2020; Kashyap et al., 2021).

UDA through Continued Pre-Training The limitations of invariance-based model-centric methods have encouraged the emergence of alternate approaches, based on self-supervised learning through contrastive learning (Kumar et al., 2022; Shen et al., 2022; Long et al., 2022), pseudo-labeling (Zhou and Li, 2005; Ruder and Plank, 2017, *inter alia*) or language model pre-training. Despite not being directly useful to certain downstream tasks (Uppaal et al., 2023), CPT has been used for adaptation to labeled tasks, in both full fine-tuning (Gururangan et al., 2020; Lee et al., 2020; Gao et al., 2021, *inter alia*) and PEFT setups (Kim et al., 2021; Hung et al., 2023). A smaller body of work has explored the utility of CPT in a UDA setup (Han and Eisenstein, 2019; Zhang et al., 2021b; Karouzos et al., 2021; Parović et al., 2023), identifying the class of methods to be more stable than invariance-based methods.

Generative UDA The emergence of large language models (Brown et al., 2020; Scao et al., 2022; Touvron et al., 2023, *inter alia*) introduced the concept of instruction tuning with templates (Zhang et al., 2023; Sanh et al., 2022; Ouyang et al., 2022; Wang et al., 2022; Wei et al., 2022; Gao et al., 2021; Liu et al., 2023), enabling multi-task training without any task specific architectural changes. However, the framework of casting discriminative classification tasks into generative next word prediction tasks has not yet been extended to UDA. The closest work to this setting (Ben-David et al., 2021) uses a generative model to create domain identifier prompts and feed them back into the model, however the final task label prediction is still discriminative. In our work, we focus on gaining insights to extend the powerful class of CPT methods to purely generative UDA, where prediction on the downstream task is treated as a next word prediction task. Through this, we also present novel findings on the impact of CPT to prompt-based classifiers in the UDA framework, countering previous findings from other studies in a single-domain setting (Gu et al., 2020; Levine et al., 2021; Wettig et al., 2023).

9 Conclusion

We introduce the setting of Generative UDA, and perform an investigation on the utility of continued pre-training in this setting. We compare the approach with the popular class of domain-invariance based methods for UDA, showing that CPT is both competitive with, and more stable than invariance-based approaches. Our experiments show that the

benefits of CPT extend to different architectures, tuning methods and data regimes. We motivate the need for target domain exposure through CPT by showing that performance on the target domain gradually degrades with increasing masking rate. Finally, we shed light on the interplay between masking and classification performance, and how this aids UDA. Our analysis shows that in predicting masked words that are informative to the downstream task, the model implicitly learns about the downstream task, furthering the benefits of directly learning the task. Our work connects the body of UDA research with that of instruction tuning, enabling an initial step towards a wider applicability of modern language models.

Limitations

Our work presents an investigation into continued pre-training for UDA in a *generative* setting. Since generative UDA is an almost completely unexplored area, we establish a proof of concept by using sentence classification tasks for our analysis. We leave the extending our analysis to more complex tasks to future work.

In our study, we consider a class of PEFT methods that involve inserting learnable parameters between the layers of the model. Other classes of PEFT methods were not considered. However, we use Adapters and He et al. (2022) have shown connections between the method with Prefix Tuning (Li and Liang, 2021) and LoRA (Hu et al., 2022).

Due to the high variance across runs in PEFT-based learning, we note that the performance can vary significantly across random seeds. We attempt to make our findings reproducible by averaging every experiment over 3 seeds. Taking environmental costs into consideration, we reduce our computational budget by running a majority of our experiments with a smaller-sized model. Learning with larger models is discussed in Section 4.

Ethics Statement

Our project aims to extend the problem of unsupervised domain adaptation to the generative setting, matching current needs with large language models. This is an effort towards improving the reliability and safety of language models, which can be fragile under distribution shift (Ribeiro et al., 2020) and incur great costs over incorrect predictions (Ulmer et al., 2020; Zhang et al., 2021a).

Our study does not involve any human subjects or violation of legal compliance. We do not anticipate any potentially harmful consequences to our work. As detailed in Appendix A, all of our experiments are conducted using publicly available datasets. Our code shall be released for reproducibility. Through our study and releasing our code, we hope to raise stronger research and societal awareness toward the problem of unsupervised domain adaptation in natural language processing.

References

Nachman Aronszajn. 1950. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Févry, et al. 2022. Promptsource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104.

Ahsaas Bajaj, Pavitra Dangati, Kalpesh Krishna, Pradhiksha Ashok Kumar, Rheeya Uppaal, Bradford Windsor, Eliot Brenner, Dominic Dotterrer, Rajarshi Das, and Andrew Mccallum. 2021. Long document summarization in a low resource setting using pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 71–80.

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2021. Pada: A prompt-based autoregressive approach for adaptation to unseen domains. *arXiv preprint arXiv:2102.12206*, 3.

Eyal Ben-David, Carmel Rabinovitz, and Roi Reichart. 2020. Perl: Pivot-based domain adaptation for pre-trained deep contextualized embedding models. *Transactions of the Association for Computational Linguistics*, 8:504–521.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79:151–175.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 con-*

ference on empirical methods in natural language processing, pages 120–128.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. *Advances in neural information processing systems*, 29.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yun-Shiuan Chuang, Rheeya Uppaal, Yi Wu, Luhang Sun, Makesh Narsimhan Sreedhar, Sijia Yang, Timothy T Rogers, and Junjie Hu. 2023. Evolving domain adaptation of pretrained language models for text classification. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Robert M Fano. 1961. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29(11):793–794.

Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12799–12807.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830.

Pengfei Ge, Chuan-Xian Ren, Xiao-Lin Xu, and Hong Yan. 2023. Unsupervised domain adaptation via deep conditional adaptation network. *Pattern Recognition*, 134:109088.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2020. Train no evil: Selective masking for task-guided pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6966–6974, Online. Association for Computational Linguistics.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2020. Multi-source domain adaptation for text classification via distancenet-bandits. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7830–7838.

Xu Guo, Boyang Li, and Han Yu. 2022. Improving the sample efficiency of prompt tuning with domain adaptation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3523–3537, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Kuan-Hao Huang, I Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Premkumar Natarajan, Kai-Wei Chang, Nanyun Peng, Heng Ji, et al. 2023. A reevaluation of event extraction: Past, present, and future challenges. *arXiv preprint arXiv:2311.09562*.

Chia-Chien Hung, Lukas Lange, and Jannik Strötgen. 2023. Tada: Efficient task-agnostic domain adaptation for transformers. *arXiv preprint arXiv:2305.12717*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Constantinos Karouzos, Georgios Paraskevopoulos, and Alexandros Potamianos. 2021. Udalm: Unsupervised domain adaptation through language modeling. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2579–2590.

Abhinav Ramesh Kashyap, Devamanyu Hazarika, Min-Yen Kan, and Roger Zimmermann. 2021. Domain divergences: A survey and empirical analysis. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1830–1849.

Seungwon Kim, Alex Shum, Nathan Susanj, and Jonathan Hilgart. 2021. Revisiting pretraining with adapters. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 90–99.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. 2021. Pmi-masking: Principled masking of correlated spans. In *International Conference on Learning Representations*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR.

Quanyu Long, Tianze Luo, Wenya Wang, and Sinno Pan. 2022. Domain confused contrastive learning for unsupervised domain adaptation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2982–2995.

Bhavitvya Malik, Abhinav Ramesh Kashyap, Min-Yen Kan, and Soujanya Poria. 2023. Udapter-efficient domain adaptation using adapters. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2241–2255.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Marinela Parović, Alan Ansell, Ivan Vulić, and Anna Korhonen. 2023. Cross-lingual transfer with target language-ready task adapters. *arXiv preprint arXiv:2306.02767*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in nlp—a survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Kendrick Shen, Robbie M Jones, Ananya Kumar, Sang Michael Xie, Jeff Z HaoChen, Tengyu Ma, and Percy Liang. 2022. Connect, not collapse: Explaining contrastive learning for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 19847–19878. PMLR.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Baochen Sun, Jiashi Feng, and Kate Saenko. 2017. Correlation alignment for unsupervised domain adaptation. *Domain adaptation in computer vision applications*, pages 153–171.

Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. 2019. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. *arXiv preprint arXiv:2404.10774*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Nghia Ngo Trung, Linh Ngo Van, and Thien Huu Nguyen. 2022. Unsupervised domain adaptation for text classification via meta self-paced learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4741–4752.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *conference on computer vision and pattern recognition*.

Dennis Ulmer, Lotta Meijerink, and Giovanni Cinà. 2020. Trust issues: Uncertainty estimation does not enable reliable ood detection on medical tabular data. In *Machine Learning for Health*, pages 341–354. PMLR.

Rheeya Uppaal, Apratim De, Yiting He, Yiquao Zhong, and Junjie Hu. 2024. Detox: Toxic subspace projection for model editing. *arXiv preprint arXiv:2405.13967*.

Rheeya Uppaal, Junjie Hu, and Yixuan Li. 2023. Is fine-tuning needed? pre-trained language models are near perfect for out-of-domain detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2985–3000, Dubrovnik, Croatia. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1112–1122.

Garrett Wilson and Diane J Cook. 2020. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46.

Hui Wu and Xiaodong Shi. 2022. Adversarial soft prompt tuning for cross-domain sentiment analysis. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2438–2447.

Oliver Zhang, Jean-Benoit Delbrouck, and Daniel L Rubin. 2021a. Out of distribution detection for medical images. In *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Perinatal Imaging, Placental and Preterm Image Analysis*, pages 102–111. Springer.

Rongsheng Zhang, Yinhe Zheng, Xiaoxi Mao, and Minlie Huang. 2021b. Unsupervised domain adaptation with adapter. *Proceedings of the Workshop on Efficient Natural Language and Speech Processing*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. 2019. On learning invariant representations for domain adaptation. In *International conference on machine learning*, pages 7523–7532. PMLR.

Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675*.

Yftah Ziser and Roi Reichart. 2018. Pivot based language modeling for improved neural domain adaptation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1241–1251.

A Preparation of Evaluation Benchmarks

We use two classification datasets, with 5 domains each. This results in a total of 40 pairs of source and target domains. For brevity, we include results of 24 domain pairs in the main paper, and the remaining 16 in Appendix C. For both datasets, we use the train, validation and test splits from (Malik et al., 2023). More statistics about each dataset is available in Table 4. The listed datasets are intended for research purposes only. We do not make any commercial use of them.

MNLI The Multigenre Natural Language Inference (MNLI) corpus (Williams et al., 2018) contains sentence pairs across multiple genres: Travel (T), Fiction (F), Government (G), Slate (S) and Telephone (Te). The NLI task involves classifying every premise-hypothesis sentence pair as Entailment, Neutral or Contradiction.

Amazon The Multi Domain Sentiment Analysis Dataset (Blitzer et al., 2007) contains Amazon product reviews for different type of products. We use reviews from the Apparel (A), Baby (B), Books (Bo), Cameras (C) and Movies (M) domains. Each review is labelled as positive or negative.

Dataset	Language	License	Statistics per Domain		
			Train	Val	Test
MNLI	English	cc-by-4.0	69600*	7735**	1945
Amazon	English	cc-by-4.0	1440	160	400

Table 4: Artifacts used in our study. The dataset statistics report the values used in our study.
* All domains contain approximately 69,600 examples. The exception is the Telephone domain, with 75,013 examples.
** All domains contain 7735 validation examples, except for Slate and Telephone, which contain 7731 and 8336 examples respectively.

B Details on Implementation

Models and Implementation We use T5v1.1, T0 and GPT-2 and LLaMA-2 from the HuggingFace library[8], and use PyTorch[9] to train our models.

Training We use the default hyperparameters from Liu et al. (2022), except for batch size and training duration. We perform a grid search for these values. We train each training phase for 30,000 steps on MNLI and 15,000 steps on the Amazon dataset, with a batch size of 8. For the T5v1.1 XL and T0 models (3B parameters each), we use a batch size of 1. We train with Adam and use a learning rate of 0.003. We set the maximum sequence length to 256 tokens. We use length normalization during evaluation, as proposed by Liu et al. (2022). For each experiment, we report the mean and standard deviation across 3 runs.

[8] https://github.com/huggingface/transformers
[9] https://pytorch.org/

Training Steps	Source Accuracy	Target Accuracy
5,000	93.2 (0.4)	81.9 (0.4)
10,000	93.4 (0.5)	81.6 (0.6)
15,000	93.5 (0.4)	83.3 (0.5)

Table 5: We use early stopping on one domain pair to determine the number of training steps, which we then use for all domain pairs of that dataset. For example, the Apparel→Movies domain pair of the Amazon Reviews dataset shown in the table saturates at 15,000 steps.

We choose the number of training steps based on early stopping on the validation set for one domain, and use that number of steps for all domains within that dataset. We report the test set performance after a varying number of training steps in the table below. For example, for the Apparel→Movies domain pair of the Amazon Reviews dataset, the performance saturates at 15,000 steps, as shown in Table 5.

Computations Using the $(IA)^3$ PEFT framework, training the T5v1.1 Base model (60 million parameters) for 15,000 steps takes approximately two hours on a single NVIDIA RTX A6000 GPU. The T5v1.1 XL model and T0 model (3 billion parameters) take approximately 8 hours for 15,000 steps of training. For reproducibility, each experiment is repeated thrice, with changing random seeds. In total, we run 540 experiments with the Base model and 72 experiments with the larger models. This results in a total compute time of approximately 2400 GPU hours.

C Detailed results with the Amazon and MNLI Datasets

Table 6 shows the performance of CPT on the Amazon and MNLI datasets.

On the Amazon dataset, CPT is competitive with the state of the art UDAPTER method from Malik et al. (2023) on average. We confirm this by checking for a significant difference in the performance of CPT and UDAPTER on the 20 dataset pairs. The Mann-Whitney U test and Student's t-test both re-

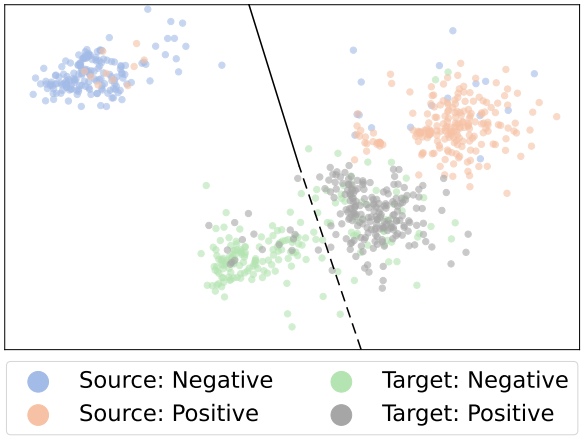

Figure 8: UMap visualizations of sentence embeddings from the Apparel → Movies data pair, using the T5v1.1 base model and $(IA)^3$ PEFT method. Despite not promoting domain-invariance, CPT may be learning sentence embeddings that are separable by class labels, regardless of the domain of these sentences. The classification hyperplane for the source domain has been imagined as a solid line for illustration purposes, and its extension to the target domain is shown as a dashed line.

sulted in non-significant p-values of 0.5516 and 0.8316, confirming the hypothesis that there is no significant difference between CPT and UDAPTER on the Amazon dataset.

However, on the MNLI dataset, where all domains have larger gaps, both significant tests showed a significant difference between CPT and UDAPTER, with CPT being more powerful. This is exemplified through cases like Travel (T) → Government (G), where CPT yields an accuracy of 83.6% on the target domain, equalling the upper bound of the Src+Tgt baseline.

Comparison with other Model Centric Approaches In addition to the MMD based method of Malik et al. (2023), we also compare CPT with other methods that promote domain invariance: 1) DANN (Ganin et al., 2016), which is the most widely used UDA method in NLP (Ramponi and Plank, 2020), but has been shown to be highly unstable; 2) CORAL (Sun et al., 2017), which minimizes second order statistics of the data embeddings. Additionally, with an emerging class of weight interpolation based methods, we make a comparison with task vector arithmetic (Ilharco et al., 2022). The use of task vectors with PEFT methods beyond LoRA (Hu et al., 2022) has been unexplored in the literature, and we find that the method does not work with IA3. With fully fine-tuned models, the method improves in performance, but is still weaker than CPT.

CPT may learn representations that generalize across domains To better understand the improved UDA performance, we visualize the sentence embeddings learned by CPT in Figure 8. Using UMap (McInnes et al., 2018), the figure visualizes embeddings for the Apparel→Movies domain pair from the Amazon Product Review dataset. We see that CPT learns sentence embeddings that generalize across domains. For illustration, we draw a black line that cuts across both source and target domains. Note that the solid line suggests that there exists a classification hyperplane learned on the source labeled data (in blue and green). The same classifier can be potentially used to separate target data (in gray and orange). The visualization suggests that CPT achieves competitive UDA results without having to explicitly promote domain-invariant representations.

D CPT across Model Architectures and Scales

We evaluate the performance of CPT over T5v1.1 XL and the instruction tuned T0 (3B) (Sanh et al., 2022) in Table 8.

E PEFT Frameworks

The framework proposed in Section 2 is general and can be applied to fine-tune all model parameters. Additionally, our CPT framework is compatible with the parameter-efficient fine-tuning approach. The PEFT approach is desirable because it adds only a small amount of learnable parameters ϕ to a pre-trained language model θ, and fine-tunes only ϕ to perform prediction while keeping the other model parameters θ frozen. We use two instantiations in our implementations: Adapters (Houlsby et al., 2019) and $(IA)^3$ (Liu et al., 2022).

$(IA)^3$ is a state of the art PEFT learning method, and uses around a tenth of learnable parameters compared to popular methods like Adapters. $(IA)^3$ works by element-wise multiplication (i.e. rescaling) of the model's activations against a learned vector. In this case, the set of learnable parameters ϕ is a set of vectors $\{l_v, l_k, l_{ff}\}$ applied to each attention mechanism and feed-forward layer as,

	Amazon				MNLI		
	Src+Tgt	**UDAPTER**	**CPT**		**Src+Tgt**	**UDAPTER**	**CPT**
A → B	94.7 (0.2)	93.8 (0.3)	**93.9** (0.3)	T → F	77.2 (0.4)	69.7 (0.8)	**74.1** (0.9)
A → Bo	94.3 (0.4)	**92.5** (1.1)	90.2 (1.2)	T → G	83.6 (0.7)	79.3 (0.5)	**83.6** (0.3)
A → C	95.0 (0.2)	91.8 (0.5)	**92.1** (0.5)	T → S	72.3 (0.5)	69.6 (0.1)	**70.7** (0.6)
A → M	85.8 (0.5)	81.3 (0.6)	**83.3** (0.5)	T → Te	77.8 (0.1)	69.4 (0.8)	**76.8** (0.0)
B → A	93.4 (0.3)	93.3 (0.2)	**93.4** (0.4)	F → T	79.9 (0.1)	**69.9** (0.2)	65.4 (0.8)
B → Bo	94.7 (0.7)	**93.8** (0.3)	92.2 (0.1)	F → G	82.3 (0.1)	54.3 (23.4)	**78.8** (2.5)
B → C	94.7 (0.8)	**93.4** (0.1)	92.1 (0.3)	F → S	72.1 (0.2)	64.6 (1.8)	**65.3** (1.6)
B → M	85.3 (0.2)	81.3 (0.7)	**82.8** (0.2)	F → Te	78.3 (0.6)	64.6 (0.7)	**72.5** (0.2)
Bo → A	94.6 (0.3)	**91.6** (0.5)	91.3 (0.2)	G → T	79.9 (0.4)	**75.9** (0.3)	75.8 (0.6)
Bo → B	94.8 (0.2)	**92.9** (0.6)	90.9 (0.2)	G → F	76.7 (0.1)	69.9 (0.2)	**73.5** (0.2)
Bo → C	94.3 (0.2)	89.8 (0.1)	**90.3** (0.4)	G → S	73.1 (0.0)	**69.4** (0.1)	68.0 (1.8)
Bo → M	85.5 (0.9)	**84.6** (0.7)	80.1 (1.2)	G → Te	78.1 (0.6)	69.9 (0.3)	**73.5** (0.6)
C → A	93.4 (0.4)	92.3 (0.3)	**92.5** (0.6)	S → T	79.5 (0.3)	74.4 (1.7)	**76.8** (0.1)
C → B	95.0 (0.6)	**94.1** (0.1)	92.1 (0.2)	S → F	77.7 (0.2)	**73.1** (0.0)	72.4 (0.5)
C → Bo	93.9 (0.8)	**91.3** (0.5)	89.0 (0.1)	S → G	83.4 (0.2)	**78.2** (0.5)	76.3 (0.9)
C → M	85.8 (0.1)	**81.5** (0.7)	79.7 (1.2)	S → Te	78.5 (0.0)	66.7 (0.2)	**74.8** (0.3)
M → A	94.2 (0.7)	89.1 (1.4)	**90.1** (0.5)	Te → T	79.8 (0.3)	71.4 (0.0)	**76.5** (0.4)
M → B	95.3 (0.5)	81.0 (16.1)	**89.9** (1.2)	Te → F	77.9 (0.1)	69.9 (0.5)	**74.3** (0.5)
M → Bo	94.1 (0.4)	80.5 (18.6)	**91.5** (0.0)	Te → G	82.5 (0.1)	75.6 (1.6)	**82.0** (0.6)
M → C	94.3 (0.5)	**90.5** (0.0)	89.7 (0.3)	Te → S	72.2 (0.0)	68.0 (0.4)	**71.3** (0.5)

Table 6: Comparison of CPT and UDAPTER by target domain classification accuracy on the Amazon Product Review and MNLI datasets. Each row represents a Source→ Target pair. On average, CPT is competitive with UDAPTER, often outperforming it. We use the T5v1.1 base model, and $(IA)^3$ as a PEFT method. The highest values between CPT and UDAPTER have been marked in bold.

Method	**Accuracy**
CPT	**83.3** (0.9)
UDAPTER	81.3 (0.6)
DANN	52.3 (1.7)
CORAL	80.9 (0.4)
Task Vectors	48.0 (0.7)
Task Vectors (fine-tuning)	69.0 (0.4)

Table 7: Comparison of CPT with more baselines, using the T5v1.1 base model and $(IA)^3$ PEFT method on the Apparel→Movies pair from the Amazon review dataset. For task vectors, we include versions with $(IA)^3$ as well as full fine-tuning. CPT outperforms all baselines.

$$h = \sigma\left(\frac{Q(l_{\text{k}} \circledcirc K^T)}{\sqrt{d_k}}\right)(l_{\text{v}} \circledcirc V)$$
$$h = (l_{\text{ff}} \circledcirc \gamma(W_1 x)W_2)$$

Here, K, Q and V are the key, query and value representations used in an attention block, and W_1 and W_2 are the weights in the feed-forward layer following an attention block. $l_{\text{k}} \in \mathbb{R}^{d_k}$, $l_{\text{v}} \in \mathbb{R}^{d_v}$, $l_{\text{ff}} \in \mathbb{R}^{d_{\text{ff}}}$, σ is the softmax function while γ is any non-linearity.

Model	**Src+Tgt**	**UDAPTER**	**CPT**
T5 v1.1 Base	85.8 (0.5)	78.6 (1.3)	**83.3** (0.5)
T5 v1.1 XL	93.0 (0.5)	65.2 (9.5)	**92.0** (1.5)
T0 3B	92.2 (0.7)	51.8 (0.8)	**93.8** (0.4)

Table 8: The performance gap between CPT and UDAPTER increases with larger models, from T5v1.1 Base (60M parameters) to T5v1.1 XL (3B parameters), and further increases with instruction tuning (T0 3B).

Intuitively, each vector l simply learns weights measuring the importance of each feature in an activation of the pre-trained model, for the specific downstream task the model is trained on.

Adapters are a popularly used and high performing PEFT framework, and He et al. (2022) have shown equivalence in the operations applied by Adapters, Prefix Tuning (Li and Liang, 2021) and LoRA (Hu et al., 2022).

Adapters work by adding small learnable modules between transformer layers. Specifically, down and up projections $W_{\text{down}} \in \mathbb{R}^{d \times r}$ and $W_{\text{up}} \in \mathbb{R}^{r \times d}$ are learnt such that $\phi =$

$\{W_{\text{up}}, W_{\text{down}}\}$. A residual connection and non-linearity γ is added at every layer,

$$h = h + \gamma(hW_{\text{down}})W_{\text{up}}$$

Table 9 shows CPT beats UDAPTER across different tuning methods. We also note that fine-tuning yields slightly better performance for all UDA methods.

F CPT in a Few-Shot Setup

Table 10 accompanies Figure 1 (Section 4), showing the 256-shot performance of CPT and other baselines, across model sizes. Similarly, Table 11 accompanies Figure 3, showing the relative performance of all baselines across varying k.

G Impact of Target Domain Exposure

The experiments in this section use the T5v1.1 base model on the Apparel→Movies domain pair of the Amazon reviews dataset.

Table 12 accompanies results from Figure 5, which show the impact of varying masking rates on CPT. Using the T5v1.1 base model, we train CPT using varying random masking rates on the Apparel → Movies domain pair, and report the mean and standard deviation over three runs. With high masking rates, the performance on the source domain is largely maintained, but the performance on the target domain rapidly deteriorates.

H Understanding how CPT aids UDA

Table 13 (accompanies Figure 7) shows the impact of masking sequences at inference, on classification accuracy. Words are selected for masking based on their their "informativeness", measured by their PMI to the inference class label. The performance of the model is best with the original unmasked sequences, indicating the presence of both informative and uninformative words are essential for strong classification performance.

Table 15 accompanies Figure 7 and shows the impact of varying masking strategies on classification performance, in a few-shot setting. We also consider two different few-shot setups: one with access to the full unlabelled datasets in phase 1 pre-training, and another where even the unlabelled data is few-shot.

To isolate any effects of PEFT methods or pre-training data, we repeat the analysis from Table 15 in Table 14 with fine-tuning Flan-T5 in a full data setting, and note similar trends.

I Single Phase CPT Training

Our proposed approach in Section 2 involves two stages of training, which is more expensive than standard single phase UDA approaches. In this section, we propose a single training phase variant to CPT, and show that it performs similarly to the original method. We use the two phase pipeline in our experiments in the main paper, but note that the single and two phase pipelines are interchangeable.

We simply replace the two phase training with a joint multi-task objective as follows,

$$\mathcal{L}(\mathcal{D}, \mathcal{D}_{\text{src}}; \theta) = \frac{1}{|D|}\frac{1}{|D_{\text{src}}|}\sum_{x' \in \mathcal{D}}\sum_{(x,y)\in\mathcal{D}_{\text{src}}} (\lambda\, l(\mathbb{C}(x,y);\theta) + (1-\lambda)\, l(\mathbb{M}(x');\theta))$$

where l is the cross-entropy loss defined in Eq. (1), and $\mathbb{M}$ and $\mathbb{C}$ are the templates defined in Section 2. λ is the adaptation factor which gradually changes from 0 to 1 over the course of training. This results in the model being trained almost exclusively on the MLM task early on in training, and the CLS task towards the end of training.

Table 16 compares the performance of the single phase and two phase variants of CPT. We also compare with a vanilla joint single phase objective, where λ is fixed at 0.5 through training (called Single Phase Vanilla). The performance of the single and two phase variants are almost identical, and either can be used interchangeably. In comparison, the vanilla single phase method is significantly weaker on the target domain.

J Instability of Domain Invariance Methods for UDA

The Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) measures the difference between first order moments of variables in a Reproducing Kernel Hilbert Space (Aronszajn, 1950). Multiple lines of work have shown that minimizing divergence measures like MMD, when combined with auxiliary task-specific loss functions, results in training instabilities and vanishing gradients (Kashyap et al., 2021; Han and Eisenstein, 2019).

We also note that as minimizing MMD does not use any label information, there is a possibility for embeddings of the target domain to be aligned with the closest source domain class cluster. For example, Figure 9 shows us a setting where both

Method	Src+Tgt	UDAPTER	CPT
Fine-Tuning	86.4 (0.4)	82.4 (1.6)	**84.4** (0.3)
(IA)3	85.8 (0.5)	81.3 (0.6)	**83.3** (0.5)
Adapters	85.3 (0.5)	79.1 (0.3)	**82.7** (0.5)

Table 9: Performance of CPT across different adaptation methods with the T5v1.1 base model on the Apparel → Movies domain pair. CPT remains more powerful than UDAPTER across all methods.

Model	Src+Tgt	UDAPTER	CPT
T5v1.1 Base	77.8 (0.4)	60.9 (1.6)	73.1 (1.7)
T5v1.1 XL	84.4 (0.1)	84.8 (1.5)	89.9 (1.1)
T0 3B	88.3 (0.5)	81.8 (1.3)	93.9 (0.4)

Model	Src+Tgt	UDAPTER	Two Phase	CPT
T5v1.1 Base	82.8 (0.6)	62.5 (0.7)	79.8 (1.4)	81.2 (0.7)
T5v1.1 XL	92.5 (0.4)	71.7 (7.8)	84.3 (0.9)	86.8 (2.2)
T0 3B	91.8 (0.6)	79.5 (6.7)	53.5 (0.4)	92.8 (0.2)

Table 10: Performance of CPT across different models, in a k-shot learning setup on the Apparel → Movies domain pair. We see CPT retaining strong performance on the target domain across models. **Left**: 32-shot. **Right**: 256-shot.

Number of Shots	Src+Tgt	UDAPTER	Two Phase UDAPTER	CPT
32	77.8 (0.4)	60.9 (1.6)	59.4 (2.0)	73.1 (1.7)
128	82.5 (0.5)	75.1 (0.6)	62.8 (0.6)	78.8 (1.0)
256	82.8 (0.6)	62.5 (0.7)	79.8 (1.4)	81.2 (0.7)

Table 11: Performance of CPT across different number of shots, on the Apparel → Movies domain pair, using the T5v1.1 base model. We see CPT retaining strong performance on the target domain across shots.

Masking Rate	Accuracy	
	Source	Target
5%	92.8 (0.8)	78.8 (1.8)
15%	**93.5** (0.4)	**83.3** (0.5)
30%	92.8 (0.6)	78.8 (1.4)
60%	92.5 (0.9)	71.0 (3.0)
90%	92.3 (0.5)	70.4 (1.5)

Table 12: Impact of Masking Rate on CPT. We train CPT using varying random masking rates on the Apparel → Movies domain pair. With high masking rates, the performance on the source domain is largely maintained, but the performance on the target domain rapidly deteriorates.

Method	Accuracy	
	Source	Target
Original	**93.5**	**83.3**
Informative Masking	88.0	78.8
Uninformative Masking	92.0	79.0

Table 13: Impact of masking at inference. We evaluate CPT on the Apparel → Movies domain pair, and select words for masking based on their "informativeness" to the classification task.

Masking Strategy	Accuracy	
	Source	Target
Random	**95.8** (0.0)	**86.8** (0.3)
Informative	93.9 (0.6)	85.3 (0.3)
Uninformative	95.0 (0.0)	84.8 (0.1)

Table 14: Impact of word selection for masking during training, using Flan-T5 base and no PEFT methods.

Phase 1 Data	Masking Strategy	Accuracy	
		Source	Target
256 Shot	Random	**91.0** (0.9)	**78.1** (2.4)
	Informative	90.4 (0.5)	76.0 (0.7)
	Uninformative	89.6 (1.2)	73.5 (1.6)
Full Data	Random	90.1 (0.5)	**81.2** (0.7)
	Informative	**91.8** (0.5)	78.0 (0.9)
	Uninformative	89.3 (0.5)	72.8 (1.1)

Table 15: Impact of word selection for masking, in a 256-shot learning setup. We evaluate CPT on the Apparel → Movies domain pair, and select words for masking based on their "informativeness" to the classification task. Random masking is most powerful for the target domain, indicating that both semantic and background features are necessary for effective classification on the unlabelled domain. However, informative masking is significantly more useful than uninformative masking.

classes of the target domain (shown in green and gray) are mapped to the cluster of negative class

Method	Accuracy	
	Source	Target
Two Phase	93.7 (0.3)	**83.3** (0.9)
Singe Phase	93.5 (0.4)	**83.3** (0.5)
Singe Phase Vanilla	93.6 (0.1)	75.0 (5.7)

Table 16: Comparison of single and two-phase variants of CPT, on the Apparel → Movies domain pair. The single and two phase variants are almost identical in performance.

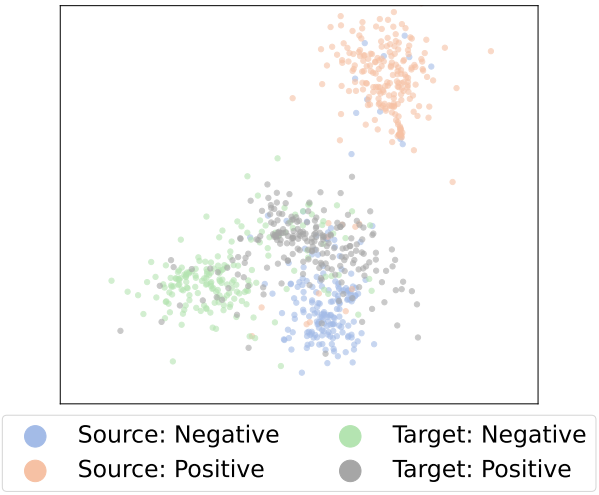

Figure 9: UMap visualizations of sentence embeddings from the Apparel → Movies data pair, using the T5v1.1 base model and $(IA)^3$ PEFT method. Training with UDAPTER risks stability issues, and all embeddings from the target domain can be mapped to the closest source class cluster. This results in poor classification performance on the target domain.

source embeddings (shown in blue).

We compare variants of the UDAPTER method in Table 17 and show that the loss is sensitive to small changes in the loss design. Specifically we compare the UDAPTER method used in the main paper with:

- MMD over Logits: Measures the MMD between the logits of source and target domains, instead of using intermediate model outputs.
- Fixed Weight MMD: Instead of the multi-task loss for the MMD reduction and classification tasks, we use fixed weights for both tasks[10].
- Two Phase MMD: The first training phase is used to minimize MMD between source and target embeddings, while the second phase is used to train the model for classification on the source domain.

CPT remains more powerful than all variants.

Method	Accuracy	
	Source	Target
CPT	93.7 (0.3)	**83.3** (0.9)
UDAPTER	**94.7** (0.3)	81.3 (0.6)
UDAPTER over Logits	95.0 (0.2)	81.3 (0.7)
Fixed Weight MMD	93.4 (0.2)	78.6 (1.3)
Two Phase UDAPTER	90.1 (0.1)	68.7 (2.0)

Table 17: Comparison of variants of minimizing MMD, on the Apparel → Movies domain pair. CPT remains more powerful than all variants.

[10]For the weighted loss, $\mathcal{L}_{\text{CLS}} + 3\ \mathcal{L}_{\text{MMD}}$ was found to be the best performing.