Toward the Explainable Soft Prompts: How does Prompt-tuning Exploit a Multilingual Pre-trained Language Model?

Anonymous ACL submission

Abstract

Since training soft prompts is a parameter-efficient way to tune a Pre-trained Language Model (PLM) on a target task, recent works suggest various training methods utilizing soft prompts. However, few studies investigate the explainability of soft prompts, which is critical to enhancing the confidence of PLM in a real-world scenario. To deal with the problems of the unexplainable soft prompts, this study explores the effects of Prompt-tuning v1 (Lester et al., 2021) and Prompt-tuning v2 (Liu et al., 2022) on PLM. More precisely, we conducted the experiments using a multilingual GPT to generalize our observations not only on tasks but also on languages. We first confirmed whether soft prompts are gathered according to tasks or languages, and then analyzed how soft prompts utilize PLM in terms of the two main architectures of GPT: the attention mechanism and the activated neurons. As a result, we conclude that soft prompts are trained while employing the knowledge PLM obtained during pre-training to solve the target task, which is consistent with language. Our findings reveal that deep soft prompts are explainable when they can employ varying knowledge from every layer.

1 Introduction

In Natural Language Processing (NLP), prompt-based learning has emerged as a promising paradigm to employ a Pre-trained Language Model (PLM) effectively. Particularly, **Prompt-tuning v1** (Lester et al., 2021) shows that training only **soft prompts**, which are vectors of real numbers prepended to the embedding layer of PLM, can achieve comparable performances to traditional fine-tuning. Additionally, soft prompts are fed to all layers of PLM in **Prompt-tuning v2** (Liu et al., 2022), where **deep soft prompts** encode more task-specific weights and affect the model directly.

Despite the great promise of the *pre-train and prompt-tune* paradigm, it is problematic that the difficulty of faithfully interpreting soft prompts in natural language could potentially lead to concealed adversarial attacks (Khashabi et al., 2022). Ultimately, it makes PLM unexplainable in that we cannot assure how soft prompts operate in PLM. However, few studies have tried to reveal the details of the relationship between Prompt-tuning and PLM, which are essential to the explainable soft prompts.

Meanwhile, several studies have shown that soft prompts are one of the parameter-efficient training methods in a cross-lingual setting (Zhao and Schütze, 2021; Vu et al., 2022), which means that soft prompts encode the knowledge about a target task in the multilingual space. Since the generalizability of tasks and languages can improve the explainability of PLM, it is important to capture the features shared in each task and language. Accordingly, in this study, we investigate how soft prompts save and utilize the information of a multilingual PLM, mGPT (Shliazhko et al., 2022), focusing on English and Korean.

This study also aims to figure out the effects of Prompt-tuning v1 and Prompt-tuning v2 on the multilingual PLM, so that we provide fundamental directions to enhance the interpretability and explainability of PLM and Prompt-tuning. To analyze the effects of soft prompts, we focus on the changes after Prompt-tuning in terms of two major structures of GPT: the attention mechanism and the activated neurons.

To this end, we address the following research questions:

1. Can we distinguish soft prompts according to the encoded information about target tasks or target languages?

2. Can we find any explainable patterns in the changes in the attention mechanism after Prompt-tuning?

3. Can we observe any explainable patterns in the activated neurons of soft prompts through layers?

We first present the visualizations of soft prompts to show that deep soft prompts are gathered according to target tasks in the multilingual setting. Second, we find that the attention distribution changes in some layers more than in other layers regardless of tasks and languages. Also, in Prompt-tuning v2, the changes are explainable because the most changed attention layers are composed of content-dependent heads rather than position-based heads. These results suggest that soft prompts utilize the knowledge encoded in the attention mechanism to solve the target task. Third, we observe a special phenomenon, where the activated neurons show task-common behavior rather than task-specific behavior in the second to last layer. We conclude that this may be due to the characteristics of the PLM, like isotropy. Finally, the ablation study supports these findings as where deep soft prompts fed into the actively changed layers show higher performances than the ones fed into the non-actively changed layers. To the best of our knowledge, this study is the first to probe the changes in the PLM after Prompt-tuning v1 and Prompt-tuning v2.

2 Related Works

Soft prompts. Li and Liang (2021) propose Prefix-tuning for lightweight model training. They prepend the continuous task-specific vectors to the input so that their own parameters are only trained. As the continuous vectors do not limit their space on the embedding of PLM, they are more expressive and can affect all layers in PLM. Using GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2020), they show that the performances on Natural Language Generation (NLG) tasks are comparable to the performances of fine-tuning both in fully- and semi-supervised settings.

Liu et al. (2021) suggests P-tuning which is more flexible to task types and LM types than Prefix-tuning. They use a discrete template that include soft prompts and train both PLM and those soft prompts. They report that performances improve both in GPT and BERT via P-tuning.

Similarly, Lester et al. (2021) propose Prompt-tuning v1, where they use only soft prompts by freezing the parameters of PLM. They show that the performances of Prompt-tuning v1 are comparable to the performances of fine-tuning, where the capacity of PLM has a key role. Thus, they conclude that Prompt-tuning is a parameter-efficient way to employ the knowledge LM obtained during pre-training.

Furthermore, Liu et al. (2022) introduce Prompt-tuning v2, where they implement Prompt-tuning more deeply by injecting soft prompts into every layer of PLM. While Prompt-tuning v1 has low performances on the hard sequence labeling tasks, Prompt-tuning v2 improves the performances of various tasks, namely extracting question answering and named entity recognition.

The explainable soft prompts. Again, few studies endeavor to enhance the explainability of soft prompts. Lester et al. (2021) try to interpret soft prompts by measuring the similarities between the embeddings of learned soft prompts and the vocabulary of PLM. They observe that soft prompts have 'word-like' representations, which are relevant to the domain of the target task. An example of which is the BoolQ dataset (Clark et al., 2019) of the nature/science category, where the soft prompts are close to the words such as '*science*', '*technology*' and '*engineering*' in the embedding space.

To interpret soft prompts in human language, Khashabi et al. (2022) investigate the Prompt Waywardness hypothesis. In this hypothesis, there exists a soft prompt that can solve the target task while becoming close to the arbitrary discrete prompt, which is not relevant to the target task. They observe that soft prompts satisfying the Prompt Waywardness hypothesis do exist. They also provide some explanations. First, soft prompts cannot be projected to exactly one embedding of discrete prompts. Second, when soft prompts are injected only into the first layer, the deeper layers have more expressivity, where the effects of

Task	English	Korean
STS	GLUE-STS	KLUE-STS
NLI	SNLI	KorNLI
SA	SST2	NSMC
TC	AGNews	KLUE-Ynat
QA	SQuAD 2.0	KorQuAD 1.0
CG	CommonGen	KommonGen

Table 1: The datasets used in this study.

Waywardness get stronger. Finally, they discuss that it is hard to discover the human-interpretable soft prompts, which leads to the side effects in the real-world scenario, such as the concealed adversarial attacks.

Besides the interpretability of soft prompts, the changes that soft prompts cause to PLM are also crucial to enhance the explainability of soft prompts. Also, to deal with the second reason Khashabi et al. (2022) mentioned, we need to investigate Prompt-tuning v2 where all layers are controlled by soft prompts. Thus, our study focuses on the effects of Prompt-tuning v1 and Prompt-tuning v2 on PLM through layers, so that we can figure out the operation of soft prompts in an explainable way.

3 Methods

3.1 Training soft prompts

We conduct the experiments using mGPT, which has the same architecture with GPT-3 and 1.3B parameters. Since GPT is an autoregressive model, we make the model generate the label words after the separator token </s> until the end token <|*endoftext*|> and train soft prompts by computing the conditional probability for the label words. The label words for each task are in Table 3 in Appendix 0. Via this method, we can train soft prompts in the same way for all tasks, including classification and generation tasks. Also, we freeze mGPT and update only the parameters of the prepended soft prompts.

For a single-input type, we feed the input $\{p_1, \dots, p_k, < s >, x_1, \dots, x_n, </s >, w_{gold}, < |endoftext| >\}$ to the model, where $P = \{p_1, \dots, p_k\}$ is soft prompts with the length k, $X = \{x_1, \dots, x_n\}$ is the input sentence with the length n, and w_{gold} is the label words. Then, following Lester et al. (2021), we train soft prompts by maximizing the probability:

$$Pr_{\theta;\theta_P}(w_{gold}, < |endoftext| > | p_1, \dots, p_k; < s >, x_1, \dots, x_n, </s >) \quad (1)$$

where θ is the parameters of the model and θ_P is the parameters of soft prompts.

Similarly, for a pair-input type, the model maximizes the probability to generate the label words after the second separator token:

$$Pr_{\theta;\theta_P}(w_{gold}, < |endoftext| > | p_1, \dots, p_k; < s >, x_1^1, \dots, x_n^1, </s >, x_1^2, \dots, x_m^2, </s >) \quad (2)$$

For both Prompt-tuning v1 and Prompt-tuning v2, we randomly initialize soft prompts ranging from -0.5 to 0.5. Also, we use the prompt length $k = 20$. Additionally, we slightly modify the injection of deep soft prompts in Prompt-tuning v2. Unlike Liu et al., 2022, we inject *key* and *value* as the same one, so that we get only one prompt embedding per layer, which has the exact same dimension with the model. Otherwise, soft prompts are composed of two separate embeddings.

Using principal component analysis (PCA), we visualize soft prompts of each target task to see how soft prompts are clustered. In the case of deep soft prompts, we analyze soft prompts for each layer.

3.2 The Attention Variability and the Changes in the Attention Mechanism

In GPT, the attention mechanism works in the left-to-right direction, so most tokens give the maximum attention to their previous token. Vig and Belinkov (2019) consider that such a tendency of the attention distribution in GPT is based on not the content but the position. To measure how attention varies over different input sequences, they suggest attention variability, which adopts the basics of the mean absolute deviation:

$$Variability_{\alpha} = \frac{\sum_{x \in X} \sum_{i=1}^{|x|} \sum_{j=1}^{i} |\alpha_{i,j}(x) - \bar{\alpha}_{i,j}|}{2 \cdot \sum_{x \in X} \sum_{i=1}^{|x|} \sum_{j=1}^{i} \alpha_{i,j}(x)} \quad (3)$$

where $\alpha_{i,j}(x)$ is the attention score x_i gives to x_j, and $\bar{\alpha}_{i,j}$ is the mean of $\alpha_{i,j}(x)$ overall $x \in X$.

Meanwhile, the first token of each input sequence tends to receive the maximum attention score. Thus, the first token is excluded to calculate the variability. We also compute the variability with

the first N tokens ($N = 10$), excluding the first token.[1]

Since we do not update the parameters of mGPT, we believe that the changes in the attention mechanism after Prompt-tuning show the effects of Prompt-tuning on mGPT. Thus, we investigate the difference of the attention distribution between the pre-trained mGPT and the prompt-tuned mGPT. We use Kullback-Leibler divergence which is commonly used to measure the difference between two probability distributions:

$$KLdiv(P(X), M(Y)) = \sum_{x \in X, y \in Y} x \log \frac{x}{y} \quad (4)$$

where P(·) is a prompt-tuned model and M(·) is a pre-trained PLM.

To get the KL divergence per head, we post-process the attention distributions. We first sum the attention scores for each token, and then remove the attention score of the special tokens <*s*> and </*s*> in each input. In the case of P(·), we exclude the attention scores of the tokens of soft prompts before post-processing. Finally, we replace x and y with 1e-20 when they are 0 after passing the softmax to avoid the infinity when computing KL divergence.

Lastly, we get the correlation between the attention variability and the attention changes through layers to see whether we can interpret the patterns of the changes in terms of the knowledge mGPT learned during pre-training.

3.3 The Activated Neurons

Geva et al. (2021) suggest that the FFN layers in pre-trained Transformer mirror the key-value memories, since the neural memory and the FFN layer have a similar structure (Equation 5 & Equation 6). The difference between them is the function applied; FFN has a non-linear activation function, and the neural memory has the softmax function.

$$FFN(x) = f(x \cdot K^T) \cdot V \quad (5)$$

$$MN = softmax(x \cdot k^T) \cdot V \quad (6)$$

[1] For commonsense generation task, we use first 2 tokens without the first token because all input sequences include only 3 words.

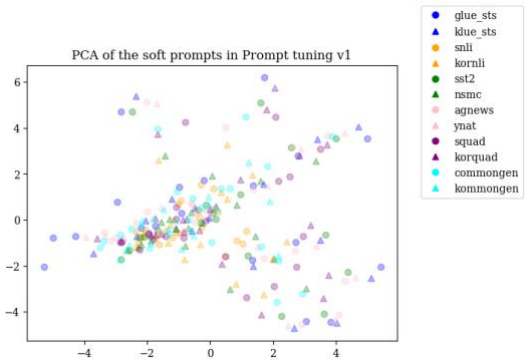

Figure 1: The PCA result of soft prompts learned for the target task in Prompt-tuning v1.

They observe that keys in the FFN layers, which are the activated neurons, have positive correlations with the human-interpretable input patterns such as shallow and semantic patterns.

Motivated by the activated neurons, Su et al. (2022) propose ON score, where we can measure the response of the prompt-tuned model. Following them, this study computes ON score using the decoding token </*s*> per layer to investigate how task-specific the prompts are. Given an input sequence $\{P, </s>\}$, we get the output of the activation state $AS(P)$ of soft prompts P.

$$ON(P^{t_1}, P^{t_2}) = \frac{AS(P^{t_1}) \cdot AS(P^{t_2})}{||AS(P^{t_1})||\ ||AS(P^{t_2})||} \quad (7)$$

Recently, Wang et al. (2022) introduce skill neurons that can be detected via Prompt-tuning. They report that similar neurons are activated in similar tasks in the upper layers, which means that the neurons encode the task-specific skills. This study also explores the response of the model through layers to discuss which skills each layer has.

4 Experiments

4.1 Tasks and Performance Results

We conduct experiments on 6 tasks including classification, extraction, and generation.

STS. Semantic Textual Similarity is a task to measure the similarity score between two sentences from 1 to 5. The higher the score is, the higher the similarity is. In this study, we binarize

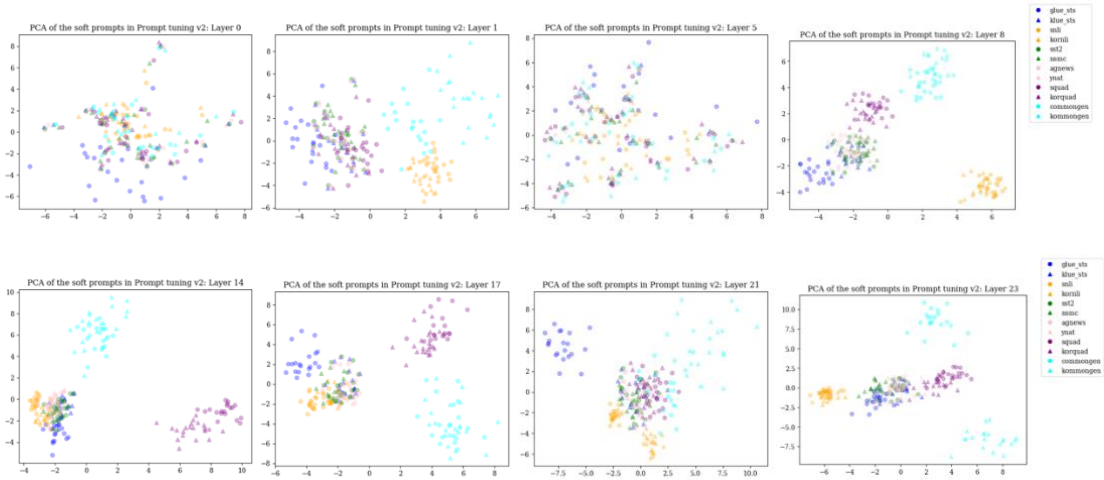

Figure 2: The PCA result of soft prompts learned for the target task in Prompt-tuning v2.

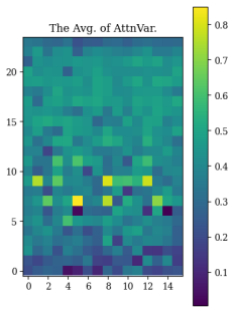

Figure 3: The average of the attention variability on all tasks. We present the results for each task in Figure 12 in Appendix C.

the score into similar and dissimilar class. If the score is more than 3, it is labeled as similar, and if the score is less than 3, it is labeled as dissimilar.

NLI. Natural Language Inference is a task to decide the semantic relationship of two sentences, a premise and a hypothesis, among three categories: entailment, contradiction, neutral. When given a premise first, the model should determine if a hypothesis is true or false or undetermined.

SA. Sentiment Analysis is a task to determine the sentiment expressed in a sentence. Usually, the sentence is classified into two labels: positive and negative. In this study, we use the movie review dataset, which is one of the popular domains.

TC. Topic Classification task includes sentences or paragraphs from different themes. We use news topic classification task, where the topic such as *IT/science* and *Social* is annotated for each headline. Also, the topics vary for each language.

QA. For Question and Answering for reading comprehension, the model should extract the answer to a question from the given context. The answer can be a word or spans consisting of more than two words.

CG. Commonsense Generation is a task proposed to assess the model's ability to generative commonsense reasoning. Using a set of three common concepts including an object (noun) and action (verb), the model generates a full grammatical sentence, which coheres with an everyday scenario.

The datasets in each language are presented in Table 1. Also, we report the performance results in Table 7 in Appendix C. Every task gets improved in Prompt-tuning v2. Particularly, the scores of KLUE-STS and KorNLI rise by around 35 and 17, respectively.

4.2 Deep soft prompts encode task-relevant information clustering by language.

We present the visualizations of soft prompts using PCA to investigate how they encode the knowledge according to task and language. While we cannot find any meaningful clusters in Prompt-tuning v1 (Figure 1), soft prompts are grouped in Prompt-tuning v2 through layers (Figure 2).

After passing the first layer (layer 0), soft prompts tend to be gathered according to the target task. Although soft prompts are dispersed again in layer 5, they keep forming the clusters according to the target task. The task clusters in different languages are grouped, especially those with high cohesion in the middle layers (7~17). Additionally, starting from layer 14, we observe the separation of

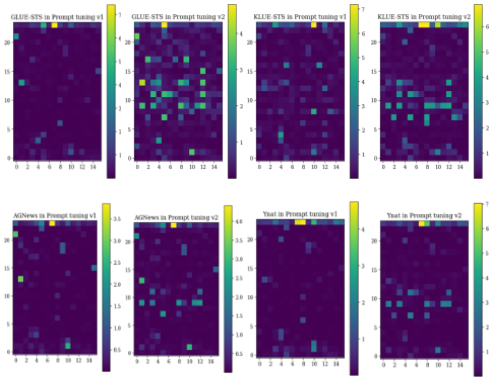

Figure 4: The KL divergence results on STS and TC. We present the results of all tasks in Figure 13 and Figure 14 in Appendix C.

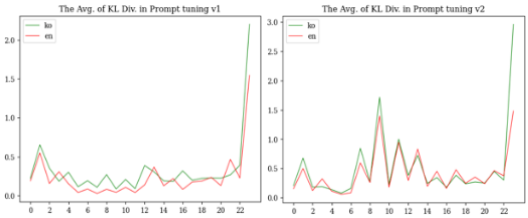

Figure 5: The average of KL divergence per layer.

some clusters (NLI, STS, TC) that are not well separated in the previous layers.

These results suggest that deep soft prompts learn task-relevant knowledge in the multilingual space since soft prompts of the same task in English and Korean are gathered. Also, the different patterns through layers imply that the knowledge employed to solve the target task varies over layers. Deep soft prompts controlling across layers might lead to low expressivity in the deeper layers, which means that it can contribute to the interpretability of soft prompts. We leave this implication as an open question for future works.

4.3 Deep soft prompts employ the context-dependent attention heads.

We find that the lower layers have lower variability (Figure 3), which is consistent with the observations in the prior work (Vig and Belinkov, 2019). Also, the attention heads in layers 7~9 have high attention variability regardless of tasks and languages. These results show that the attention heads of mGPT have encoded context information in a robust way during pre-training.

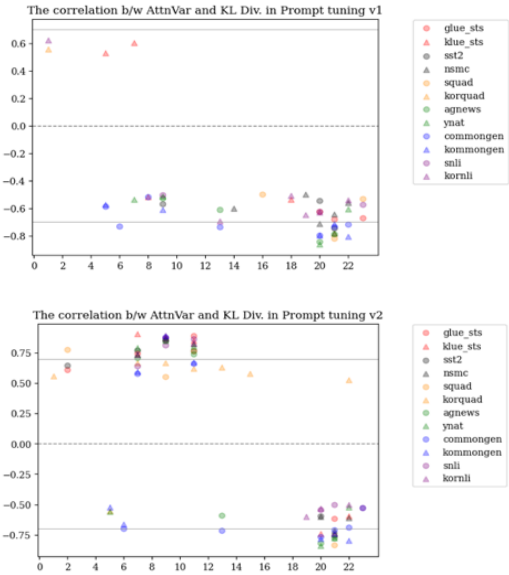

Figure 6: The correlation between the attention variability and the KL divergence.

Figure 4 displays the KL divergence results in Prompt-tuning v1 and Prompt-tuning v2 on STS and TC tasks. In Prompt-tuning v1, we observe that the attention distribution of the upper layers changes a lot, followed by the initial layers. In Prompt-tuning v2, the layers between 6 and 13 change significantly as well, which consist of the content-dependent heads. Additionally, the results are consistent with the same task rather than with the same language.

To analyze the changes by layer, we plot the average KL divergence for each layer. Figure 5 shows that the middle layers from 7 to 17 change remarkably in Prompt-tuning v2 with the peaks at the odd layers {7, 9, 11, 13, 15, 17}. On the other hand, in Prompt-tuning v1, the changes in the middle layers are relatively minor. Also, the peaks appear at different layers for each language; the even layers {6, 8, 10} for both languages, the even layers {12, 16} for Korean, and the odd layers {13, 15} for English. This is because they do not have a direct impact on the deeper layers in PLM (Liu et al, 2022) since soft prompts are injected only in the embedding layer in Prompt-tuning v1.

Next, we present the results of the Pearson correlation (p-value < 0.05) between the attention variability and the results of KL divergence in each task (Figure 6). Grouping the layers into 4 groups according to the depth, we sum up the observations in the attention mechanism.

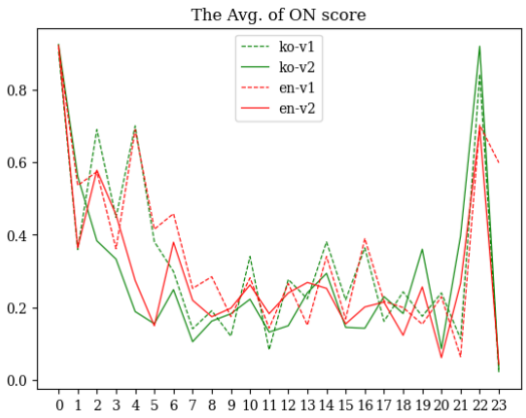

Figure 7: The average of ON score.

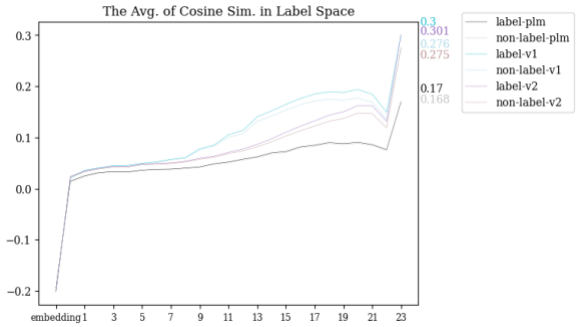

Figure 8: The cosine similarity of the decoding token between the label words and between the non-label words. If the label word is '*positive*', the non-label word is '*negative*'. Also, if there are multiple non-label words, we randomly select one non-label word. To calculate the similarity, we use tasks without extraction and generation.

First, the lower layers (layers 0~5), where the position-based heads are concentrated, show a small change both in Prompt-tuning v1 and Prompt-tuning v2. Second, the middle-lower layers (layers 6~11) have different patterns in Prompt-tuning v1 and Prompt-tuning v2. Considering that the content-dependent heads are gathered in layers 7~9 (Figure 3), the positive correlations in Prompt-tuning v2 suggest that deep soft prompts are trained while employing the context information encoded in mGPT. Third, the middle-upper layers (layers 12~17) do not show significant changes, yet, we observe that the changes in the pair-input type tasks in these layers are comparable to the changes in the middle-lower layers (See Figure 15 in Appendix C). Lastly, the upper layers (layers 18~23) have the same pattern in both Prompt-tuning v1 and Prompt-tuning v2, where the last layer shows a significant change. Simultaneously, they have negative correlations, which implies that the additional elements are involved in activating the attention head to predict the label in these layers.

4.4 The activated neurons of soft prompts are task-specific in the deeper layers, where other features of PLM make the neurons common as well.

The results of the average ON score between all combinations of tasks in each language are reported in Figure 7, where the lower the score is, the more task-specific the layer is. We find that the scores in the first layer are high and the scores reduce rapidly in the second layer, which means that the first layer has encoded the common information regardless of task and language.

Meanwhile, the results in layers from 2 to 5 have some fluctuations in Prompt-tuning v1, while there are few flu ㄴ ctuations in Prompt-tuning v2. The middle layers (layers 6~17) also have fluctuations, but with more narrow gaps between layers. Notably, the scores become the lowest in layer 21 in Prompt-tuning v1 and in layer 20 in Prompt-tuning v2.

While these results suggest that the deeper layers have task-specific neurons, the second to last layer shows peaks with high scores near the scores in the first layer, which has not been observed in previous works. Since most tasks have highly similar neurons in the second to last layer, we hypothesize that other features of PLM affect the neurons when solving the target tasks.

To test our intuitions, we present the cosine similarity in the label space. Figure 8 illustrates that the cosine similarities of the decoding token *</s>* with the gold label words get higher than ones with the non-gold label words in the deeper layers. Indeed, the second to last layer has the trough, where the gaps between two similarities narrow down. We believe that the anisotropic embedding space of mGPT could lead the task-common neurons in the second to last layer because the desired label words are actually similar in the space. Thus, we conclude that these observations are one of the possible explanations as to why soft prompts are hard to interpret.

	English		Korean	
Task	Peak	Trough	Peak	Trough
STS	**84.17**	83.74	60.32	**63.26**
NLI	**85.35**	84.85	**61.97**	58.18
SA	**90.02**	89.79	**86.73**	86.25
TC	**86.66**	86.46	**83.75**	82.66
QA	**64.40/ 49.29**	61.82/ 47.11	**68.82/ 62.41**	66.57/ 60.42
CG	**82.56**	81.62	**90.43**	90.21

Table 2: The performance results of the compressed Prompt-tuning v2.

5 Ablation study

For the ablation study, we try to investigate whether deep soft prompts can better utilize the specific layers, where effects are notable.

To this end, we feed soft prompts to the layers, where changes and roles are clear in order to confirm the effects of Prompt-tuning v2 on the model. In Section 4.3, we find that the changes in the attention distribution show a zigzag trend line. The peaks are detected in layers {1, 7, 9, 11, 13, 15, 17, 21} and the troughs are detected in layers {2, 8, 10, 12, 14, 16, 18, 20, 22}, both in English and Korean. We hypothesize that soft prompts are more helpful for the layers at the peaks than the layers at the troughs.

Furthermore, looking at the results of the activated neurons in Section 4.4, the last layer (23) is a task-specific layer, and the second-last layer shows a special phenomenon, where the neurons of all tasks are similar to each other. Also, the neurons in layer 20 have the lowest ON score, which means that the neurons are activated depending on tasks.

Motivated by these observations, we group the layers into two categories: peak and trough. The peak group includes layers {0, 1, 7, 9, 11, 13, 15, 17, 20, 21, 22, 23}, and the trough group includes layers {0, 2, 8, 10, 12, 14, 16, 18, 20, 21, 22, 23}. We set the first layer {0} and the last four layers {20, 21, 22, 23} in common, in consideration that the former is the input layer, and the latter ones are task-specific layers. Each group includes half of the number of layers of mGPT, which means that they have half parameters of soft prompts in Prompt-tuning v2.

This method is similar to Liu et al. (2022). They conduct the ablation study of Prompt-tuning v2 by adding soft prompts to certain layers grouping into two groups, the first four layers and the last four layers. On the contrary, this study tries to classify the layers in an explainable way. We believe that the alternating dense and sparse attention mechanism of mGPT prevent the performances from the significant drop when excluding some layers.

Table 2 shows the performances of each group on all tasks. Compared to the results of vanilla Prompt-tuning v2, most scores drop but raise in comparison to the results of Prompt-tuning v1. Notably, the scores of both groups on SST2 raise compared to Prompt-tuning v2.

Except for KLUE-STS, the peak groups show higher performances than the trough groups. Thus, we conclude that soft prompts in the peak groups employ the knowledge of mGPT better.

6 Conclusion

In this study, we investigate how soft prompts and deep soft prompts encode task-relevant knowledge and employ the knowledge from the PLM. Using mGPT, we conduct the experiments on various tasks, including classification and generation in each language. First, we find that deep soft prompts obtain task-specific knowledge in the deeper layers, while soft prompts in Prompt-tuning v1 do not. Also, deep soft prompts have shared task information within languages. Second, we observe that the changes in the attention mechanism after Prompt-tuning v2 can be explained in terms of the attention variability. The higher the attention variability is where the more significant the changes are. Simultaneously, the last four layers show negative correlations between the attention variability and the changes in the attention distribution after Prompt-tuning v1 and Prompt-tuning v2. Thus, we conclude that these layers play another key role to solve the target task. Third, the response of the model shows that the deeper layers have more task-specific neurons. Unlike previous studies, we report a special phenomenon in the second to last layer, where most of all tasks have unexpected common neurons.

Finally, we confirm the observations of the attention mechanism and the activated neurons in the ablation study. We hope that this study provides a guide to the explainable soft prompts and the explainable PLM.

Limitations

Although we explore the multilingual space of PLM, our findings are limited to English and Korean. We choose Korean as a non-English language because Korean is understudied in Prompt-tuning and has different linguistic properties from English, such as typology. Also, even though our experiments have a fixed seed (42), other trials with other seed numbers are required, since different seed numbers can lead to fluctuating results. Additionally, we fail to analyze the observations relating to the performances. We encourage further studies to include more various languages and more systemic experiments.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer Feed-Forward Layers Are Key-Value Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiyeon Ham, Yo Joong Choe, Kyubyong Park, Ilji Choi, and Hyungjoon Soh. 2020. KorNLI and KorSTS: New Benchmark Datasets for Korean Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 422–430, Online. Association for Computational Linguistics.

Daniel Khashabi, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, and Yejin Choi. 2022. Prompt Waywardness: The Curious Case of Discretized Interpretation of Continuous Prompts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3631–3643, Seattle, United States. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt-tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. Korquad1.0: Korean qa dataset for machine reading comprehension. arXiv preprint arXiv:1909.07005.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A Constrained Text Generation Challenge for Generative Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. arXiv:2103.10385.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-Tuning: Prompt-tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Taehwan Oh, et al.

2021. Klue: Korean language understanding evaluation. arXiv preprint arXiv:2105.09680.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Jaehyung Seo, Chanjun Park, Hyeonseok Moon, Sugyeong Eo, Myunghoon Kang, Seounghoon Lee, and Heuiseok Lim. 2021. KommonGen: A Dataset for Korean Generative Commonsense Reasoning Evaluation. In *Proceedings of the 33th Annual Conference on Human & Cognitive Language Technology 2021*, October, pages 55-60.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. arXiv preprint arXiv:2204.07580.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. 2022. On Transferability of Prompt-tuning for Natural Language Processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3949–3969, Seattle, United States. Association for Computational Linguistics.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the Structure of Attention in a Transformer Language Model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. 2022. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. arXiv preprint arXiv:2205.12647.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022. Finding Skill Neurons in Pre-trained Transformer-based Language Models. arXiv preprint at arXiv:2211.07349.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Annual Conference on Neural Information Processing Systems 2015*, December 7-12, 2015, Montreal, Quebec, Canada, pages 649–657.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and Soft Prompting for Multilingual Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8547–8555, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Appendices

A Dataset Details

Table 3 shows the details of datasets including sources. Also, Table 4 and Table 5 show the verbalizers and the number of labels in each dataset. We designed the verbalizers to be semantically close to the actual label words.

B Experimental Details

We used two A100 GPUs with 80G memory. Also, we used the seed number 42. The number of trainable parameters for Prompt-tuning v1 is 40960 and the one for Prompt-tuning v2 is 983040. The hyperparameters for each task is presented in Table 6. The performance results are presented in Table 7. For task STS, NLI, SA, and TC, we use sklearn package (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html) for scoring. For QA, we use the codes from the official website (SQuAD: https://rajpurkar.github.io/SQuAD-explorer/ ; KorQuAD: https://korquad.github.io/KorQuad%201.0/). Also, For CG, we use the codes from the official github (CommonGen: https://github.com/INK-USC/CommonGen; KommonGen: https://github.com/nlpai-lab/KommonGen).

C Experimental Results

Figure 9, Figure 10, and Figure 11 illustrate the PCA results in Prompt-tuning v2 over all layers. Also, Figure 12 shows the attention variability for all tasks. Figure 13 and Figure 14 show the results of KL divergence for all tasks. Lastly, Figure 15 presents the average KL divergence per input type.

Type	Task	English	Source	Korean	Source
Pair CLS	STS	GLUE-STS	Wang et al., 2019	KLUE-STS	Park et al., 2021
	NLI	SNLI	Ham et al., 2020	KorNLI	Ham et al., 2020
Single CLS	SA	SST2	Socher et al., 2013	NSMC	https://github.com/e9t/nsmc
	TC	AGNews	Zhang et al., 2015	KLUE-Ynat	Park et al., 2021
Extraction	QA	SQuAD 2.0	Rajpurkar et al., 2018	KorQuAD 1.0	Lim et al., 2019
Generation	CG	CommonGen	Lin et al., 2020	KommonGen	Seo et al., 2021

Table 3: The details about datasets used in this study.

Task	English Verbalizer (train/validation/test)
STS	similar (2994/629), dissimilar (2755/871)
NLI	entailment (183414/3329/3368), contradiction (183185/3278/3237), neutral (182762/3235/3219)
SA	positive (37569/444), negative (29780/428)
TC	business, scitech, sports, world (30000/1900)

Table 4: The verbalizers (label words) and the number of each example for English classification datasets. If test dataset is not provided, we use validation dataset instead.

Task	Korean Verbalizer (train/validation/test)
STS	유사(similar) (5602/220), 상이(dissimilar) (6066/299)
NLI	함의(entailment) (183382/523/1670), 모순(contradiction) (183382/524/1670), 중립(neutral) (183382/523/1669)
SA	긍정(positive) (74825/25171), 부정(negative) (75710/24826)
TC	정치(polticis) (7379/722), 경제(economics) (6118/1348), 생활문화(livingculture) (5751/1369), 사회(society) (5133/3701), IT 과학(ITscience) (5235/554), 세계(world) (8320/835), 스포츠(sports) (7742/578)

Table 5: The verbalizers (label words) and the number of each example for Korean classification datasets. If test dataset is not provided, we use validation dataset instead.

Task	Batch size	Epochs	Max length
STS	16	10	180
NLI	32	16	150
SA	32	20	256
TC	64/32	10	80
QA	16	12	400
CG	16	20	60

Table 6: The hyperparameters of each task. For batch size in task TC, 64 is for English dataset and 32 is for Korean dataset.

Scoring Method	English	Score		Korean	Score	
		V1	V2		V1	V2
Accuracy	GLUE-STS	83.2	84.23	KLUE-STS	38.43	71.59
Macro-F1	SNLI	80.95	86.46	KorNLI	45.67	62.8
Accuracy	SST2	87.15	88.18	NSMC	84.94	87.18
Macro-F1	AGNews	85.8	87.27	KLUE-Ynat	81.18	84.27
F1/EM	SQuAD 2.0	61.98/ 47.25	67.21/ 51.91	KorQuAD 1.0	65.41/ 59.62	72.32/ 66.50
Coverage	CommonGen	78.4	82.97	KommonGen	87.94	91.33

Table 7: The performances of all tasks in Prompt-tuning v1 and Prompt-tuning v2.

Figure 9: The PCA result of soft prompts learned for the target task in Prompt-tuning v2 in layers 0~7.

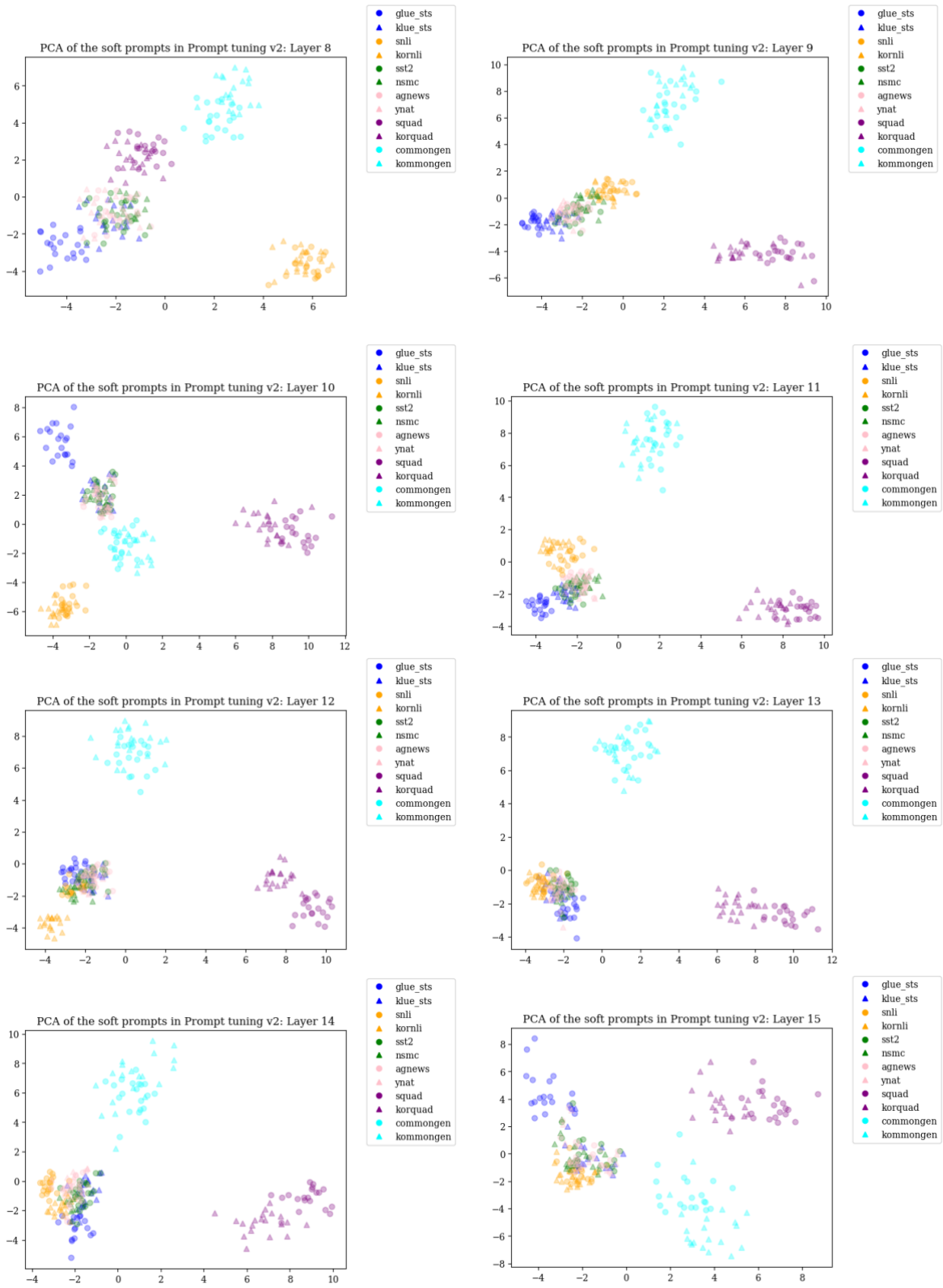

Figure 10: The PCA result of soft prompts learned for the target task in Prompt-tuning v2 in layers 8~15.

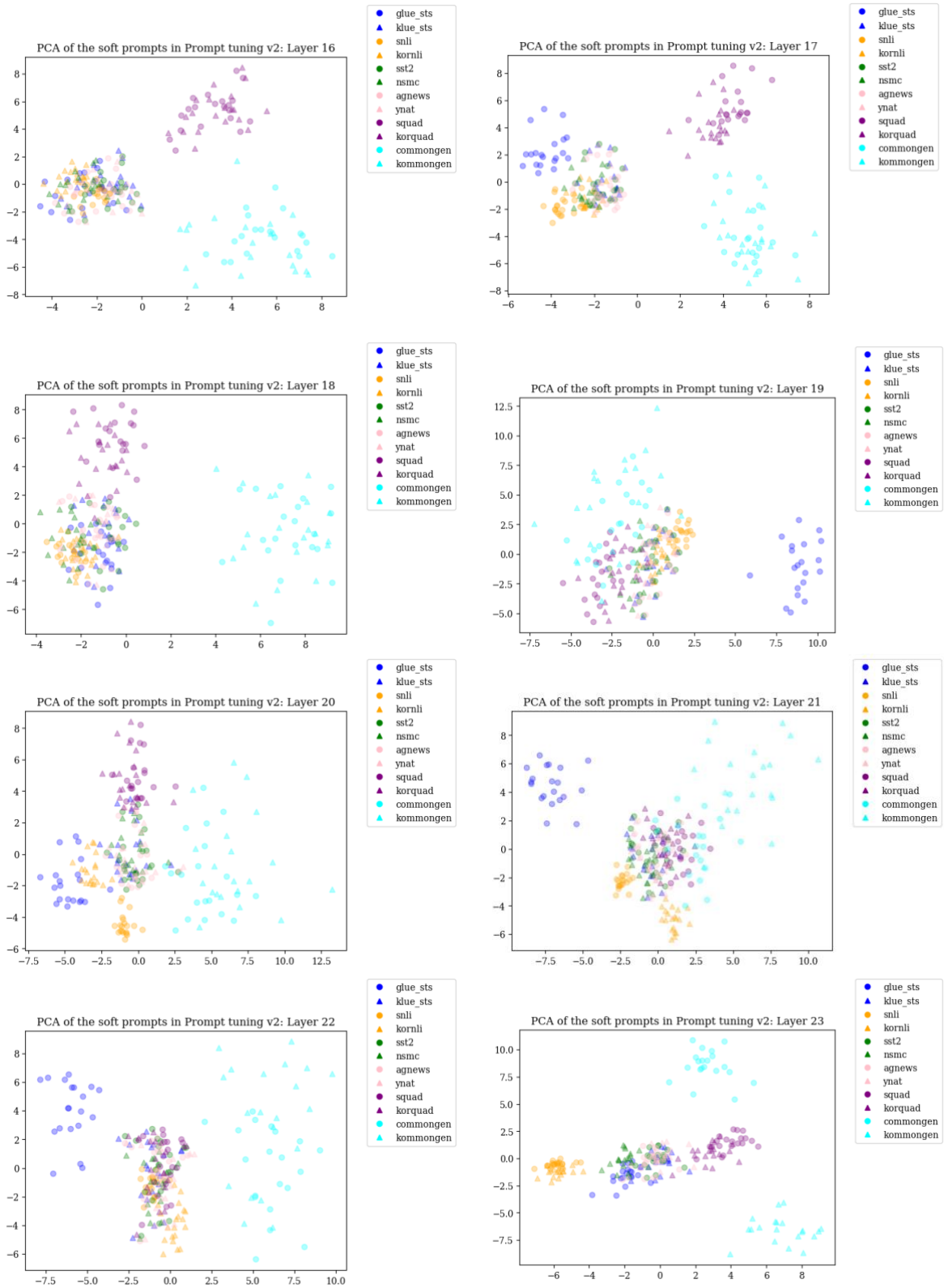

Figure 11: The PCA result of soft prompts learned for the target task in Prompt-tuning v2 in layers 16~23.

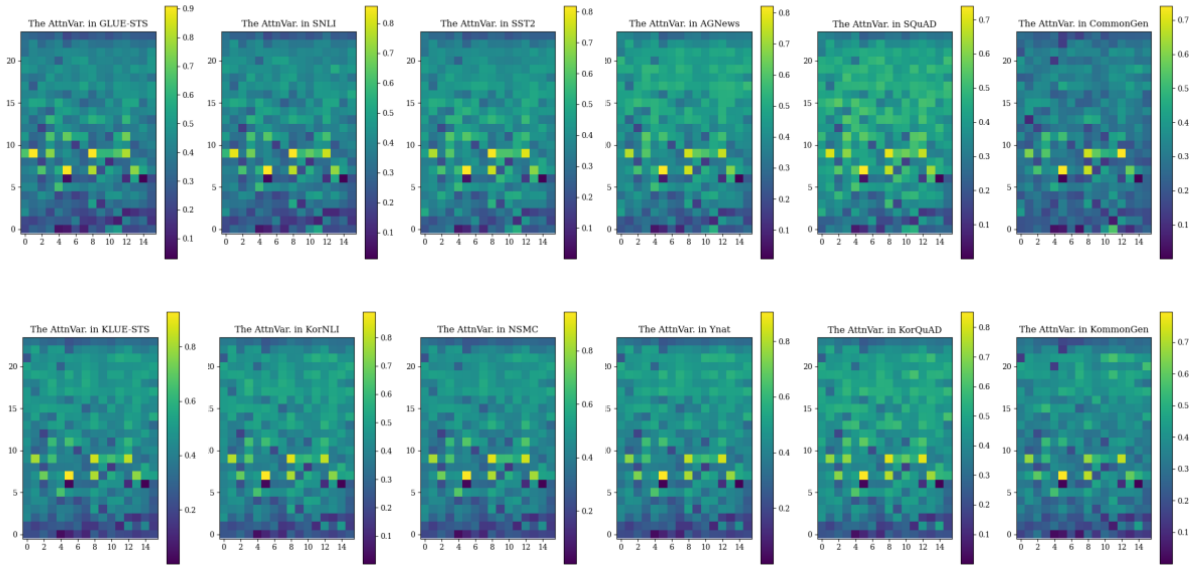

Figure 12: The attention variability of each task.

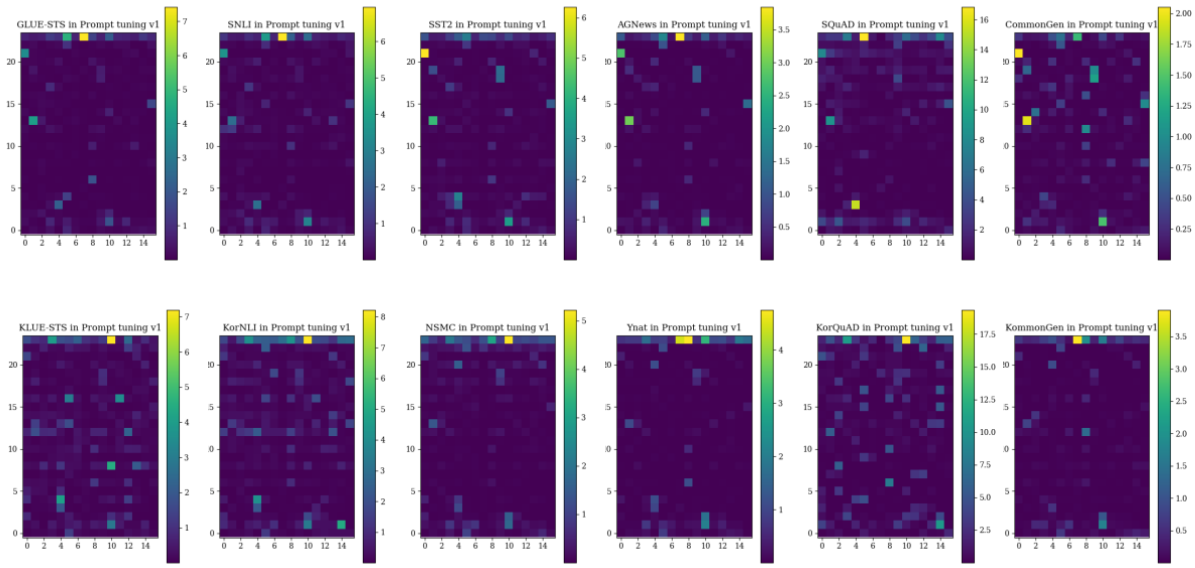

Figure 13: The KL divergence of each task in Prompt-tuning v1.

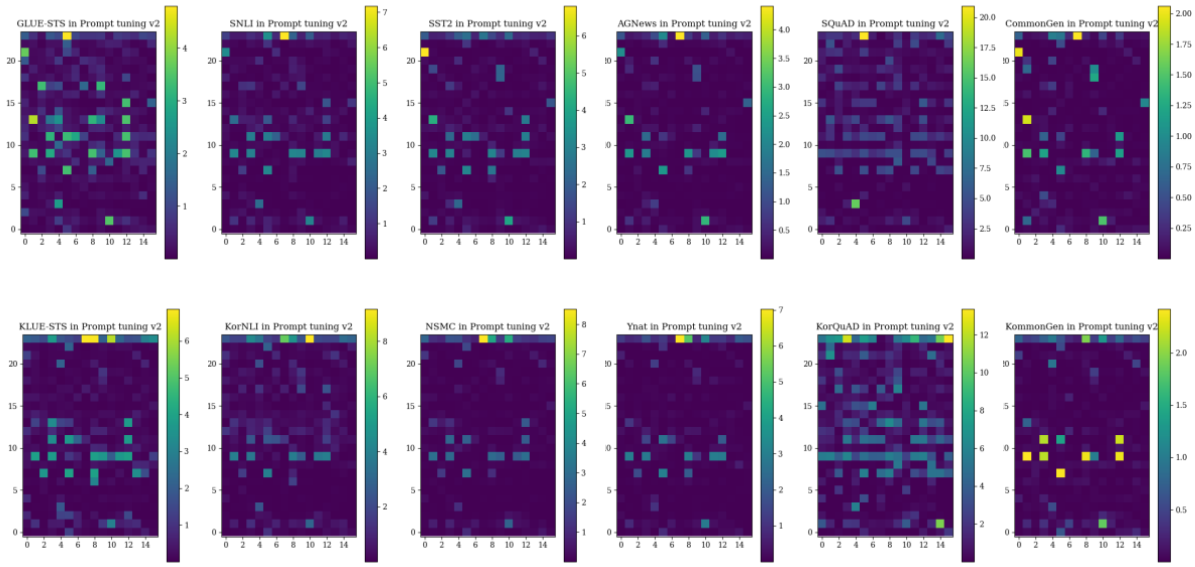

Figure 14: The KL divergence of each task in Prompt-tuning v2.

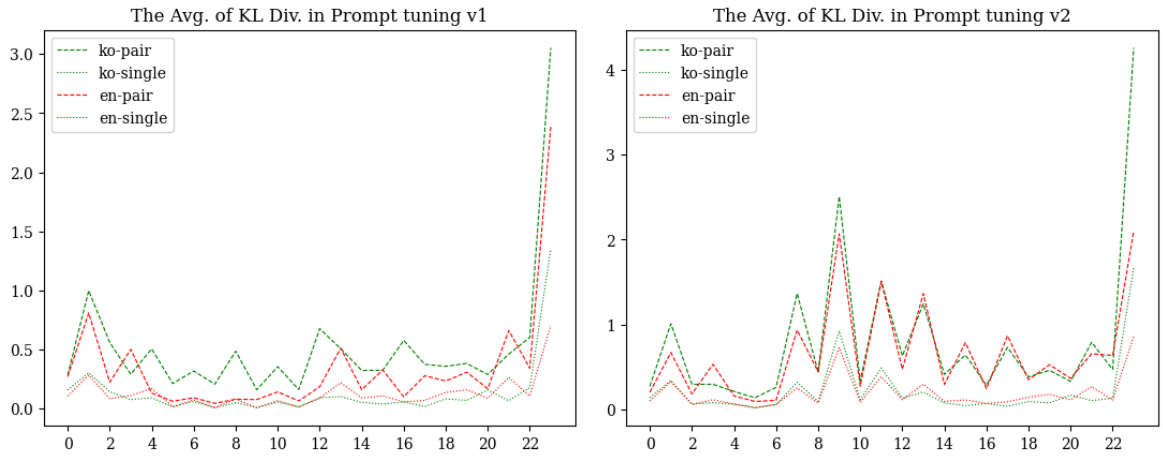

Figure 15: The average KL divergence results grouped by input type.