西安交通大學
XI'AN JIAOTONG UNIVERSITY

Concentrate Attention: Towards Domain-Generalizable Prompt Optimization for Language Models

Chengzhengxu Li, Xiaoming Liu⋆ , Zhaohan Zhang, Yichen Wang, Chen Liu, Yu Lan, Chao Shen
Faculty of Electronic and Information Engineering, Xi'an Jiaotong University
{czx.li, lcoder}@stu.xjtu.edu.cn {xm.liu, ylan2020, chaoshen}@xjtu.edu.cn
zhaohan.zhang@qmul.ac.uk yichenzw@uchicago.edu
⋆ Corresponding author

Agenda

- Motivation
- The Proposed Method
  - Definition of Concentration
  - Pilot Experiment Results
  - Concentrative Soft Prompt Optimization
  - Concentrative Hard Prompt Optimization
- Experiment Results
- Conclusion & Future work

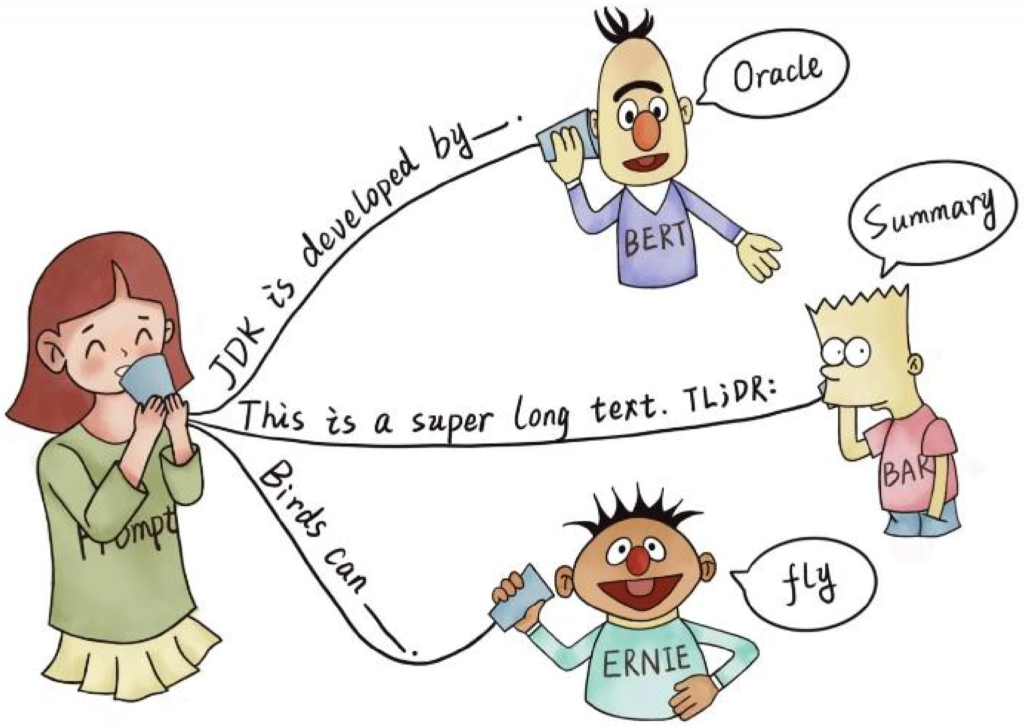
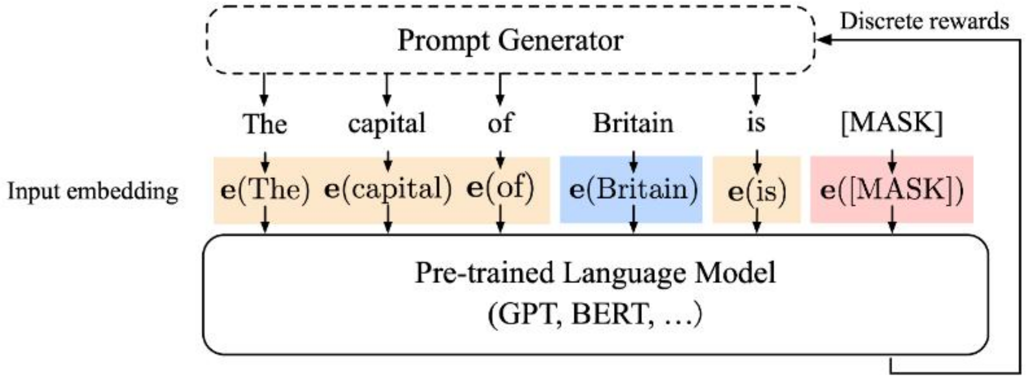


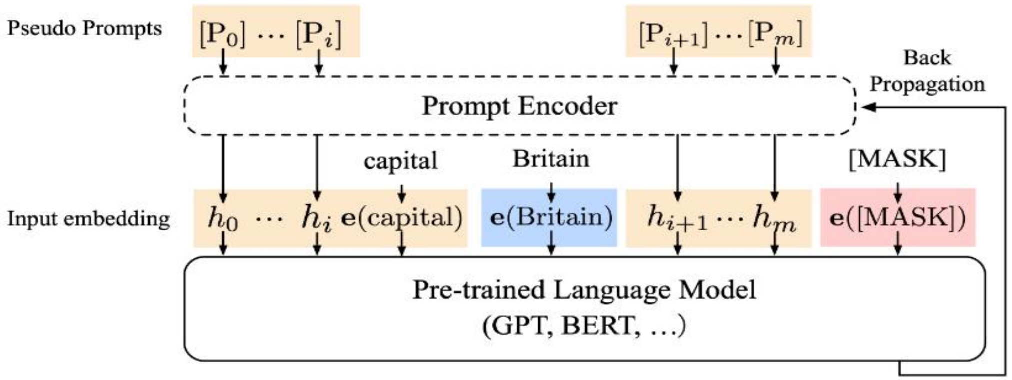
Image source: arXiv:2107.13586

Motivation

- Discrete Prompt Optimization
  - Requiring considerable expertise
  - Inefficient optimization
  - Only one prompt for the whole dataset
- Continuous Prompt Optimization
  - High computational cost
  - Low readability and generalizability
  - Only one prompt for the whole dataset
- Challenges
  - Three challenges for few-shot: **data scarcity, overfitting and noise impact**
  - Two challenges for prompting: **training efficiency and robustness to verbalizers**



Discrete Prompt Optimization



Continuous Prompt Optimization

Image source: arXiv:2103.10385

Motivation

- However, the domain generalization ability of trained prompts still lacks exploration.

- Domain Adaptation Methods
  - Target domain availability based
  - Align target domain by unsupervised learning
  - Serious data reliance

- Pre-training Prompt Methods
  - Large data requirements
  - Low efficiency training
  - High computational cost

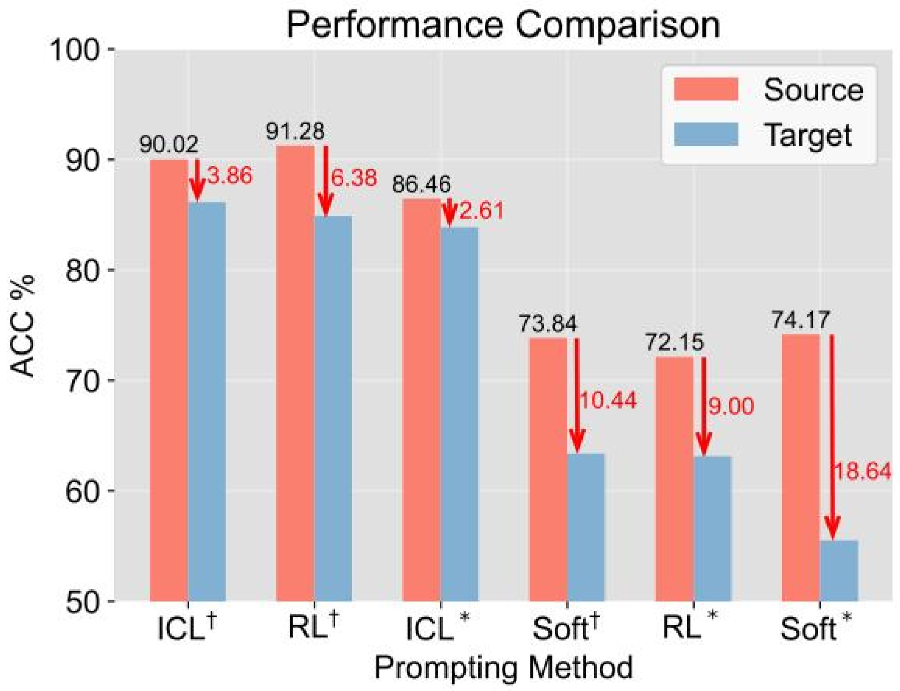


Figure 1. Domain generalization capabilities across various prompting methods in sentiment classification tasks.

Motivation

What Nature Do Well-generalized Prompts Have?

- **Domain Adaptation Methods**
  - Target domain availability based
  - Align target domain by unsupervised learning
  - Serious data reliance

- **Pre-training Prompt Methods**
  - Large data requirements
  - Low efficiency training
  - High computational cost

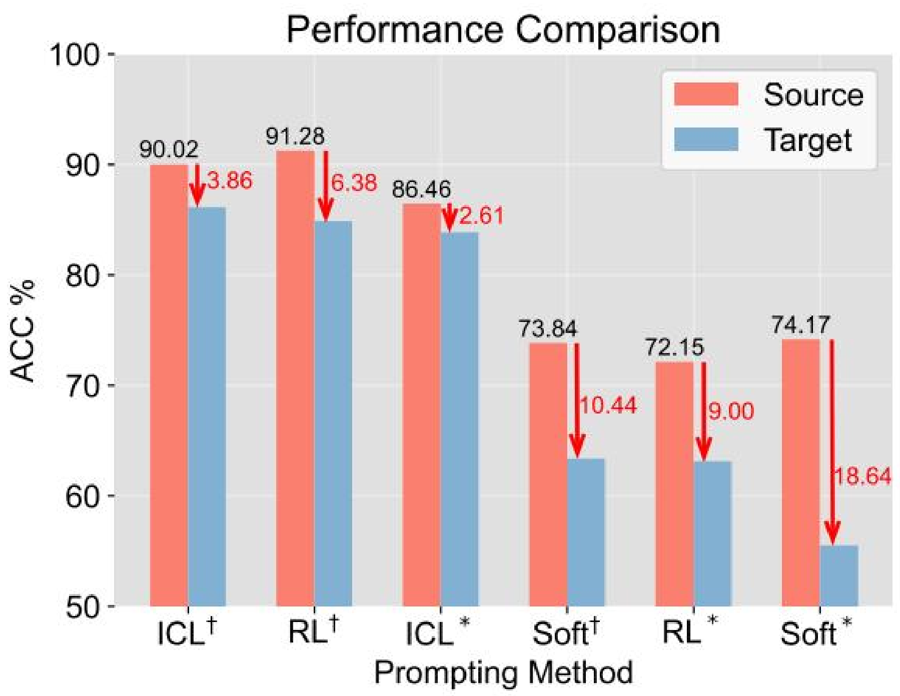


Figure 1. Domain generalization capabilities across various prompting methods in sentiment classification tasks.

The Proposed Method: Definition of Concentration

Heuristicly, concentration represents the "lookback" attention from current decoding token to prompt tokens, as shown in Figure 2.

Definition 3.1. Let $z = (z1, z2, ..., zL)$ and $x = (e1, e2, ..., eT)$ be prompt and original input with $z, x \in S$, where S is the set of all possible textual sequences over the vocabulary. Let $f_{\theta l}$ be the attention block in layer l of a PLM parameterized by θl. Then concentration is a function Concentration : $S \rightarrow R^{+}$

$$\text{Concentration}(z \oplus x; \theta_l) = \sum_{z_i \in z} f_{\theta_l}(z_i \oplus x).$$

Definition 3.2. Let $z = (z_1, z_2, ..., z_L)$ and $x = (e1, e2, ..., eT)$ be prompt and original input with $z, x \in S$, where S is the set of all possible textual sequences over the vocabulary. Let $D = (x_1, x_2, ..., x_M)$ be the input dataset. Let $f_{\theta l}$ be the attention block in layer l of a PLM. Then concentration strength is a function Strength : $D \rightarrow R^{+}$

$$\text{Strength}((z, \mathcal{D}); \theta_l) = \frac{1}{|\mathcal{D}|} \sum_{x_i \in \mathcal{D}} \text{Concentration}(z \oplus x_i; \theta_l).$$

Definition 3.3. Let $D = (x_1, x_2, ..., x_M)$ be the set of textual sequences sampled from target domain D_{Tt} , where $x_i \in S$. Then the concentration fluctuation is a function Fluctuation : $D \rightarrow R^{+}$

$$\text{Fluctuation}((z, \mathcal{D}); \theta_l) = \sqrt{\frac{1}{|\mathcal{D}|} \sum_{x_i \in \mathcal{D}} [\text{Concentration}(z \oplus x_i; \theta_l)) - \text{Strength}((z, \mathcal{D}); \theta_l)]^2}.$$

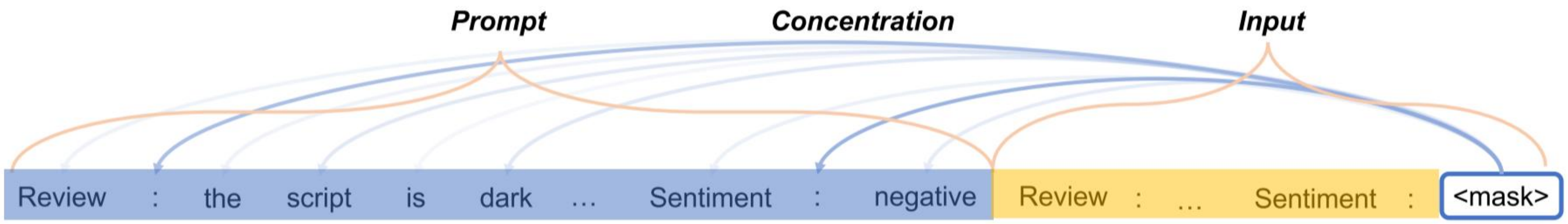
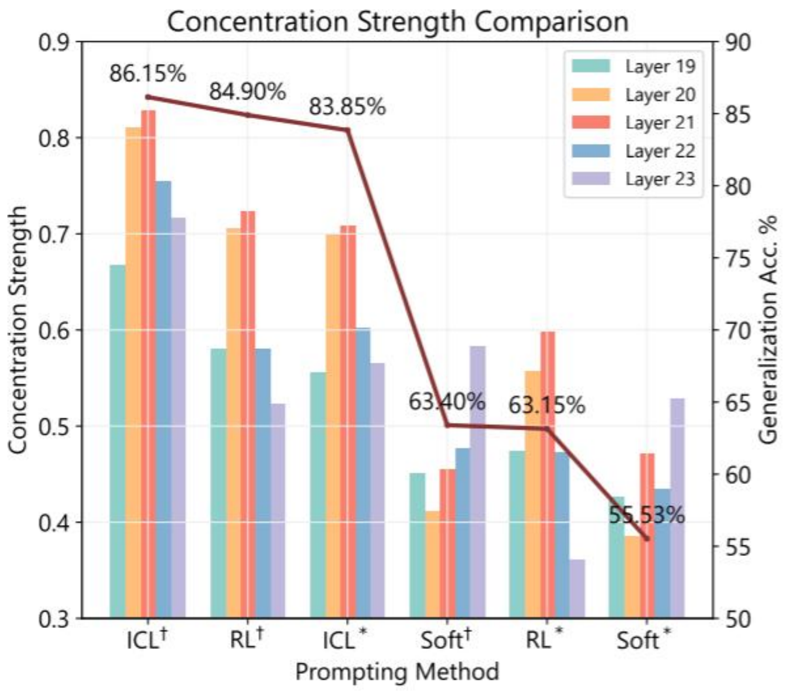


Figure 2: Illustration of Concentration.

The Proposed Method: Pilot Experiment Results

$\mathcal{F}_1$: Prompts gaining *more attention weight* from PLMs' deep layers are more generalizable.
$\mathcal{F}_2$: Prompts with *more stable attention distributions* in PLMs' deep layers generalize better.



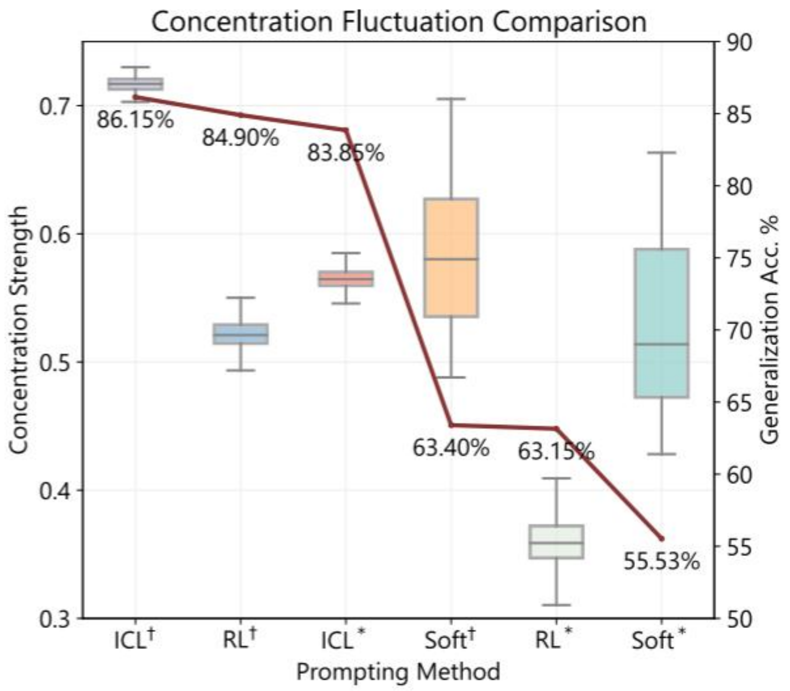


Figure 3. Left: concentration strength of various prompting methods in the last 5 layers (layers 19 to 23). Right: boxplots of the concentration strength in the last layer. Overall, prompts that exhibit good domain generalization gain higher concentration strength and lower concentration fluctuation.

The Proposed Method: Concentrative Soft Prompt Optimization

These methods optimize follow loglikelihood objective given a trainable prompt z and a fixed PLM parameterized by θ for the input x:

$$\max_z \log P(y|(z \oplus x); \theta).$$

According to our findings in §3, domain-generalizable prompts should be high in concentration strength and low in concentration fluctuation. Thus, we reformulate Eq. 6 to get the objective for domain-generalizable prompts:

$$\max_z (\log P(y|(z \oplus x); \theta) + \text{Strength}((z, \mathcal{D}_{\text{train}}); \theta)) \quad s.t. \ \min_z \text{Fluctuation}((z, \mathcal{D}_{\text{train}}); \theta).$$

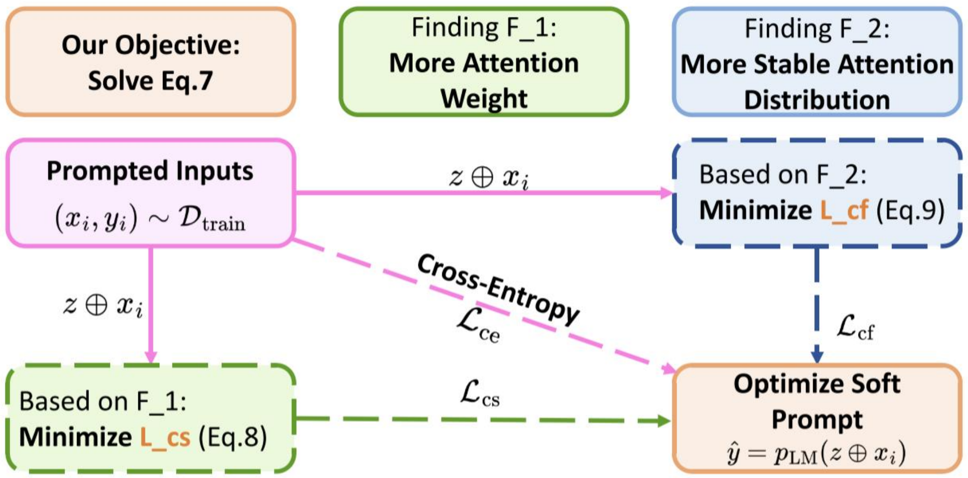


Figure 4. Framework for Soft Prompt Optimization.

Towards the reformulated objective above, we propose the concentration-reweighting loss for soft prompt optimization methods. First, we minimize the concentration strength on input x to improve concentration strength on prompt z by designing loss function L_{cs}. In addition, to reduce concentration fluctuation of prompts, we propose to use every token's concentration strength as hidden state feature of prompts, denoted as Ci = (c1, c2, ..., cL) where L is the length of prompts. We design a contrastive loss to cluster C with same label together to reduce concentration fluctuation:

$$\mathcal{L}_{\text{cf}} = \sum_{i=1}^{|\mathcal{D}_{\text{train}}|} \frac{-1}{P(i)} \sum_{p \in P(i)} \log \frac{\exp(sim(\mathbb{C}_i, \mathbb{C}_p)/\tau)}{\sum_{j=1}^{|\mathcal{D}_{\text{train}}|} \mathbf{1}_{i \neq j} \exp(sim(\mathbb{C}_i, \mathbb{C}_j)/\tau)}, \qquad \mathcal{L}_{\text{cs}} = 1 - \text{Strength}((z, \mathcal{D}_{\text{train}}); \theta).$$

The Proposed Method: Concentrative Hard Prompt Optimization

In contrast to soft prompt optimization, hard prompt optimization searches suitable prompt in discrete space in a non-parameterized fashion. We focus on improving the generalization ability of input-level optimization methods. Generally, the mainstream of input-level optimization technique for hard prompts could be encapsulated as: filter (by metric) and match (by RL agents). The findings of concentration could be applied to this optimization process by adjusting filter metric and agent reward. We illustrate the framework for hard prompt optimization in Figure 5.

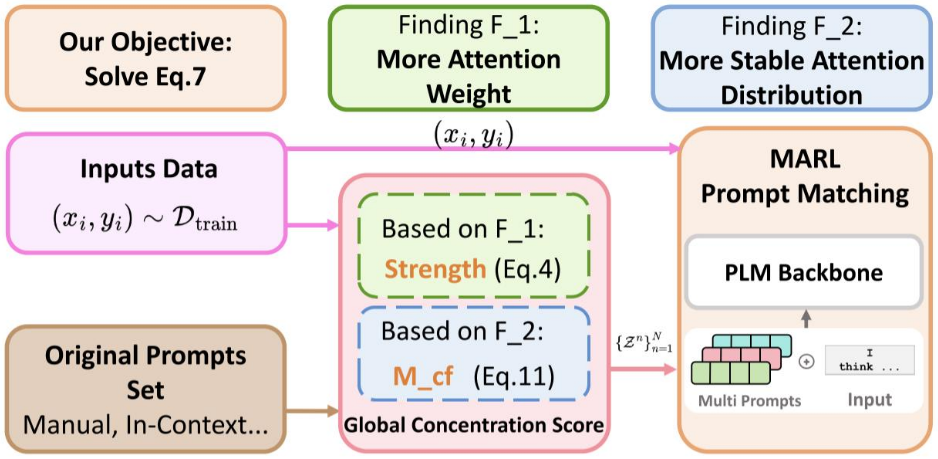


Figure 5. Framework for Hard Prompt Optimization.

For previous filter metric only considering the overall accuracy on training set, we introduce a new metric called Global Concentration Score (GCS), which involves our ideas of concentration strength and concentration fluctuation. We use concentration strength as first metric to filter out prompts which cannot get much concentration from model. Metric for reducing concentration fluctuation could be regarded as minimizing Kullback-Leibler (KL) divergence between the concentration features Ci of input with same label and the average of Ci on whole inputs set Dtrain:

$$M_{\text{cf}}(z, \mathcal{D}_{\text{train}}) = \sum_{y \in \mathcal{Y}} \sum_{i \in \mathcal{D}_{\text{train}}(y)} \text{KL}(\text{Softmax}(\mathbb{C}_i) \parallel \text{Softmax}(\mathbb{C}^y_{\text{avg}}))$$

The Proposed Method: Experiment Results

Paradigms	*Methods*	Sentiment S+M→C	Sentiment C+M→S	Sentiment S+C→M	NLI Q+R→W	NLI W+R→Q	NLI Q+W→R
Prompt Tuning Lester et al. [2021]	Vanilla PT	$64.73_{3.82}$	$65.51_{2.65}$	$65.12_{3.85}$	$41.20_{1.55}$	$49.83_{1.47}$	$49.66_{1.67}$
	PT with $\mathcal{L}_{cs}$	$65.83_{3.83}$	$66.38_{2.42}$	$65.33_{2.37}$	$41.53_{1.56}$	$49.60_{2.37}$	$49.43_{1.41}$
	PT with $\mathcal{L}_{cf}$	$65.09_{3.72}$	$67.40_{2.43}$	$65.40_{2.33}$	$42.17_{2.03}$	$49.22_{1.59}$	$49.73_{1.31}$
	PT with both	**$66.19_{3.69}$**	**$69.54_{2.52}$**	**$65.89_{2.32}$**	**$42.48_{1.72}$**	**$50.31_{1.33}$**	**$50.42_{1.34}$**
Prefix Tuning Li and Liang [2021]	Vanilla Prefix	$65.91_{3.24}$	$83.25_{0.41}$	$75.51_{0.91}$	$50.26_{0.31}$	$51.88_{0.29}$	$50.02_{0.28}$
	Prefix with $\mathcal{L}_{cs}$	$66.23_{3.37}$	$84.32_{0.48}$	$76.58_{0.82}$	$50.69_{0.33}$	$51.44_{0.28}$	$49.77_{0.22}$
	Prefix with $\mathcal{L}_{cf}$	$66.82_{3.19}$	$83.70_{0.39}$	$77.17_{0.75}$	$51.73_{0.32}$	$52.12_{0.28}$	$50.73_{0.26}$
	Prefix with both	**$68.29_{2.97}$**	**$85.07_{0.42}$**	**$77.53_{0.43}$**	**$52.05_{0.30}$**	**$53.32_{0.25}$**	**$51.26_{0.27}$**
P-Tuning v2 Liu et al. [2021]	Vanilla Pv2	$65.92_{1.61}$	$83.84_{1.69}$	$75.89_{0.36}$	$50.63_{0.31}$	$52.76_{1.01}$	$51.31_{1.37}$
	Pv2 with $\mathcal{L}_{cs}$	$66.06_{1.77}$	$83.32_{1.59}$	$75.07_{0.35}$	$51.37_{0.37}$	$50.93_{0.92}$	$50.20_{1.30}$
	Pv2 with $\mathcal{L}_{cf}$	$66.72_{1.62}$	$84.12_{1.51}$	$76.41_{0.33}$	$51.32_{0.38}$	$52.64_{1.04}$	$51.28_{1.22}$
	Pv2 with both	**$67.07_{1.53}$**	**$84.86_{1.42}$**	**$77.26_{0.37}$**	**$51.87_{0.28}$**	**$53.83_{0.95}$**	**$51.57_{1.16}$**
GrIPS Prasad et al. [2022]	-	$80.07_{2.57}$	$84.28_{1.38}$	$85.19_{1.12}$	$54.37_{2.40}$	$52.77_{1.73}$	$53.52_{1.66}$
RLPrompt Deng et al. [2022]	-	$86.05_{1.32}$	$89.36_{0.91}$	$85.95_{1.90}$	$52.77_{2.82}$	$53.82_{2.34}$	$54.63_{1.39}$
Manual Prompt Bach et al. [2022]	Vanilla MP	$52.73_{4.43}$	$55.81_{3.31}$	$50.85_{1.58}$	$41.70_{1.17}$	$50.80_{0.84}$	$51.60_{1.50}$
	MP with MARL	$56.37_{1.18}$	$58.42_{0.46}$	$52.15_{0.49}$	$44.27_{1.02}$	$51.36_{0.84}$	$52.18_{1.23}$
	MP with Metric	$54.63_{2.12}$	$57.84_{1.65}$	$51.79_{1.75}$	$42.86_{0.94}$	$51.02_{0.68}$	$52.03_{1.14}$
	MP with both	**$56.76_{0.40}$**	**$59.44_{0.32}$**	**$53.15_{0.35}$**	**$45.05_{0.28}$**	**$52.03_{0.25}$**	**$52.46_{1.24}$**
In-Context Demo Brown et al. [2020]	Vanilla IC	$84.33_{2.15}$	$84.81_{1.39}$	$80.21_{2.17}$	$50.86_{1.28}$	$52.63_{0.94}$	$58.04_{2.23}$
	IC with MARL	$85.33_{5.03}$	$87.02_{2.74}$	$82.14_{1.65}$	$52.82_{3.29}$	$53.75_{1.32}$	$59.87_{2.07}$
	IC with Metric	$84.70_{3.17}$	$85.10_{2.12}$	$82.60_{4.91}$	$51.19_{4.80}$	$52.72_{4.31}$	$59.46_{4.64}$
	IC with both	**$87.29_{2.72}$**	**$88.49_{1.52}$**	**$83.52_{0.98}$**	**$52.94_{1.59}$**	**$54.24_{0.73}$**	**$60.32_{1.20}$**
DP_2O Li et al. [2024]	Vanilla DP_2O	$89.06_{0.76}$	$90.75_{0.91}$	$86.53_{0.80}$	$54.84_{0.62}$	$54.85_{0.37}$	$59.78_{0.79}$
	DP_2O with MARL	$87.36_{3.17}$	$91.60_{2.39}$	$86.03_{4.03}$	$54.71_{2.21}$	$53.13_{1.97}$	$60.62_{3.47}$
	DP_2O with Metric	$86.79_{1.32}$	$90.13_{1.07}$	$86.60_{0.83}$	$53.21_{1.16}$	$54.02_{0.79}$	$60.54_{1.47}$
	DP_2O with both	**$89.63_{0.52}$**	**$92.87_{0.33}$**	**$87.85_{0.47}$**	**$56.42_{0.36}$**	**$55.32_{0.33}$**	**$61.27_{0.81}$**

Table 1. Performance comparison of text classification tasks in accuracy with MFDG setting.

Paradigms	*Methods*	Sentiment SST-2	Sentiment CR	Sentiment MR	Sentiment Avg.	NLI WNLI	NLI QNLI	NLI RTE	NLI Avg.	Avg Gap
Prompt Tuning	Vanilla PT	$73.84_{3.52}$	$75.89_{1.72}$	$74.17_{2.32}$	74.63	$47.64_{1.02}$	$49.71_{0.93}$	$54.73_{1.72}$	50.69	+6.66
	PT with both	$72.61_{2.72}$	$76.07_{2.24}$	$74.37_{2.12}$	74.35	$46.79_{1.52}$	$49.50_{0.98}$	$54.21_{1.49}$	50.17	+4.71
Prefix Tuning	Vanilla Prefix	$87.39_{2.98}$	$77.37_{0.79}$	$82.65_{0.65}$	82.47	$55.88_{0.37}$	$60.27_{0.44}$	$54.82_{0.31}$	56.99	+6.93
	Prefix with both	$87.29_{3.12}$	$76.73_{1.28}$	$83.32_{0.83}$	82.45	$56.18_{0.35}$	$59.74_{0.32}$	$55.38_{0.42}$	57.10	+5.19
P-Tuning v2	Vanilla Pv2	$86.71_{1.57}$	$77.65_{1.49}$	$82.27_{0.42}$	82.21	$55.57_{0.73}$	$60.73_{1.64}$	$55.16_{1.83}$	57.15	+6.29
	Pv2 with both	$87.03_{1.32}$	$77.71_{1.50}$	$82.05_{0.54}$	82.08	$56.31_{0.69}$	$60.46_{1.37}$	$55.20_{1.69}$	57.32	+5.38
Manual Prompt	Vanilla MP	$61.62_{3.42}$	$57.75_{2.92}$	$53.13_{2.33}$	57.50	$44.27_{2.80}$	$53.42_{0.98}$	$52.63_{0.60}$	50.11	+3.22
	MP with both	$61.33_{2.32}$	$56.07_{1.61}$	$53.47_{0.42}$	56.96	$44.05_{0.89}$	$53.77_{1.35}$	$52.60_{0.39}$	50.14	+0.40
In-Context Demo	Vanilla IC	$85.91_{1.42}$	$85.57_{0.92}$	$83.75_{1.39}$	85.08	$52.37_{1.45}$	$53.42_{0.72}$	$59.73_{0.81}$	55.17	+1.65
	IC with both	$86.33_{1.34}$	$85.14_{0.87}$	$84.31_{2.12}$	85.26	$52.25_{1.62}$	$52.96_{0.49}$	$59.36_{0.73}$	54.86	+0.93
DP_2O	Vanilla DP_2O	$93.62_{0.72}$	$90.76_{0.50}$	$88.58_{0.91}$	90.99	$55.26_{1.02}$	$55.13_{0.39}$	$61.07_{0.81}$	57.15	+1.44
	DP_2O with both	$93.20_{0.81}$	$90.38_{0.47}$	$88.37_{2.12}$	90.65	$56.47_{0.41}$	$55.42_{0.79}$	$61.29_{0.63}$	57.73	+0.37

Table 2. In-domain comparison.

Models	*Methods*	MCQA S + C → R	MCQA C + R → S	MCQA R + S → C	MCQA Avg.	Acc Gap
Llama-2-7b-chat	Vanilla Prefix	$62.32_{2.15}$	$66.30_{2.30}$	$73.15_{2.53}$	67.26	—
	Prefix with both	$63.70_{1.96}$	$68.47_{0.97}$	$75.32_{1.09}$	69.16	+1.90
	Vanilla IC	$63.13_{1.25}$	$65.50_{1.98}$	$77.59_{1.14}$	68.74	—
	IC with both	$65.13_{1.03}$	$68.33_{2.13}$	$79.83_{0.88}$	70.10	+1.36
Vicuna-7b-v1.5	Vanilla Prefix	$67.72_{1.79}$	$81.09_{2.17}$	$88.97_{2.64}$	79.26	—
	Prefix with both	$68.75_{1.04}$	$83.93_{1.79}$	$89.76_{2.60}$	80.81	+1.55
	Vanilla IC	$68.37_{2.24}$	$83.23_{4.12}$	$90.98_{1.99}$	80.86	—
	IC with both	$69.67_{1.58}$	$85.50_{5.06}$	$93.39_{1.23}$	82.85	+1.99
Alpaca-7b-wdiff	Vanilla Prefix	$61.52_{3.79}$	$70.03_{2.88}$	$87.91_{2.73}$	73.15	—
	Prefix with both	$63.89_{2.93}$	$72.15_{2.07}$	$89.58_{2.81}$	75.21	+2.06
	Vanilla IC	$60.81_{1.14}$	$69.11_{2.46}$	$89.66_{2.37}$	73.19	—
	IC with both	$63.16_{1.56}$	$70.57_{1.95}$	$91.19_{2.00}$	74.97	+1.78

Table 3: Performance comparison of large models on MCQA task accuracy.

Conclusion & Future work

- To explore the nature of prompt generalization on unknown domains, we conduct pilot experiments and find that (i) Prompts gaining more attention weight from PLMs' deep layers are more generalizable and (ii) Prompts with more stable attention distributions in PLMs' deep layers are more generalizable.
- We adapt this new objective to popular soft prompt and hard prompt optimization methods, respectively. Extensive experiments demonstrate that our idea improves comparison prompt optimization methods by 1.42% for soft prompt generalization and 2.16% for hard prompt generalization in accuracy , while maintaining satisfying in-domain performance.
- Future improvement:
  - Try to systematically study more cues and attention distribution phenomena;
  - Try to apply this objective to more complex downstream tasks (e.g., open-ended generation tasks);

Thanks!

Paper

Contact

Code