# Generating Synthetic Datasets for Few-shot Prompt Tuning

**Xu Guo,** Zilin Du, Boyang Li, Chunyan Miao

Conference on Language Modeling, 2024

# Background

Manually crafting prompts for each task is challenging.

Prompt tuning, which learns "soft" prompts from a labeled dataset, can outperform manual prompts and closes the gap with model tunning (Pros).

However, it requires a sufficiently large labeled dataset to be effective. In few-shot learning scenarios, it significantly underperforms model tuning (Cons).

Can we synthesize a labeled training set for each (low-resource) task?

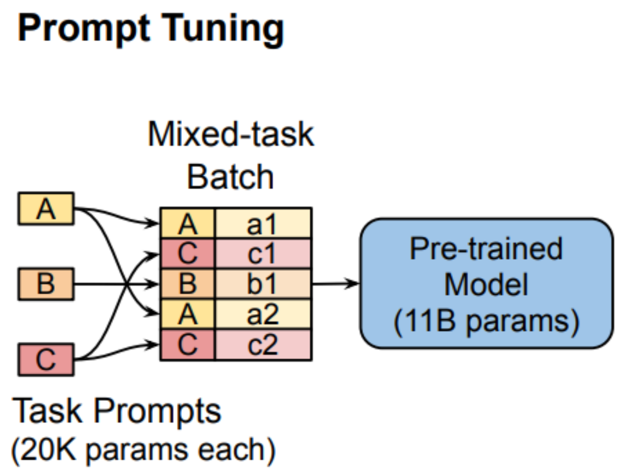



Image source [1]

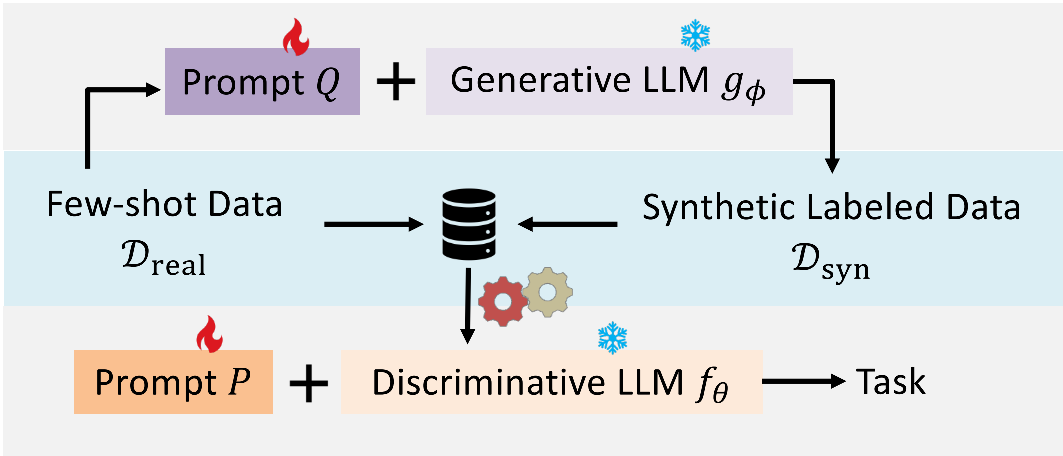
[1] The Power of Scale for Parameter-Efficient Prompt Tuning

# Methodology


Requirements

1. Domain-relevant: e.g., the writing style for movie reviews
2. Label relevant: e.g., being recognized as negative

# DawGen: Distribution-Aligned Weighted GENerator tuning




Step 1: Generator tuning

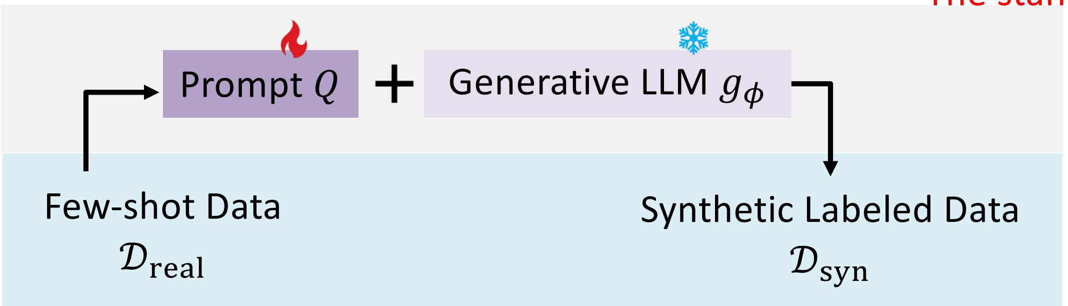
Step 2: Generating Data

Step 3: Train Soft Prompts

# Learning to generate **domain-relevant tokens**

$$\mathcal{L}_{\text{gen}}(Q_l) = -\frac{1}{|\mathcal{D}_{real}|} \sum_{X \in \mathcal{D}_{real}, y = Y_l} \sum_{x_j \in X} \log \Pr_{\phi}(x_j | x_{<j}; Q_l).$$

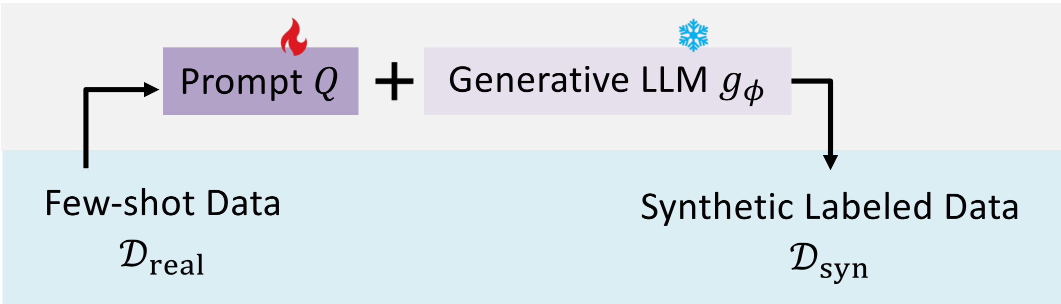
The standard LM loss treats all tokens equally



# Learning to generate **label-relevant tokens**

$$\mathcal{L}_{\text{wGen}}(Q_l) = -\mathbb{E}_{X \in \mathcal{D}_{real}, Y=Y_l} \mathbb{E}_{x_j \in X} W_j \cdot \log \Pr_\phi(x_j | x_{<j}; Q_l).$$

Optimize the generation of the label-discriminative tokens




$$\mathcal{L}_{\text{disc}}(W) = -\mathbb{E}_{x_j \in X} \frac{\Pr_\phi(x_j | x_{<j}; Q_l(W))}{\sum_{l'} \Pr_\phi(x_j |, x_{<j}; Q_{l'}(W))}.$$

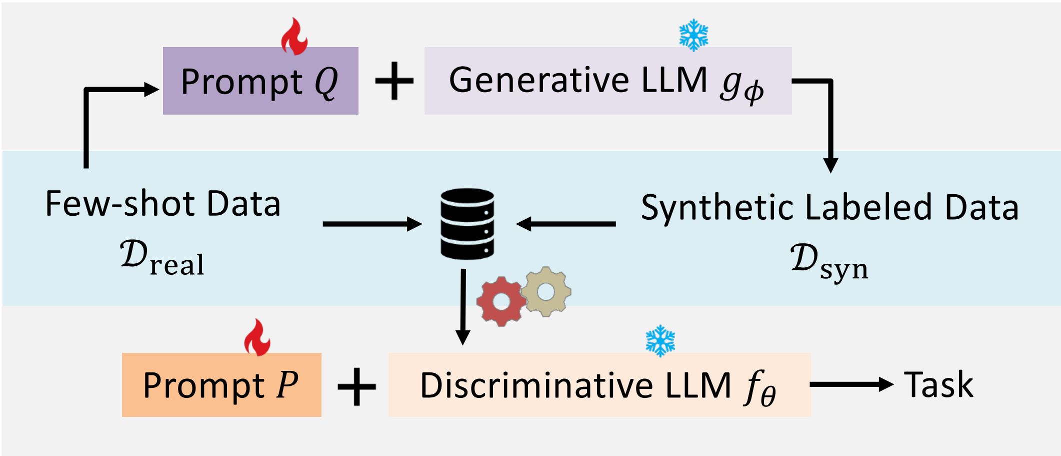
generate tokens that are more related to the given label than other labels

“It is not a good movie” “not” is not a generalizable label-discriminative token

Sentence-level regularization:

$$\mathcal{L}_{\text{dist}}(Q) = \mathbb{E}_{(X,y) \in \mathcal{D}_{real}} max(0, 1 - D(W \cdot Z_{i,l}, W \cdot Z_{j,l}) + D(W \cdot Z_{i,l}, W \cdot Z_{j,l'}))$$

# Mitigating Data Discrepancy




$$\text{Proj}_{\delta_{real}}(\delta_{syn}) = \frac{\delta_{syn} \cdot \delta_{real}}{\delta_{real} \cdot \delta_{real}} \delta_{real} = \frac{\delta_{syn} \cdot \delta_{real}}{\| \delta_{real} \|} \cdot \frac{\delta_{real}}{\| \delta_{real} \|}$$

**Algorithm 2:** Prompt Tuning.

**Data:** Few-shot $\mathcal{D}_{real}$ and synthetic $\mathcal{D}_{syn}$.
**Initialize:** Prompts $P$, Pre-trained LLM $f_\theta$.
initialize $t = 0$;
**while** $t < T + 1$ **do**
  $t+ = 1$;
  $\mathcal{B}_{real}, \mathcal{B}_{syn} \leftarrow$ Sample a batch from $\mathcal{D}_{real}, \mathcal{D}_{syn}$;
  Compute gradients $\delta_{real} = \frac{\partial \mathcal{L}_{ce}(P)}{\partial P}$ on $\mathcal{B}_{real}$;
  Compute gradients $\delta_{syn} = \frac{\partial \mathcal{L}_{ce}(P)}{\partial P}$ on $\mathcal{B}_{syn}$;
  **if** $\delta_{syn} \cdot \delta_{real} < 0$ **then**
    $\delta'_{\text{syn}} = \delta_{syn} - \text{Proj}_{\delta_{real}}(\delta_{syn})$;
  **end**
  $\delta = \delta_{real} + \epsilon \cdot \delta'_{syn}$;
  $P \leftarrow P - \eta \cdot \delta$;
**end**
**Output:** Soft prompt $P$

Gradient surgery

# Evaluation setting

- Generating task-specific training data for 7 tasks:
  - Paraphrase detection: QQP, MRPC
  - Natural Language Inference: SNLI, MNLI, QNLI, RTE, SICK
- Data preparation: 16-shot per class sampled with 5 random seeds
- Model backbones: T5-large, Flan-T5-large (further tuned with instruction datasets)
- Baselines
  - Zero-shot prompting
  - Few-shot prompting (i.e., in-context learning)
  - Full-model Fine-tuning (FT)
  - Prompt-based Full-model Fine-tuning (PFT)
  - Pre-trained Prompt Tuning (PPT)
  - FewGen for synthetic data generation

# Comparisons with model tuning and transfer learning

- Using the synthetic data produced by DawGen improves few-shot prompt tuning performance by ~18% on average and outperforms full-model tuning by 3.8% on T5-large.
- The results are competitive with transfer learning using a large real-world dataset on QQP, MRPC, and SICK.

| Method | #Trainable Params | QQP | MRPC | MNLI | SNLI | QNLI | RTE | SICK | AVG |
|---|---|---|---|---|---|---|---|---|---|
| | | | | T5-large | | | | | |
| Prompting | 0 | 42.63 | 33.80 | 33.20 | 33.31 | 49.46 | 52.35 | 14.51 | 37.04 |
| In-Context | | 59.55 | 33.52 | 34.49 | 33.80 | 49.72 | 48.52 | 40.90 | 42.93 |
| FT | 770M | **72.50** | 61.72 | 42.82 | 48.90 | 50.11 | 55.81 | **77.90** | 58.54 |
| Prompt-based FT | | 60.15 | 59.66 | 42.94 | **54.16** | 51.75 | 57.18 | 69.98 | 56.55 |
| PFT + soft prompt | | 60.22 | 56.18 | 43.86 | 48.45 | 57.38 | 55.60 | 76.23 | 56.85 |
| PPT | 410K | 46.11 | 52.37 | 34.05 | 35.28 | 52.86 | 48.59 | 45.64 | 44.99 |
| Prompt Tuning | 102K | 47.28 | 58.94 | 33.29 | 33.21 | 52.68 | 51.70 | 27.80 | 43.49 |
| Ours | 102K | 66.77 | **69.67** | **53.20** | 46.81 | **69.84** | **57.40** | 72.73 | **62.35** |
| SPOT† | 102K | 64.5 | 68.7 | 74.3 | 78.8 | - | - | 72.9 | - |
| OPTIMA† | | 69.1 | 71.2 | 78.4 | 82.1 | - | - | 73.3 | - |
| | | | | Flan-T5-large | | | | | |
| Prompting | 0 | 62.15 | 67.71 | 62.13 | 64.07 | 80.29 | 26.35 | 33.31 | 56.57 |
| In-Context | | **82.84** | 75.27 | 62.44 | 54.87 | **89.98** | 19.06 | 38.02 | 60.35 |
| FT | 770M | 79.17 | 78.29 | 79.76 | 86.37 | 56.86 | **86.57** | **83.73** | 78.68 |
| Prompt-based FT | | 80.28 | 78.04 | 78.42 | **88.11** | 50.56 | 84.84 | 80.96 | 77.32 |
| PFT + soft prompt | | 79.64 | 77.65 | **79.87** | 86.90 | 80.37 | 84.91 | 70.60 | **79.99** |
| Prompt Tuning | 102K | 70.40 | 72.82 | 59.89 | 63.26 | 83.73 | 26.78 | 60.61 | 62.49 |
| Ours | | 82.14 | **78.40** | 71.84 | 82.43 | 88.80 | 56.82 | 79.88 | 77.19 |

# Investigating data combination strategies

- How synthetic and real data interact during learning is crucial.
- A naive combination can lead to poor results, and label smoothing offers limited benefits.
- Starting with real data doesn't always enhance learning but pairing it with synthetic data yields better results.

| Method | Generator | QQP | MRPC | MNLI | SNLI | QNLI | RTE | SICK | AVG |
|---|---|---|---|---|---|---|---|---|---|
| T5-Large | | | | | | | | | |
| Real+Syn | FewGen | 52.64 | **70.53** | 38.15 | 33.96 | 57.08 | 52.64 | 48.01 | 50.43 |
| Real+Syn+LS | FewGen | 53.09 | 67.94 | 38.58 | 34.11 | 56.99 | 56.03 | 58.05 | 52.11 |
| Real → Syn | FewGen | 59.51 | 70.04 | 47.46 | 41.99 | **65.52** | **57.91** | 65.81 | 58.32 |
| Syn → Real | FewGen | 63.78 | 68.92 | 36.97 | 35.17 | 63.24 | 53.86 | 52.90 | 53.55 |
| (Real, Syn) | FewGen | **66.44** | 68.12 | **48.03** | **44.81** | 64.15 | 56.54 | **68.46** | **59.51** |
| Real+Syn | DawGen | **62.86** | **70.38** | 43.97 | 35.62 | 60.11 | 53.29 | 51.31 | 53.93 |
| Real → Syn | DawGen | 62.60 | 69.39 | 47.94 | 46.14 | 66.33 | **58.12** | 61.48 | 58.85 |
| Syn → Real | DawGen | 62.69 | 69.28 | 42.45 | 38.01 | 60.35 | 55.31 | 56.20 | 54.89 |
| (Real, Syn) | DawGen | 61.77 | 69.99 | **48.76** | **45.10** | **66.37** | 57.20 | **70.80** | **59.99** |
| Flan-T5-large | | | | | | | | | |
| Real+Syn | FewGen | 81.13 | 76.82 | 67.91 | 66.79 | 85.32 | 54.01 | 75.04 | 72.43 |
| Real+Syn+LS | FewGen | 79.56 | 76.06 | **73.50** | 71.42 | 82.96 | 55.88 | 70.37 | 72.82 |
| Real → Syn | FewGen | 79.09 | 76.08 | 64.94 | 63.46 | 85.76 | 57.19 | 71.15 | 71.10 |
| Syn → Real | FewGen | 82.18 | **79.00** | 68.35 | 72.24 | 82.08 | **58.70** | 77.88 | 74.34 |
| (Real, Syn) | FewGen | **82.33** | 78.04 | 68.86 | **80.14** | **87.19** | 56.68 | **78.56** | **75.97** |
| Real+Syn | DawGen | 83.60 | 76.81 | 71.85 | 72.48 | 84.11 | 53.72 | 69.17 | 73.10 |
| Real → Syn | DawGen | 80.50 | 75.64 | 66.42 | 69.41 | 86.52 | **54.22** | 73.33 | 72.29 |
| Syn → Real | DawGen | **83.26** | **78.55** | **72.15** | 77.29 | **87.51** | 50.76 | 72.17 | 74.53 |
| (Real, Syn) | DawGen | 81.83 | 76.96 | 70.18 | **79.69** | 87.38 | 51.63 | **76.97** | **74.94** |

# Ablation study

- Distribution-aligned regularization improves synthetic data quality.
- Using few-shot data to enhance learning from synthetic data is helpful.
- Gradient surgery (GS) reconciles learning conflits from real and synthetic data.

| Generator | Real | GS | QQP | MRPC | MNLI | SNLI | QNLI | RTE | SICK | AVG |
|---|---|---|---|---|---|---|---|---|---|---|
| T5-Large | | | | | | | | | | |
| FewGen | | | 56.70 | 69.25 | 42.18 | 34.63 | 56.91 | 53.87 | 34.11 | 49.66 |
| | ✓ | | 66.44 | 68.12 | 48.03 | 44.81 | 64.15 | 56.54 | 68.46 | 59.51 |
| | ✓ | ✓ | 67.85 | 70.05 | 49.52 | 46.39 | 66.96 | 55.96 | 72.08 | 61.25 |
| DawGen | | | 58.62 | 69.11 | 44.30 | 36.63 | 61.97 | 55.16 | 50.39 | 53.74 |
| | ✓ | | 61.77 | 69.99 | 48.76 | 45.10 | 66.37 | 57.20 | 70.80 | 59.99 |
| | ✓ | ✓ | 66.77 | 69.67 | 53.20 | 46.81 | 69.84 | 57.40 | 72.73 | 62.35 |
| Flan-T5-large | | | | | | | | | | |
| FewGen | | | 78.29 | 78.72 | 62.03 | 66.13 | 84.00 | 50.54 | 69.81 | 69.93 |
| | ✓ | | 82.33 | 78.04 | 68.86 | 80.14 | 87.19 | 56.68 | 78.56 | 75.97 |
| | ✓ | ✓ | 81.76 | 78.36 | 75.26 | 81.07 | 87.67 | 61.13 | 79.26 | 77.78 |
| DawGen | | | 83.11 | 77.05 | 68.21 | 70.49 | 84.07 | 49.02 | 68.66 | 71.52 |
| | ✓ | | 81.83 | 76.96 | 70.18 | 79.69 | 87.38 | 51.63 | 76.97 | 74.94 |
| | ✓ | ✓ | 82.14 | 78.40 | 71.84 | 82.43 | 88.80 | 56.82 | 79.88 | 77.19 |

# Conclusion

This paper presents a framework for generating synthetic training data with LLMs to boost prompt tuning in few-shot settings.

- We introduce Distribution-Aligned Weighted GENerator tuning (DawGen) a framework that leverages LLMs to generate label-relevant synthetic data, enhancing prompt tuning in few-shot settings.
- By applying gradient surgery, DawGen effectively integrates synthetic and real data, eliminating conflicting gradients.
- Experiments on seven sentence-pair classification datasets and two LLM backbones demonstrate that our method significantly improves prompt tuning performance with limited real data.

# Limitations: Efficiency

- High Inference Load: Generating a large number of synthetic samples simultaneously can lead to increased GPU or CPU usage, prolonging processing times.
- Throughput Constraints: Managing concurrent processes for data generation is hindered by the limit of RAM and GPU memory.
- High Energy Consumption: The extensive computational operations involved in data generation lead to higher energy consumption, increasing environmental impact.