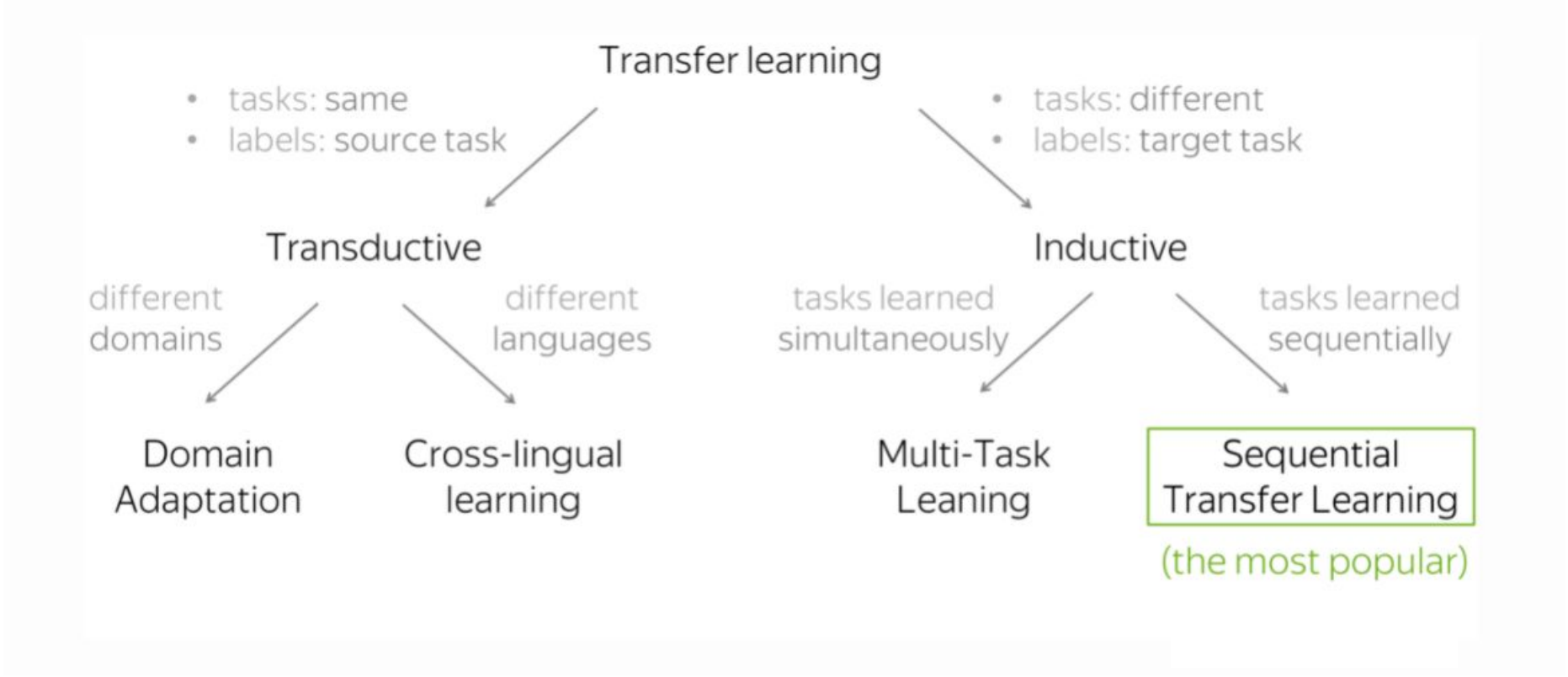
# LoRA: Low-Rank Adaptation of Large Language Models[1]

Докладчик - Артём Щербинин, 181
Рецензент - Конодюк Никита, 181
Практик-исследователь - Пахалко Илья, 181
Хакер - Смирнов Тимофей, 181

[1] https://arxiv.org/abs/2106.09685

# Transfer Learning



# Формулировка задачи Transfer Learning

Стандартное решение:

$$\max_{\Phi} \sum_{(x,y)\in\mathcal{Z}} \sum_{t=1}^{|y|} \log\left(P_{\Phi}(y_t|x, y_{<t})\right)$$

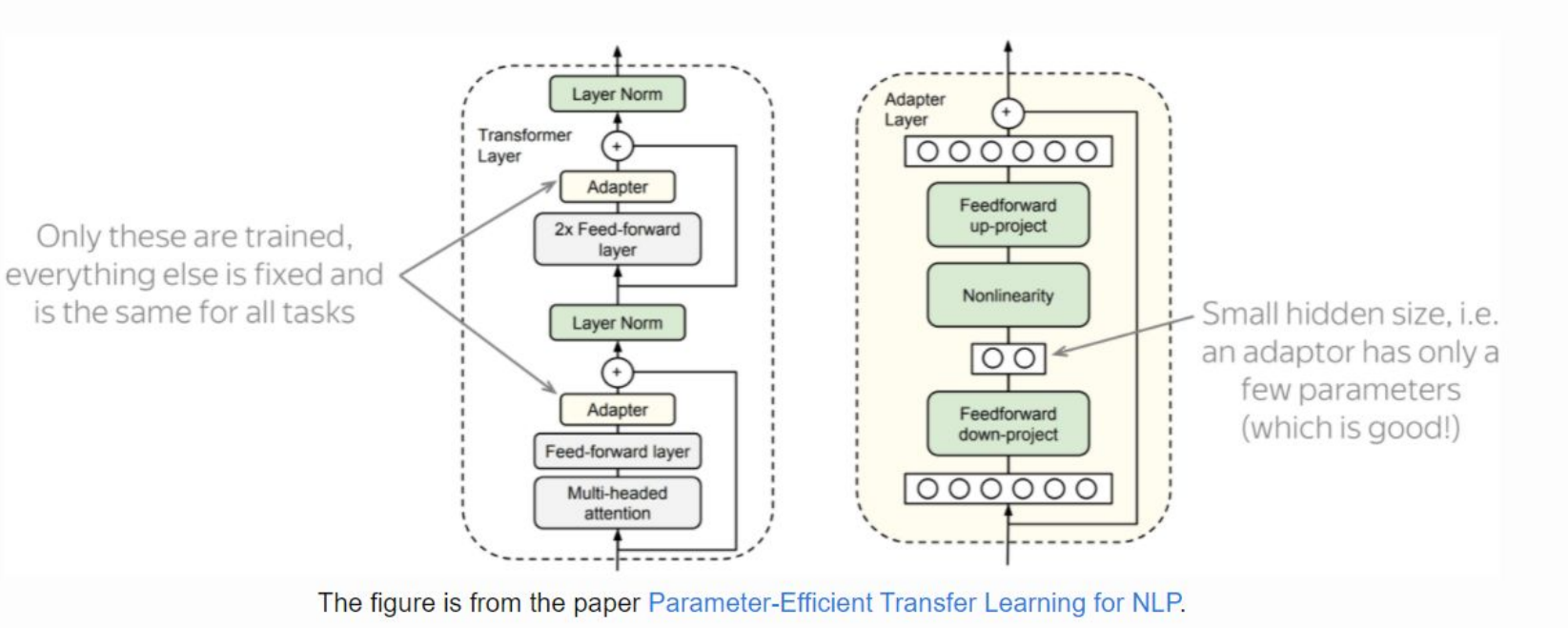
$$\Phi = \Phi_0 + \Delta\Phi$$

LoRA идея:

$$\max_{\Theta} \sum_{(x,y)\in\mathcal{Z}} \sum_{t=1}^{|y|} \log\left(p_{\Phi_0+\Delta\Phi(\Theta)}(y_t|x, y_{<t})\right)$$

$$\Delta\Phi = \Delta\hat{\Phi}(\Theta)$$

# Альтернативные подходы: Адаптеры

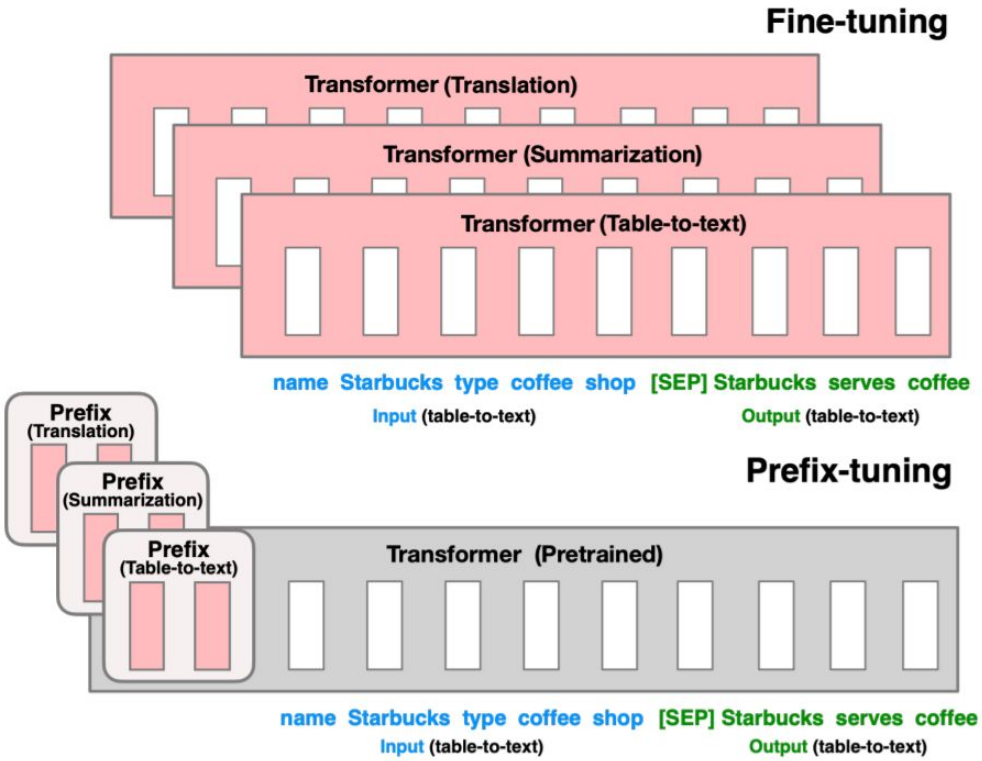



The figure is from the paper Parameter-Efficient Transfer Learning for NLP.

$$|\Theta| = \hat{L}_{Adpt} \times (2 \times d_{model} \times r + r + d_{model}) + 2 \times \hat{L}_{LN} \times d_{model}$$

$\hat{L}_{Adpt}$ - количество adapter слоёв в модели

# Альтернативные подходы: Prefix-Tuning




$$|\Theta| = d_{model} \times (l_p + l_i)$$

# Метод LoRA

$$W_0 + \Delta W = W_0 + BA, \text{ where } B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}$$

$$|\Theta| = 2 \times \hat{L}_{LoRA} \times d_{model} \times r$$

$\hat{L}_{LoRA}$ - Количество матриц, к которым применяем метод

# Бейзлайны: RoBERTa

| Model & Method | # Trainable Parameters | MNLI | SST-2 | MRPC | CoLA | QNLI | QQP | RTE | STS-B | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| $\text{RoB}_{\text{base}}$ (FT)* | 125.0M | **87.6** | 94.8 | 90.2 | **63.6** | 92.8 | **91.9** | 78.7 | 91.2 | 86.4 |
| $\text{RoB}_{\text{base}}$ (BitFit)* | 0.1M | 84.7 | 93.7 | **92.7** | 62.0 | 91.8 | 84.0 | 81.5 | 90.8 | 85.2 |
| $\text{RoB}_{\text{base}}$ ($\text{Adpt}^{\text{D}}$)* | 0.3M | $87.1_{\pm.0}$ | $94.2_{\pm.1}$ | $88.5_{\pm1.1}$ | $60.8_{\pm.4}$ | $93.1_{\pm.1}$ | $90.2_{\pm.0}$ | $71.5_{\pm2.7}$ | $89.7_{\pm.3}$ | 84.4 |
| $\text{RoB}_{\text{base}}$ ($\text{Adpt}^{\text{D}}$)* | 0.9M | $87.3_{\pm.1}$ | $94.7_{\pm.3}$ | $88.4_{\pm.1}$ | $62.6_{\pm.9}$ | $93.0_{\pm.2}$ | $90.6_{\pm.0}$ | $75.9_{\pm2.2}$ | $90.3_{\pm.1}$ | 85.4 |
| $\text{RoB}_{\text{base}}$ (LoRA) | 0.3M | $87.5_{\pm.3}$ | **95.1**$_{\pm.2}$ | $89.7_{\pm.7}$ | $63.4_{\pm1.2}$ | **93.3**$_{\pm.3}$ | $90.8_{\pm.1}$ | **86.6**$_{\pm.7}$ | **91.5**$_{\pm.2}$ | **87.2** |
| $\text{RoB}_{\text{large}}$ (FT)* | 355.0M | 90.2 | **96.4** | **90.9** | 68.0 | 94.7 | **92.2** | 86.6 | 92.4 | 88.9 |
| $\text{RoB}_{\text{large}}$ (LoRA) | 0.8M | **90.6**$_{\pm.2}$ | $96.2_{\pm.5}$ | **90.9**$_{\pm1.2}$ | **68.2**$_{\pm1.9}$ | **94.9**$_{\pm.3}$ | $91.6_{\pm.1}$ | **87.4**$_{\pm2.5}$ | **92.6**$_{\pm.2}$ | **89.0** |
| $\text{RoB}_{\text{large}}$ ($\text{Adpt}^{\text{P}}$)† | 3.0M | $90.2_{\pm.3}$ | $96.1_{\pm.3}$ | $90.2_{\pm.7}$ | **68.3**$_{\pm1.0}$ | **94.8**$_{\pm.2}$ | **91.9**$_{\pm.1}$ | $83.8_{\pm2.9}$ | $92.1_{\pm.7}$ | 88.4 |
| $\text{RoB}_{\text{large}}$ ($\text{Adpt}^{\text{P}}$)† | 0.8M | **90.5**$_{\pm.3}$ | **96.6**$_{\pm.2}$ | $89.7_{\pm1.2}$ | $67.8_{\pm2.5}$ | **94.8**$_{\pm.3}$ | $91.7_{\pm.2}$ | $80.1_{\pm2.9}$ | $91.9_{\pm.4}$ | 87.9 |
| $\text{RoB}_{\text{large}}$ ($\text{Adpt}^{\text{H}}$)† | 6.0M | $89.9_{\pm.5}$ | $96.2_{\pm.3}$ | $88.7_{\pm2.9}$ | $66.5_{\pm4.4}$ | $94.7_{\pm.2}$ | $92.1_{\pm.1}$ | $83.4_{\pm1.1}$ | $91.0_{\pm1.7}$ | 87.8 |
| $\text{RoB}_{\text{large}}$ ($\text{Adpt}^{\text{H}}$)† | 0.8M | $90.3_{\pm.3}$ | $96.3_{\pm.5}$ | $87.7_{\pm1.7}$ | $66.3_{\pm2.0}$ | $94.7_{\pm.2}$ | $91.5_{\pm.1}$ | $72.9_{\pm2.9}$ | $91.5_{\pm.5}$ | 86.4 |
| $\text{RoB}_{\text{large}}$ (LoRA)† | 0.8M | **90.6**$_{\pm.2}$ | $96.2_{\pm.5}$ | **90.2**$_{\pm1.0}$ | $68.2_{\pm1.9}$ | **94.8**$_{\pm.3}$ | $91.6_{\pm.2}$ | **85.2**$_{\pm1.1}$ | **92.3**$_{\pm.5}$ | **88.6** |
| $\text{DeB}_{\text{XXL}}$ (FT)* | 1500.0M | 91.8 | **97.2** | 92.0 | 72.0 | **96.0** | 92.7 | 93.9 | 92.9 | 91.1 |
| $\text{DeB}_{\text{XXL}}$ (LoRA) | 4.7M | **91.9**$_{\pm.2}$ | $96.9_{\pm.2}$ | **92.6**$_{\pm.6}$ | **72.4**$_{\pm1.1}$ | **96.0**$_{\pm.1}$ | **92.9**$_{\pm.1}$ | **94.9**$_{\pm.4}$ | **93.0**$_{\pm.2}$ | **91.3** |

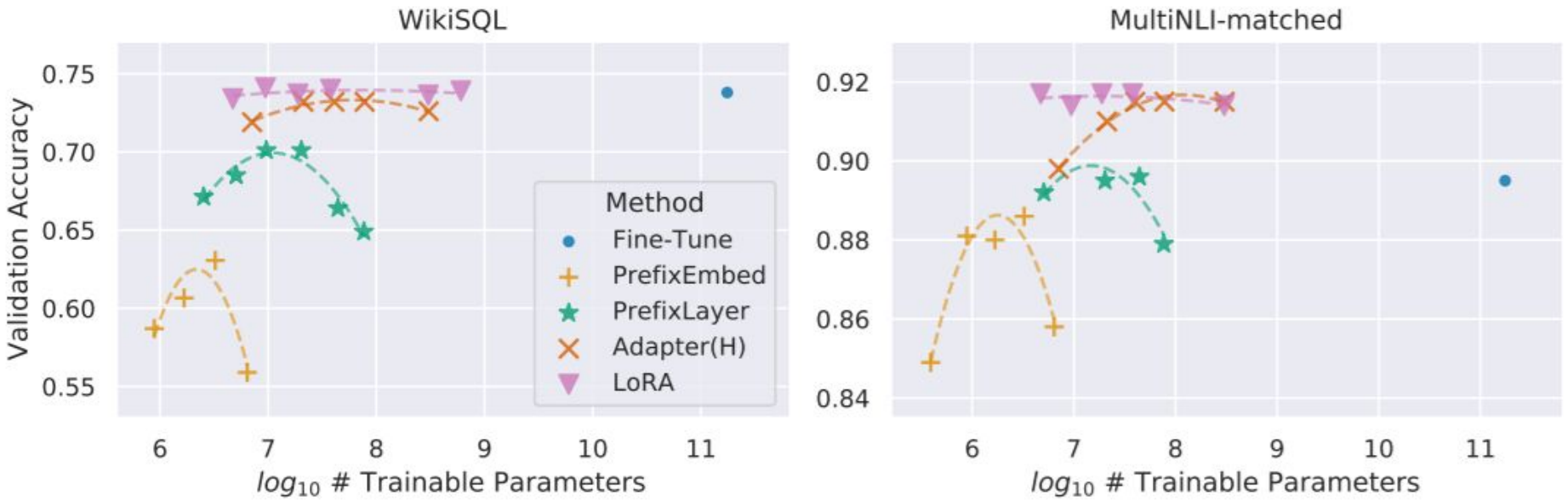
Table 2: $\text{RoBERTa}_{\text{base}}$, $\text{RoBERTa}_{\text{large}}$, and $\text{DeBERTa}_{\text{XXL}}$ with different adaptation methods on the GLUE benchmark. We report the overall (matched and mismatched) accuracy for MNLI, Matthew's correlation for CoLA, Pearson correlation for STS-B, and accuracy for other tasks. Higher is better for all metrics. * indicates numbers published in prior works. † indicates runs configured in a setup similar to Houlsby et al. (2019) for a fair comparison.

# Бейзлайны: GPT-2

| Model & Method | # Trainable Parameters | E2E NLG Challenge | | | | |
|---|---|---|---|---|---|---|
| | | BLEU | NIST | MET | ROUGE-L | CIDEr |
| GPT-2 M (FT)* | 354.92M | 68.2 | 8.62 | 46.2 | 71.0 | 2.47 |
| GPT-2 M (Adapter$^{L}$)* | 0.37M | 66.3 | 8.41 | 45.0 | 69.8 | 2.40 |
| GPT-2 M (Adapter$^{L}$)* | 11.09M | 68.9 | 8.71 | 46.1 | 71.3 | 2.47 |
| GPT-2 M (Adapter$^{H}$) | 11.09M | $67.3_{\pm.6}$ | $8.50_{\pm.07}$ | $46.0_{\pm.2}$ | $70.7_{\pm.2}$ | $2.44_{\pm.01}$ |
| GPT-2 M (FT$^{Top2}$)* | 25.19M | 68.1 | 8.59 | 46.0 | 70.8 | 2.41 |
| GPT-2 M (PreLayer)* | 0.35M | 69.7 | 8.81 | 46.1 | 71.4 | 2.49 |
| GPT-2 M (LoRA) | 0.35M | $\mathbf{70.4}_{\pm.1}$ | $\mathbf{8.85}_{\pm.02}$ | $\mathbf{46.8}_{\pm.2}$ | $\mathbf{71.8}_{\pm.1}$ | $\mathbf{2.53}_{\pm.02}$ |
| GPT-2 L (FT)* | 774.03M | 68.5 | 8.78 | 46.0 | 69.9 | 2.45 |
| GPT-2 L (Adapter$^{L}$) | 0.88M | $69.1_{\pm.1}$ | $8.68_{\pm.03}$ | $46.3_{\pm.0}$ | $71.4_{\pm.2}$ | $\mathbf{2.49}_{\pm.0}$ |
| GPT-2 L (Adapter$^{L}$) | 23.00M | $68.9_{\pm.3}$ | $8.70_{\pm.04}$ | $46.1_{\pm.1}$ | $71.3_{\pm.2}$ | $2.45_{\pm.02}$ |
| GPT-2 L (PreLayer)* | 0.77M | 70.3 | 8.85 | 46.2 | 71.7 | 2.47 |
| GPT-2 L (LoRA) | 0.77M | $\mathbf{70.4}_{\pm.1}$ | $\mathbf{8.89}_{\pm.02}$ | $\mathbf{46.8}_{\pm.2}$ | $\mathbf{72.0}_{\pm.2}$ | $2.47_{\pm.02}$ |

Table 3: GPT-2 medium (M) and large (L) with different adaptation methods on the E2E NLG Challenge. For all metrics, higher is better. LoRA outperforms several baselines with comparable or fewer trainable parameters. Confidence intervals are shown for experiments we ran. * indicates numbers published in prior works.

# Бейзлайны: GPT-3



# Авторы

**Edward Hu**

AI Researcher at Microsoft working on GPT-3 and infinite-width neural networks.

**Yelong Shen**

Kent State University

Verified email at cs.kent.edu

Data Mining  Machine Learning

# Теоретическая основа: Li et al. 2018, Aghajanyan et al. 2020

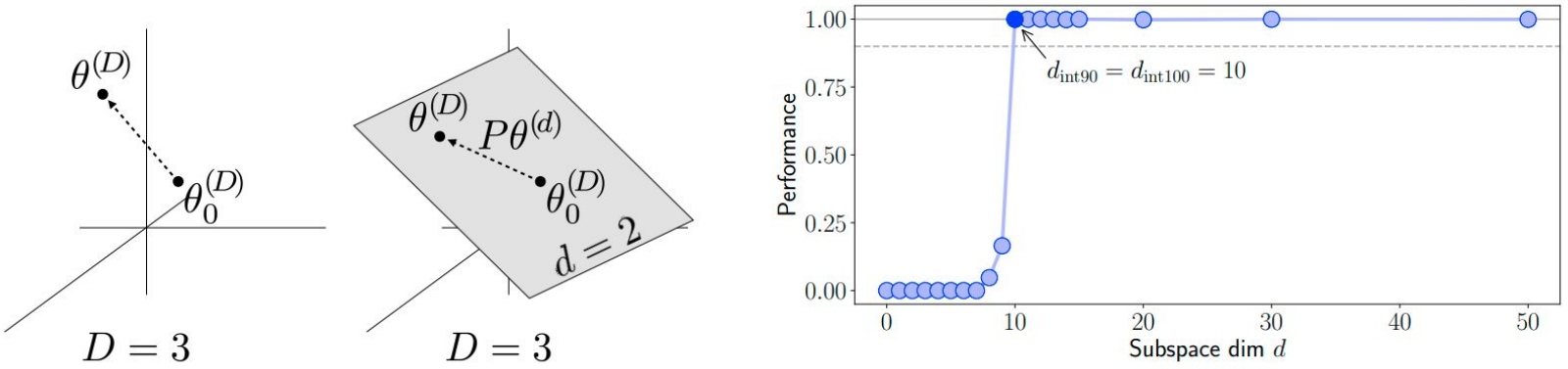
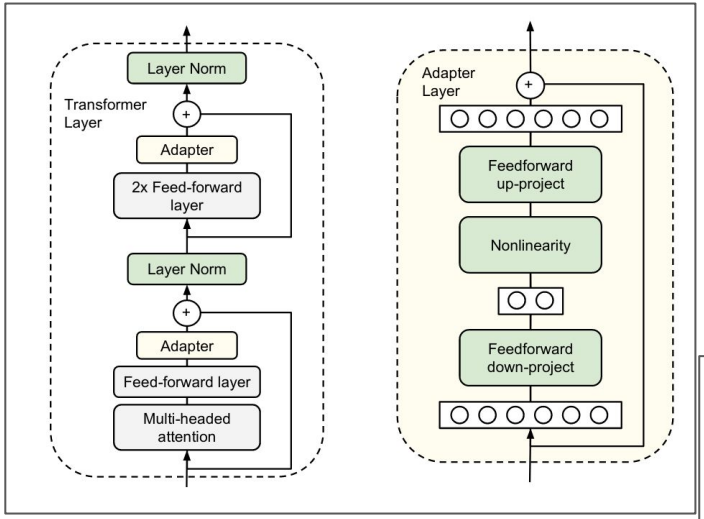



Figure 1: **(left)** Illustration of parameter vectors for direct optimization in the $D = 3$ case. **(middle)** Illustration of parameter vectors and a possible random subspace for the $D = 3, d = 2$ case. **(right)** Plot of performance *vs.* subspace dimension for the toy example of toy example of Sec. 2. The problem becomes both 90% solvable and 100% solvable at random subspace dimension 10, so $d_{\text{int90}}$ and $d_{\text{int100}}$ are 10.

# Сравнение с другими методами



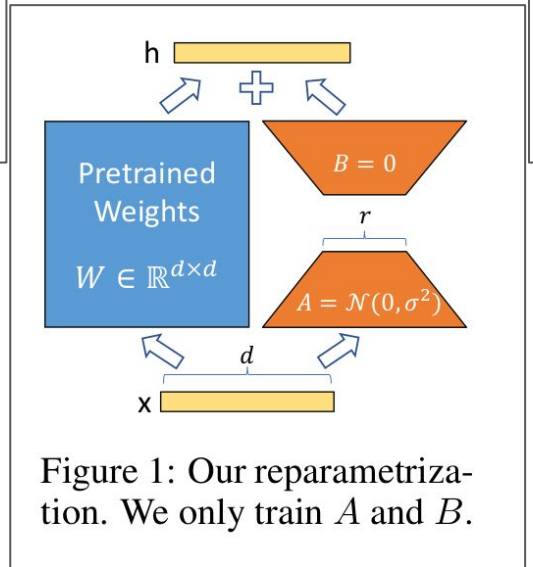
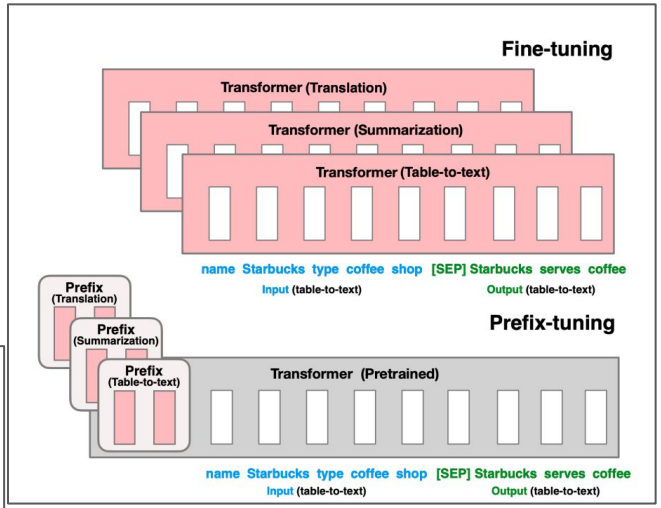




Figure 1: Our reparametrization. We only train $A$ and $B$.



# Смежные работы: Compacter (Mahabadi, Henderson, Ruder 2021)

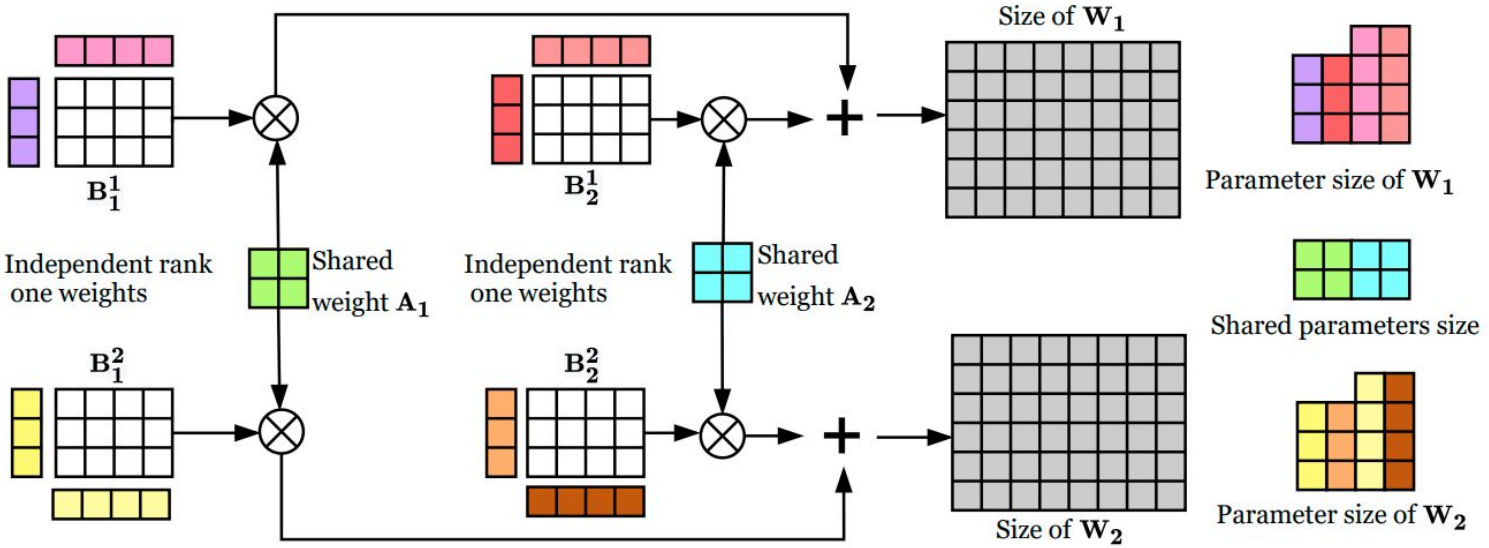



Figure 3: Illustration of generating weights of two different COMPACTER layers: $\boldsymbol{W_1} \in \mathbb{R}^{d \times k}$ (first row) and $\boldsymbol{W_2} \in \mathbb{R}^{d \times k}$ (second row). We generate $\boldsymbol{W_1}$ and $\boldsymbol{W_2}$ using $\boldsymbol{W_j} = \sum_{i=1}^{n} \boldsymbol{A_i} \otimes \boldsymbol{B_i}^{\boldsymbol{j}} = \sum_{i=1}^{n} \boldsymbol{A_i} \otimes (\boldsymbol{s_i}^{\boldsymbol{j}} \boldsymbol{t_i}^{\boldsymbol{j}\top})$ (5), by computing the sum of Kronecker products of *shared* matrices $\boldsymbol{A_i}$ and *adapter-specific* matrices $\boldsymbol{B_i^j}$, with $i \in \{1,\ldots,n\}$ and adapter index $j \in \{1,2\}$. We generate each $\boldsymbol{B_i^j}$ by multiplying independent rank one weights. In this example $n=2$, $d=6$, and $k=8$.

# В каком ключе цитируется

- Комбинация различных PELT-методов (Parameter Eficient Language Model Tuning) - Mao et al., 2021, Facebook Research + Univesity of Illinois
- Улучшение моделей, учитывающих приватность данных при обучении - Li et al., 2021 (Stanford University, Google Research)
- Fine-tuning полусиамских сетей для ранжирования текста - Jung et al., 2021 (Seoul University)