北京大学
PEKING UNIVERSITY

Training Networks in Null Space of Feature Covariance for Continual Learning *

Shipeng Wang[1], Xiaorong Li[1], Jian Sun(✉)[1,2,3], Zongben Xu[1,2,3]
[1] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, 710049, China
[2] National Engineering Laboratory of Big Data Algorithms and Analysis Technology, Xi'an, 710049, China
[3] Pazhou Lab, Guangzhou, Guangdong, 510335, China
{wangshipeng8128, lixiaorong}@stu.xjtu.edu.cn, {jiansun, zbxu}@xjtu.edu.cn

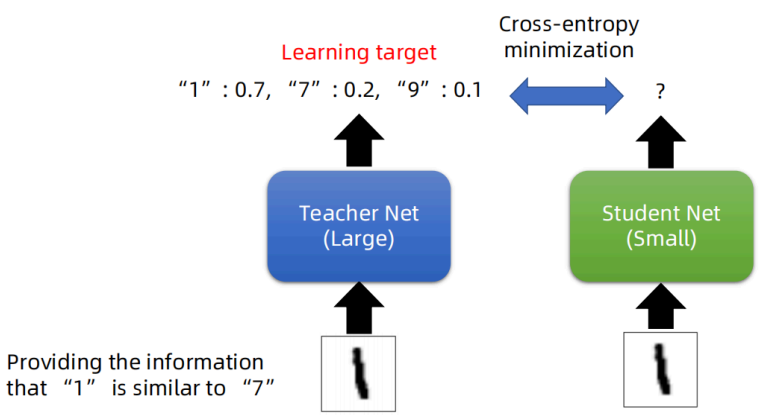
CVPR 2021 oral

2025/05/23

- Regularizetion
  l2、wise-FT、EWC（2017）...

$$\mathcal{L}_{EWC} = \sum_i \mathcal{F}_{\theta_i^{t-1}} \cdot \left(\theta_i^t - \theta_i^{t-1}\right)^2 ,$$

- Knowledge Distillation（2015）
  LwF（2017）...



- Data Replay

- Architecture
  Adapter、LoRA...

- Algorithm
  GEM（2017）、Adam-NSCL（本文）...

$$\ell(f_\theta, \mathcal{M}_k) = \frac{1}{|\mathcal{M}_k|} \sum_{(x_i,k,y_i)\in\mathcal{M}_k} \ell(f_\theta(x_i,k), y_i).$$

$$\begin{aligned} \text{minimize}_\theta \quad & \ell(f_\theta(x,t), y) \\ \text{subject to} \quad & \ell(f_\theta, \mathcal{M}_k) \le \ell(f_\theta^{t-1}, \mathcal{M}_k) \text{ for all } k < t, \end{aligned}$$

对于网络的某一线性层：

$$O^l_{p,t} = X^l_{p,t}\tilde{w}^l_t, \quad X^{l+1}_{p,t} = \sigma_l(O^l_{p,t})$$

$X^l_{p,t}$ 任务p的数据，第 l 层的输入

$O^l_{p,t}$ 第 l 层的输出

t 表示网络参数来自task t训练后

参数更新： $w_{t+1} = w_t + \Delta w$

如果能保证： $X^l_{p,t}\Delta w = 0$

那么： $X^l_{p,t}(w_t + \Delta w) = X^l_{p,t}w_t = O^l_{p,t}$

continual learning 目标：$X^l_{p,t} = X^l_{p,t-1}$ *and* $f(X_p, \tilde{\mathbf{w}}_{t-1}) = f(X_p, \tilde{\mathbf{w}}_t)$.

找一个投影矩阵P，P将梯度G投影到X的右零空间中，使得XPG恒为0，用投影后的梯度PG更新网络参数

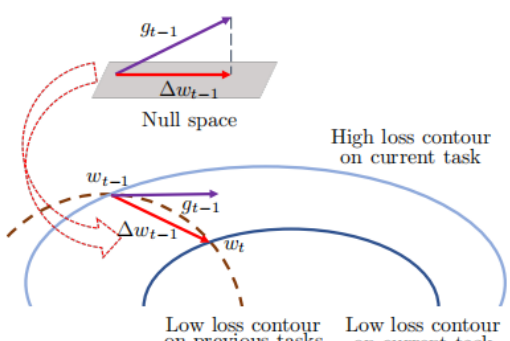


Figure 1. To avoid forgetting, we train network in the layer-wise null space of the corresponding uncentered covariance of all input features of previous tasks.

X形状为(n,d)，n通常很大
X和X^TX有相同的右零空间，（Ax=0和$A^TAx=0$的解x相同）

$$\mathcal{X}_{t-1}^l = \frac{1}{n_{t-1}}(X_{t-1}^l)^\top X_{t-1}^l$$

，形状为(d,d)

$$\bar{\mathcal{X}}_{t-1}^l = \frac{\bar{n}_{t-2}}{\bar{n}_{t-1}}\bar{\mathcal{X}}_{t-2}^l + \frac{n_{t-1}}{\bar{n}_{t-1}}\mathcal{X}_{t-1}^l,$$

By applying SVD to $\bar{\mathcal{X}}_{t-1}^l$, we have

$$U^l, \Lambda^l, (U^l)^\top = \text{SVD}(\bar{\mathcal{X}}_{t-1}^l), \tag{5}$$

where $U^l = [U_1^l, U_2^l]$ and $\Lambda^l = \begin{bmatrix} \Lambda_1^l & 0 \\ 0 & \Lambda_2^l \end{bmatrix}$. If all singular values of zero are in Λ_2^l, *i.e.*, $\Lambda_2^l = 0$, then $\bar{\mathcal{X}}_{t-1}^l U_2^l = U_1^l \Lambda_1^l (U_1^l)^\top U_2^l = 0$ holds, since U^l is an unitary matrix. It suggests that the range space of U_2^l is the null space of $\bar{\mathcal{X}}_{t-1}^l$. Thus we can get the parameter update $\Delta w_{t,s}^l$ lying in the null space of $\bar{\mathcal{X}}_{t-1}^l$ by

$$\Delta w_{t,s}^l = U_2^l (U_2^l)^\top g_{t,s}^l \tag{6}$$

with $U_2^l (U_2^l)^\top$ as projection operator

对于每一层的输入 $X_{p,t}^l$ ，都有一个投影矩阵P

实际上，零奇异值不一定存在：

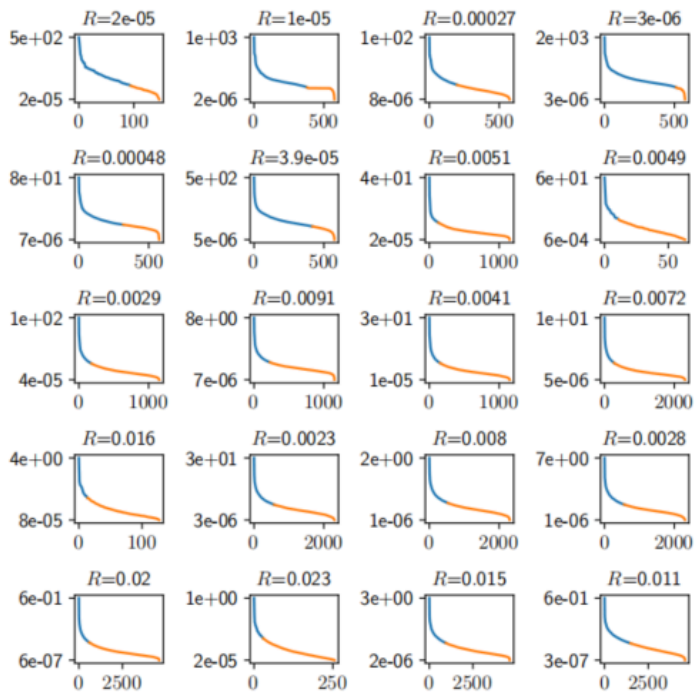
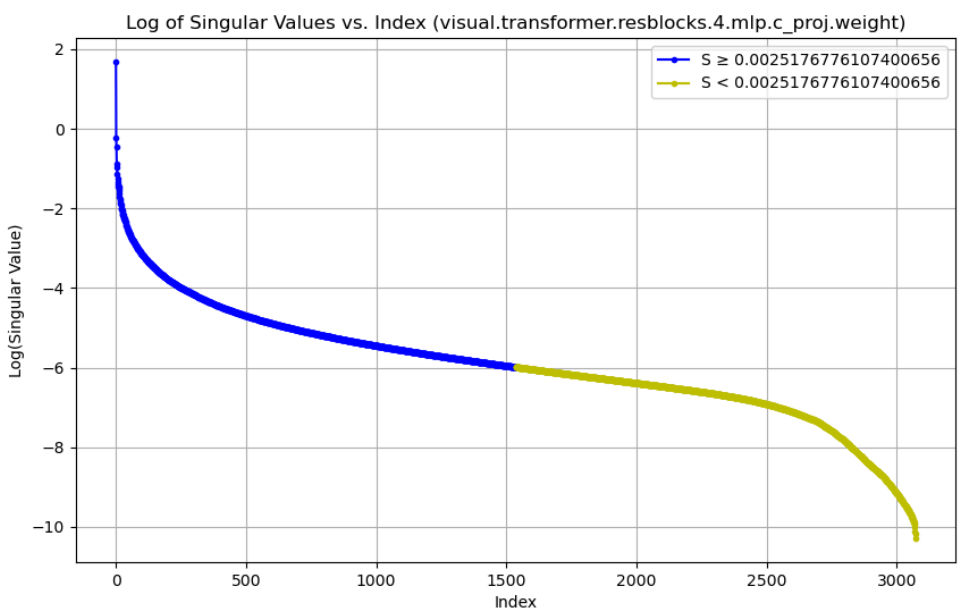


Figure 3. Singular values of uncentered covariance matrix at different layers of pretrained ResNet-18 on ImageNet ILSVRC 2012. Orange curves denote the singular values smaller than $50\lambda^l_{\min}$.

取近似：

Λ_2^l. We adaptively select Λ_2^l with diagonal singular values $\lambda \in \{\lambda | \lambda \leq a\lambda^l_{\min}\}$ $(a > 0)$, where $\lambda^l_{\min}$ is the smallest singular value. Furthermore, to empirically verify the ratio-

以中位数为阈值：



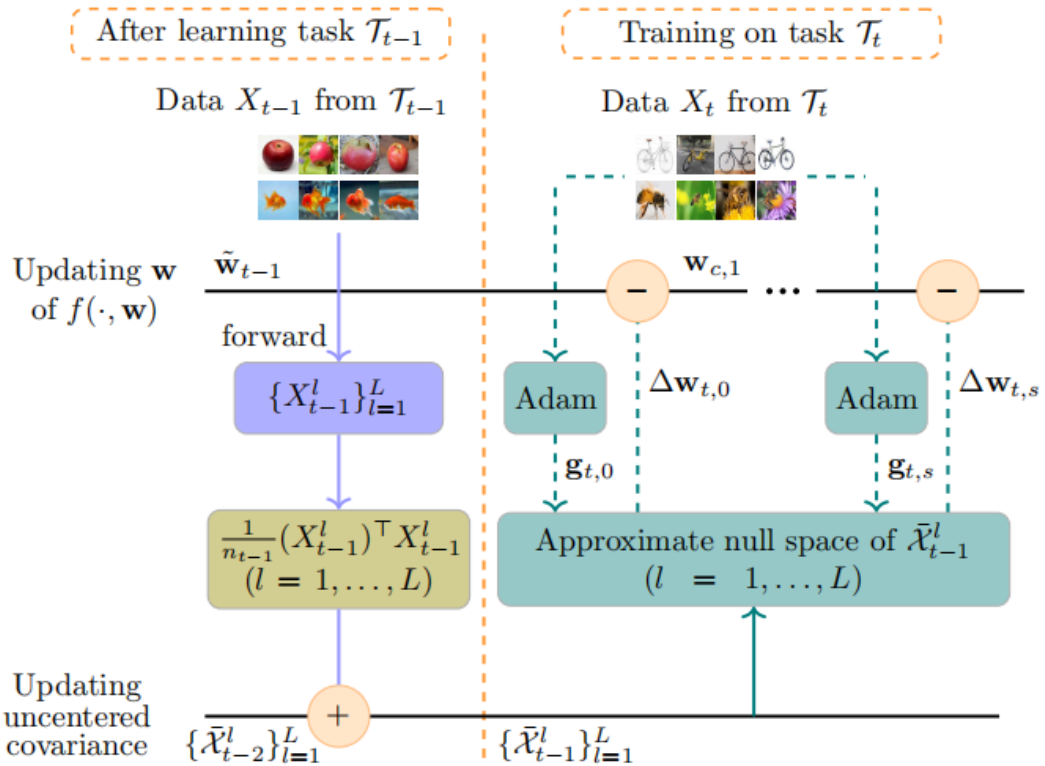


Figure 2. The pipeline of our algorithm.

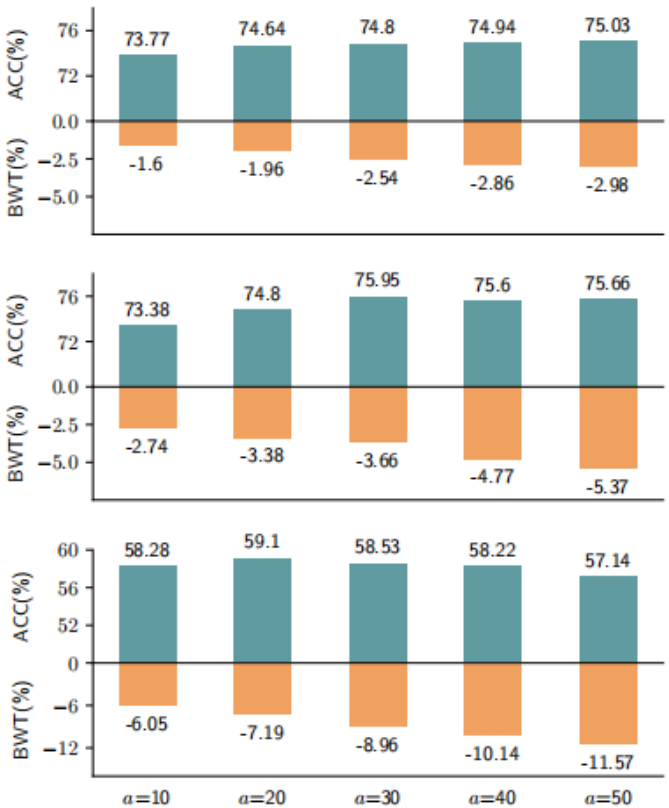


Figure 5. Stability and plasticity analysis. Top: 10-split-CIFAR-100. Middle: 20-split-CIFAR-100. Bottom: 25-split-TinyImageNet.

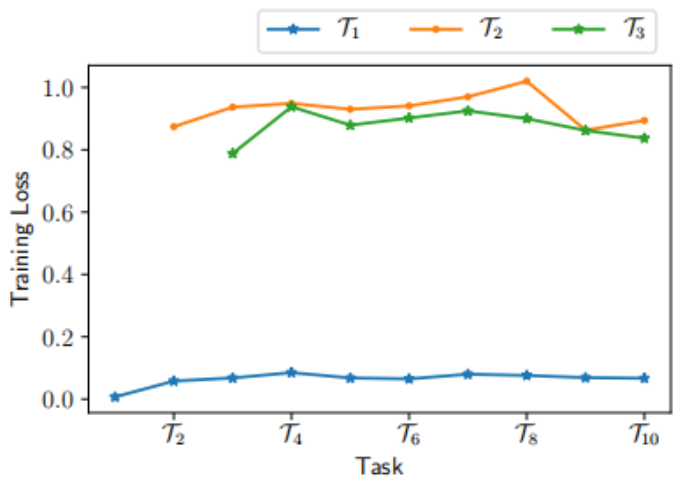


Figure 6. The curves of training losses of network on tasks $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$ when the network is trained on sequential tasks.

北京大学
PEKING UNIVERSITY

ALPHAEDIT: NULL-SPACE CONSTRAINED KNOWLEDGE EDITING FOR LANGUAGE MODELS

Junfeng Fang[1]*, **Houcheng Jiang**[2]*, **Kun Wang**[2], **Yunshan Ma**[1],
Jie Shi[1], **Xiang Wang**[2], **Xiangnan He**[2]†, **Tat-Seng Chua**[1]
[1]National University of Singapore, [2]University of Science and Technology of China
fangjf1997@gmail.com, jianghc@mail.ustc.edu.cn

ICLR 2025 Outstanding Paper

2025/05/23

An autoregressive large language model (LLM) predicts the next token $\boldsymbol{x}$ in a sequence based on the preceding tokens. Specifically, the hidden state of $\boldsymbol{x}$ at layer l within the model, denoted as $\boldsymbol{h}^l$, can be calculated as:

$$\boldsymbol{h}^l = \boldsymbol{h}^{l-1} + \boldsymbol{a}^l + \boldsymbol{m}^l, \quad \boldsymbol{m}^l = \boldsymbol{W}_{\text{out}}^l \sigma(\boldsymbol{W}_{\text{in}}^l \gamma(\boldsymbol{h}^{l-1} + \boldsymbol{a}^l)), \tag{1}$$

where $\boldsymbol{a}^l$ and $\boldsymbol{m}^l$ represent the outputs of the attention block and the feed-forward network (FFN) layer, respectively; $\boldsymbol{W}_{\text{in}}^l$ and $\boldsymbol{W}_{\text{out}}^l$ are the weight matrices of the FFN layers; σ is the non-linear activation function, and γ denotes the layer normalization. Following Meng et al. (2022), we express the attention and FFN modules in parallel here.

It is worth noting that $\boldsymbol{W}_{\text{out}}^l$ within FFN layers is often interpreted as a linear associative memory, functioning as key-value storage for information retrieval (Geva et al., 2021). Specifically, if the knowledge stored in LLMs is formalized as (s, r, o) — representing subject s, relation r, and object o (*e.g.*, s = "The latest Olympic Game", r = "was held in", o = "Paris") — $\boldsymbol{W}_{\text{out}}^l$ associates a set of input keys $\boldsymbol{k}$ encoding (s, r) with corresponding values $\boldsymbol{v}$ encoding (o). That is,

$$\underbrace{\boldsymbol{m}^l}_{\boldsymbol{v}} = \boldsymbol{W}_{\text{out}}^l \underbrace{\sigma(\boldsymbol{W}_{\text{in}}^l \gamma(\boldsymbol{h}^{l-1} + \boldsymbol{a}^l))}_{\boldsymbol{k}}. \tag{2}$$

This interpretation has inspired most model editing methods to modify the FFN layers for knowledge updates (Hase et al., 2023; Li et al., 2024a; Hu et al., 2024). For simplicity, we use $\boldsymbol{W}$ to refer to $\boldsymbol{W}_{\text{out}}^l$ in the following sections.

需要更新的知识：

$$\boldsymbol{K}_1 = [\boldsymbol{k}_1 | \boldsymbol{k}_2 | \dots | \boldsymbol{k}_u] \in \mathbb{R}^{d_0 \times u}, \quad \boldsymbol{V}_1 = [\boldsymbol{v}_1 | \boldsymbol{v}_2 | \dots | \boldsymbol{v}_u] \in \mathbb{R}^{d_1 \times u},$$

优化目标：$$\boldsymbol{\Delta} = \arg\min_{\tilde{\boldsymbol{\Delta}}} (\left\| (\boldsymbol{W} + \tilde{\boldsymbol{\Delta}}) \boldsymbol{K}_1 - \boldsymbol{V}_1 \right\|^2 + \left\| (\boldsymbol{W} + \tilde{\boldsymbol{\Delta}}) \boldsymbol{K}_0 - \boldsymbol{V}_0 \right\|^2). \tag{5}$$

其中 $\boldsymbol{W}\boldsymbol{K}_0 = \boldsymbol{V}_0$ 表示模型的保留知识

闭式解：$$\boldsymbol{\Delta} = (\boldsymbol{V}_1 - \boldsymbol{W}\boldsymbol{K}_1) \boldsymbol{K}_1^T \left(\boldsymbol{K}_0 \boldsymbol{K}_0^T + \boldsymbol{K}_1 \boldsymbol{K}_1^T \right)^{-1}. \tag{6}$$

Although $\boldsymbol{K}_0$ is difficult to obtain directly since we hardly have access to the LLM's full extent of knowledge, it can be estimated using abundant text input (Meng et al., 2023). In practical applications, $100,000$ (s, r, o) triplets from Wikipedia are typically randomly selected to encode $\boldsymbol{K}_0$ (Meng et al., 2023), making $\boldsymbol{K}_0$ a high-dimensional matrix with $100,000$ columns (*i.e.*, $\boldsymbol{K}_0 \in \mathbb{R}^{d_0 \times 100,000}$). See Appendix B.1 for detailed implementation steps.

闭式解：$\boldsymbol{\Delta} = (\boldsymbol{V}_1 - \boldsymbol{W}\boldsymbol{K}_1)\,\boldsymbol{K}_1^T \left(\boldsymbol{K}_0\boldsymbol{K}_0^T + \boldsymbol{K}_1\boldsymbol{K}_1^T\right)^{-1}.$ (6)

如果 $(\boldsymbol{W} + \boldsymbol{\Delta}')\boldsymbol{K}_0 = \boldsymbol{W}\boldsymbol{K}_0 = \boldsymbol{V}_0.$

那么此次编辑不会影响保留知识——将Δ投影到K_0的零空间中

Following the existing methods for conducting null space projection (Wang et al., 2021), we first apply a Singular Value Decomposition (SVD) to $\boldsymbol{K}_0(\boldsymbol{K}_0)^T$:

$$\left\{\boldsymbol{U}, \boldsymbol{\Lambda}, (\boldsymbol{U})^T\right\} = \text{SVD}\left(\boldsymbol{K}_0(\boldsymbol{K}_0)^T\right), \tag{8}$$

where each column in $\boldsymbol{U}$ is an eigenvector of $\boldsymbol{K}_0(\boldsymbol{K}_0)^T$. Then, we remove the eigenvectors in $\boldsymbol{U}$ that correspond to non-zero eigenvalues[1], and define the remaining submatrix as $\hat{\boldsymbol{U}}$. Based on this, the projection matrix $\boldsymbol{P}$ can be defined as follows:

$$\boldsymbol{P} = \hat{\boldsymbol{U}}(\hat{\boldsymbol{U}})^T. \tag{9}$$

This projection matrix can map the column vectors of $\boldsymbol{\Delta}$ into the null space of $\boldsymbol{K}_0(\boldsymbol{K}_0)^T$, as it satisfies the condition $\boldsymbol{\Delta}\boldsymbol{P} \cdot \boldsymbol{K}_0(\boldsymbol{K}_0)^T = \boldsymbol{0}$. The detailed derivation is exhibited in Appendix B.3.

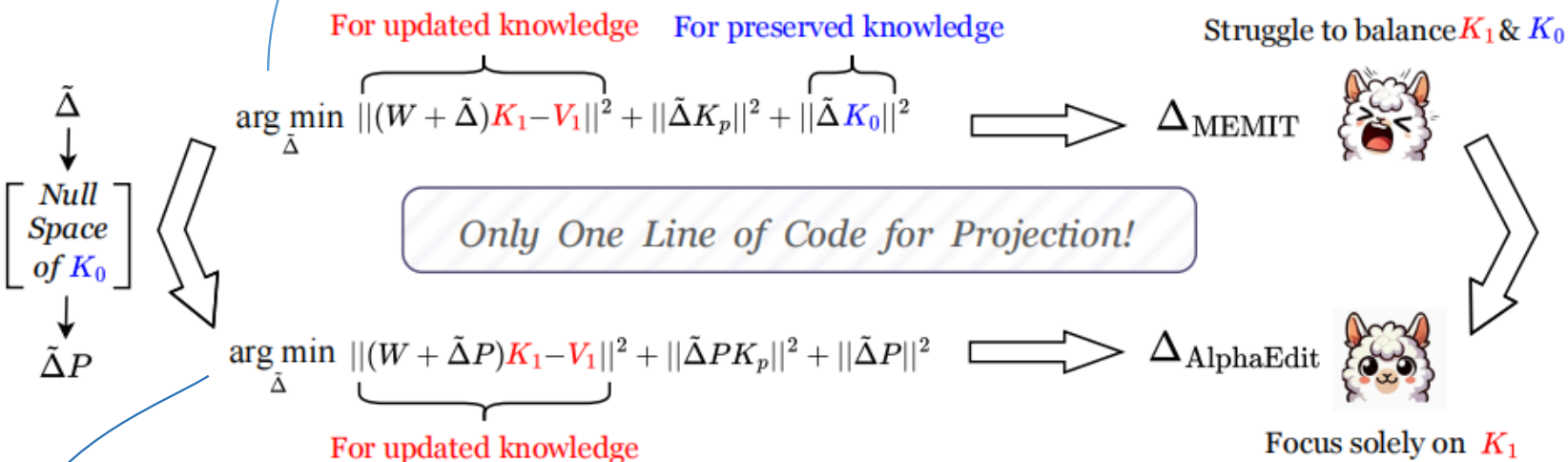
Since $\boldsymbol{K}_0$ and $\boldsymbol{K}_0(\boldsymbol{K}_0)^T$ share the same null space, we can derive $\boldsymbol{\Delta}\boldsymbol{P} \cdot \boldsymbol{K}_0 = \boldsymbol{0}$. Hence, we have:

$$(\boldsymbol{W} + \boldsymbol{\Delta}\boldsymbol{P})\boldsymbol{K}_0 = \boldsymbol{W}\boldsymbol{K}_0 = \boldsymbol{V}_0. \tag{10}$$

This shows the projection matrix $\boldsymbol{P}$ ensures that the model edits occur without interference with the preserved knowledge in LLMs.

```
"nullspace_threshold":2e-2,
```

$$\Delta_{\text{MEMIT}} = RK_1^T(K_pK_p^T + K_1K_1^T + K_0K_0^T)^{-1}. \quad (15)$$



$$\Delta = \arg\min_{\tilde{\Delta}} \left(||(W + \tilde{\Delta}P)K_1 - V_1||^2 + ||\tilde{\Delta}P||^2 + ||\tilde{\Delta}PK_p||^2 \right). \quad (12)$$

To facilitate expression, we define the residual vector of the current edit as $R = V_1 - WK_1$. Based on this, Eqn. 12 can be solved using the normal equation (Lang, 2012):

$$(\Delta PK_1 - R)K_1^T P + \Delta P + \Delta PK_pK_p^T P = 0. \quad (13)$$

Solving Eqn. 13 yields the final perturbation $\Delta_{\text{AlphaEdit}} = \Delta P$ which will be added to the model parameters W:

$$\Delta_{\text{AlphaEdit}} = RK_1^T P(K_pK_p^T P + K_1K_1^T P + I)^{-1}. \quad (14)$$

总共2000个样本，
batch size为100.

结果：好

Table 1: Comparison of AlphaEdit with existing methods on the sequential model editing task. *Eff.*, *Gen.*, *Spe.*, *Flu.* and *Consis.* denote Efficacy, Generalization, Specificity, Fluency and Consistency, respectively. The best results are highlighted in bold, while the second-best results are underlined.

Method	Model	Counterfact					ZsRE		
		Eff.↑	Gen.↑	Spe.↑	Flu.↑	Consis.↑	Eff.↑	Gen.↑	Spe.↑
Pre-edited		7.85±0.26	10.58±0.26	89.48±0.18	635.23±0.11	24.14±0.08	36.99±0.30	36.34±0.30	31.89±0.22
FT	LLaMA3	83.33±0.37	67.79±0.40	46.63±0.37	233.72±0.22	8.77±0.05	30.48±0.26	30.22±0.32	15.49±0.17
MEND	LLaMA3	63.24±0.31	61.17±0.36	45.37±0.38	372.16±0.80	4.21±0.05	0.91±0.05	1.09±0.05	0.53±0.02
InstructEdit	LLaMA3	66.58±0.24	64.18±0.35	47.14±0.37	443.85±0.78	7.28±0.04	1.58±0.04	1.36±0.08	1.01±0.05
ROME	LLaMA3	64.40±0.41	61.42±0.42	49.44±0.38	449.06±0.26	3.31±0.02	2.01±0.07	1.80±0.07	0.69±0.03
MEMIT	LLaMA3	65.65±0.47	64.65±0.42	51.56±0.38	437.43±1.67	6.58±0.11	34.62±0.36	31.28±0.34	18.49±0.19
PRUNE	LLaMA3	68.25±0.46	64.75±0.41	49.82±0.36	418.03±1.52	5.90±0.10	24.77±0.27	23.87±0.27	20.69±0.23
RECT	LLaMA3	66.05±0.47	63.62±0.43	61.41±0.37	526.62±0.44	20.54±0.09	86.05±0.23	80.54±0.27	31.67±0.22
AlphaEdit	LLaMA3	**98.90±0.10**	**94.22±0.19**	**67.88±0.29**	**622.49±0.16**	**32.40±0.11**	**94.47±0.13**	**91.13±0.19**	**32.55±0.22**
Pre-edited		16.22±0.31	18.56±0.45	83.11±0.13	621.81±0.67	29.74±0.51	26.32±037	25.79±0.25	27.42±0.53
FT	GPT-J	92.15±0.27	72.38±0.38	43.35±0.37	297.92±0.77	6.65±0.10	72.37±0.29	68.91±0.32	19.66±0.23
MEND	GPT-J	46.15±0.50	46.22±0.51	53.90±0.48	242.41±0.41	3.94±0.03	0.71±0.04	0.71±0.04	0.52±0.03
InstructEdit	GPT-J	50.62±0.58	51.73±0.42	56.28±0.50	245.89±0.44	4.21±0.04	0.92±0.07	0.88±0.03	0.65±0.06
ROME	GPT-J	57.50±0.48	54.20±0.40	52.05±0.31	589.42±0.08	3.22±0.02	56.42±0.42	54.65±0.42	9.86±0.16
MEMIT	GPT-J	98.55±0.11	95.50±0.16	63.64±0.31	546.28±0.88	34.89±0.15	94.91±0.16	90.22±0.23	**30.39±0.27**
PRUNE	GPT-J	86.15±0.34	86.85±0.29	53.87±0.35	427.14±0.53	14.78±0.11	0.15±0.02	0.15±0.02	0.00±0.00
RECT	GPT-J	98.80±0.10	86.58±0.28	72.22±0.28	617.31±0.19	41.39±0.12	96.38±0.14	91.21±0.21	27.79±0.26
AlphaEdit	GPT-J	**99.75±0.08**	**96.38±0.23**	**75.48±0.21**	**618.50±0.17**	**42.08±0.15**	**99.79±0.14**	**96.00±0.22**	28.29±0.25
Pre-edited		22.23±0.73	24.34±0.62	78.53±0.33	626.64±0.31	31.88±0.20	22.19±0.24	31.30±0.27	24.15±0.32
FT	GPT2-XL	63.55±0.48	42.20±0.41	57.06±0.30	519.35±0.27	10.56±0.05	37.11±0.39	33.30±0.37	10.36±0.17
MEND	GPT2-XL	50.80±0.50	50.80±0.48	49.20±0.51	407.21±0.08	1.01±0.00	0.00±0.00	0.00±0.00	0.00±0.00
InstructEdit	GPT2-XL	55.32±0.58	53.63±0.42	53.25±0.62	412.57±0.15	1.08±0.03	3.54±0.03	4.25±0.02	3.23±0.04
ROME	GPT2-XL	54.60±0.48	51.18±0.40	52.68±0.33	366.13±1.40	0.72±0.02	47.50±0.43	43.56±0.42	14.27±0.19
MEMIT	GPT2-XL	94.70±0.22	85.82±0.28	60.50±0.32	477.26±0.54	22.72±0.15	79.17±0.32	71.44±0.36	**26.42±0.25**
PRUNE	GPT2-XL	82.05±0.38	78.55±0.34	53.02±0.35	530.47±0.39	15.93±0.11	21.62±0.30	19.27±0.28	13.19±0.18
RECT	GPT2-XL	92.15±0.26	81.15±0.33	65.13±0.31	480.83±0.62	21.05±0.16	81.02±0.31	73.08±0.35	24.85±0.25
AlphaEdit	GPT2-XL	**99.50±0.24**	**93.95±0.34**	**66.39±0.31**	**597.88±0.18**	**39.38±0.15**	**94.81±0.30**	**86.11±0.29**	25.88±0.21

连续编辑，性能基本不变

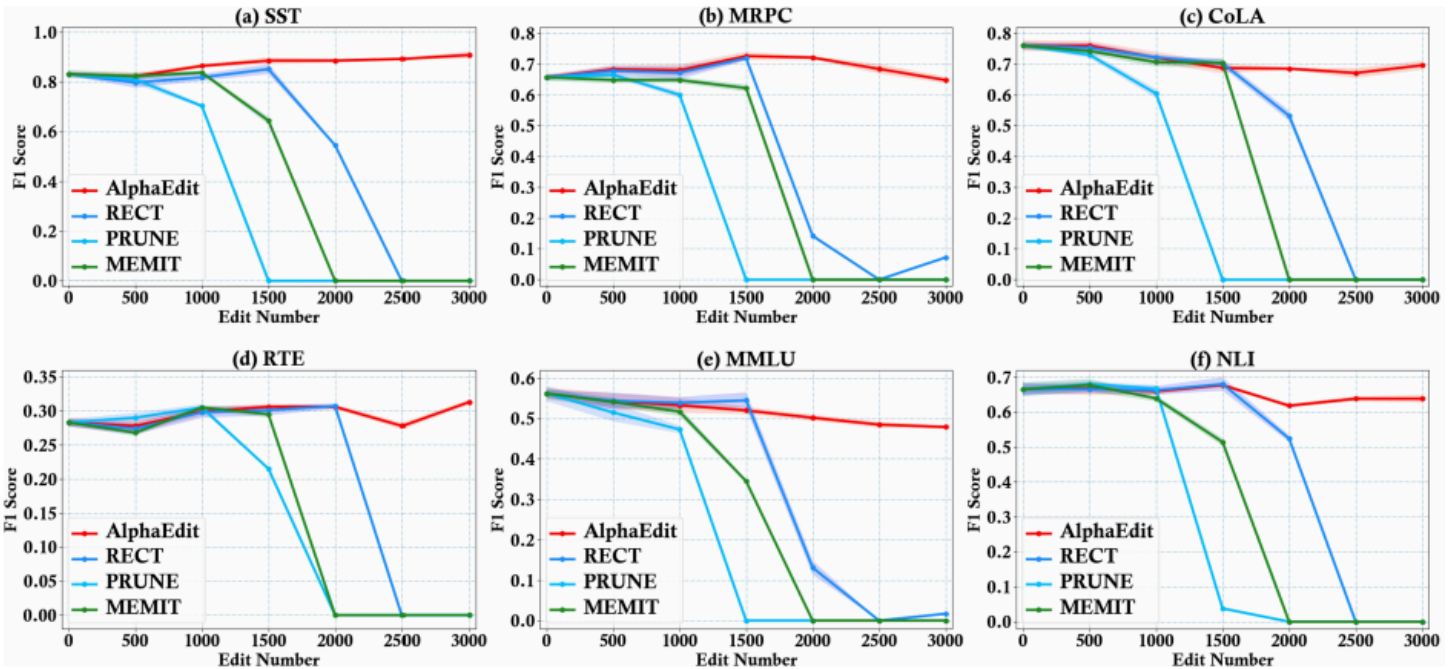


Figure 4: F1 scores of the post-edited LLaMA3 (8B) on six tasks (*i.e.*, SST, MRPC, CoLA, RTE, MMLU and NLI) used for general capability testing. Best viewed in color.

保留知识的表示
分布基本不变

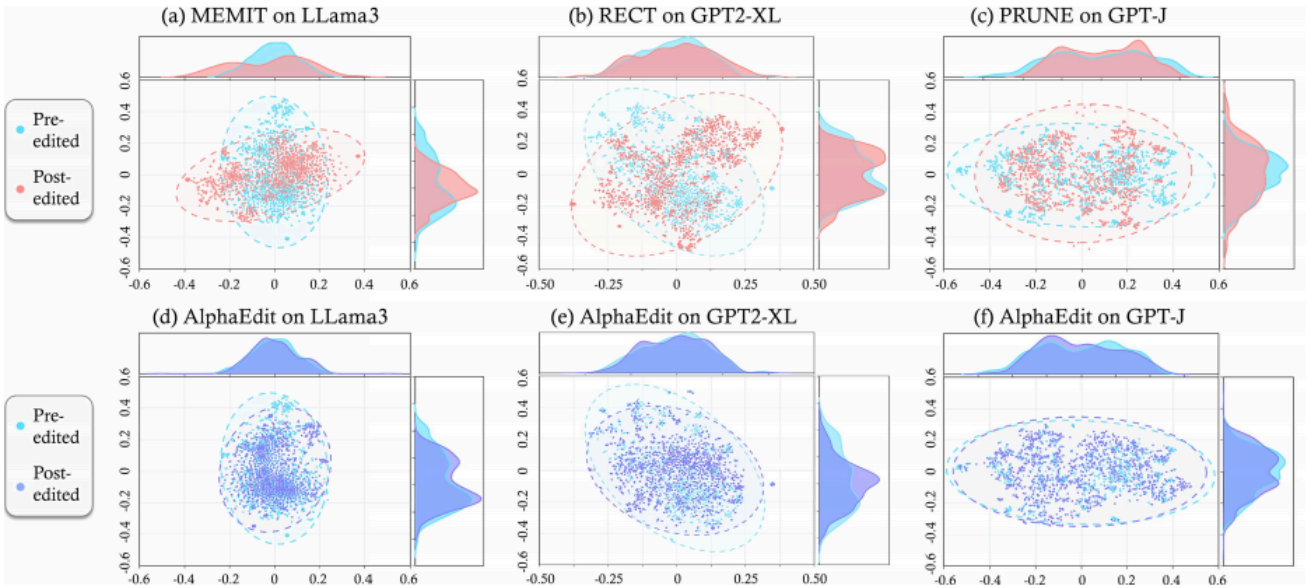


Figure 5: The distribution of hidden representations of pre-edited and post-edited LLMs after dimensionality reduction. The top and right curve graphs display the marginal distributions for two reduced dimensions, where AlphaEdit consistently exhibits minimal shift. Best viewed in color.