
Supplementary Materials for Hybrid Re-matching for Continual Learning with Parameter-efficient Tuning

Weicheng Wang¹, Guoli Jia³, Xialei Liu¹, Liang Lin^{2,4}, Jufeng Yang^{1,2,5*}

¹ VCIP & TMCC & DISec, College of Computer Science, Nankai University, Tianjin, China.

² Pengcheng Laboratory, Shenzhen, China.

³ Electronic Engineering Department, Tsinghua University, Beijing, China.

⁴ School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China.

⁵ Nankai International Advanced Research Institute (SHENZHEN-FUTIAN), Shenzhen, China.

2120230639@mail.nankai.edu.cn, exped1230@gmail.com

xialei@nankai.edu.cn, linliang@ieee.org, yangjufeng@nankai.edu.cn

1 Algorithms for Hybrid Re-Matching

The re-matching algorithms during inference for HRM-PET are illustrated in Algorithm 1. Moreover, notations in this paper are shown in Table 1.

1.1 Optimization of Hierarchical Components

In baseline, we disentangle the PET-based CL into three distinct optimization objectives following the state-of-the-art CL method, HiDe-Prompt [1]: *task-identity inference* (TII) optimizes the prediction of task-identity, *within-task prediction* (WTP) improves the performance of model within the current task and *task-adaptive prediction* (TAP) facilitates the adaptation of model across all tasks.

In TII, given the image $x \in D_t$ of current task t , the frozen pre-trained model h_{ptm} with parameters θ_{ptm} is directly used to extract its features as query $q(x) = h(x; \theta_{ptm})$. Then, the query feature is input to an auxiliary classifier g_ω parameterized by θ_ω for the prediction of task identity, which remains unfrozen throughout the training process, and employs cross-entropy to optimize TII:

$$\mathcal{L}_{\text{TII}}(\theta_\omega) = - \sum_{(x,y) \in D_t} \sum_i y_i \log(\hat{d}_i(x)), \quad (1)$$

where $\hat{d}(x) = g(q(x); \theta_\omega)$ represents the output probability distribution.

In WTP, a parameter pool $P = \{p_0, p_1, \dots, p_t\}$ is maintained. Considering the extremely small storage cost of lightweight p in PET, training a separate p and keeping it in P for each task can effectively protect task-specific knowledge. During the optimization of the current task t , parameters $\{p_0, p_1, \dots, p_{t-1}\}$ from the previous tasks are held fixed to prevent catastrophic forgetting. We get probability distribution $\tilde{d}(x) = g(h(x; p_t, \theta_{ptm}); \theta_g)$ by using the pre-trained model followed with current parameter p_t . We utilize fundamental loss cross-entropy to optimize WTP:

$$\mathcal{L}_{\text{WTP}}(p_t, \theta_g) = \mathcal{L}_{\text{CE}} = - \sum_{(x,y) \in D_t} \sum_i y_i \log(\tilde{d}_i(x)), \quad (2)$$

Following [1, 2], we improve TAP by adapting output layer g_ω and g_{θ_g} for all tasks with old-task statistics of feature in TII and WTP. To be specific, after tuning g_ω or g_{θ_g} by training data, the approximated distribution for each class label can be calculated and stored utilizing the features extracted by h_{θ_h} on corresponding training images, such as Gaussian distributions consisting of mean and variance. Then, we sample pseudo representations from the approximated distributions of all classes encountered as input features to optimize g_ω and g_{θ_g} again.

*Corresponding author.

Algorithm 1: Re-Matching of HRM-PET at test time

Given components: Pre-trained ViT h_{ptm} with θ_{ptm} , trained classification layer g_{θ_g} with θ_g , trained classification layer for task identity inference g_{ω} with ω , trained parameter pool $P = \{p_1, p_2, \dots, p_T\}$, confidence function $E(\cdot)$.

Input: test example x

- 1 Generate query feature $q(x) = h(x; \theta_{ptm})$
 - 2 Generate distribution for initial matching $\hat{d}(x) = g(q(x); \omega)$
 - 3 The initial matching for task identity $\hat{t}_f = \mathcal{T}(\underset{i}{\operatorname{argmax}} \hat{d}(x))$
 - 4 Generate initial predicted distribution $\tilde{d}(x) = g(h(x; p_{\hat{t}_f}, \theta_{ptm}); \theta_g)$
 - 5 Generate task identity from predicted distribution $\hat{t}_s = \mathcal{T}(\underset{i}{\operatorname{argmax}} \tilde{d}(x))$
 - 6 **if** $\hat{t}_f \neq \hat{t}_s$ **then**
 - 7 Generate predicted distribution and class after direct re-matching:
 - 8 $\tilde{d} = g(h(x; p_{\hat{t}_s}, \theta_{ptm}); \theta_g)$
 - 9 $\hat{y} = \underset{i}{\operatorname{argmax}} \tilde{d}(x)$ \triangleright **Direct re-matching**
 - 10 **else**
 - 11 $\hat{y} = \underset{i}{\operatorname{argmax}} \tilde{d}(x)$
 - 12 **if** $E(\tilde{d}(x)) \leq \tau$ **then**
 - 13 Generate the top-N identities Γ in initial matching $\Gamma = \underset{\{c_j\}_{j=1}^N}{\operatorname{argmax}} \hat{d}_{c_j}(x)$
 - 14 Generate task identity of the highest confidence:
 - 15 $\hat{t}_s = \underset{i \in \Gamma}{\operatorname{argmax}} E(g(h(x; p_{\mathcal{T}(i)}, \theta_{ptm}); \theta_g))$
 - 16 Generate final distribution and class after confidence-based re-matching:
 - 17 $\tilde{d} = g(h(x; p_{\hat{t}_s}, \theta_{ptm}); \theta_g)$
 - 18 $\hat{y} = \underset{i}{\operatorname{argmax}} \tilde{d}(x)$ \triangleright **Confidence-based re-matching**
 - 19 **else**
 - 20 $\hat{y} = \underset{i}{\operatorname{argmax}} \tilde{d}(x)$
-

1.2 Theory of re-matching improving the continual learning

The accuracy of task identity prediction has been shown to be critical for improving CL performance, as demonstrated in HiDe-Prompt. Assuming that in the class incremental learning (CIL) scenario, a total of T tasks are defined as $D = \{D_1, D_2, \dots, D_T\}$. In D_t of task t , \mathcal{X}_t and \mathcal{Y}_t are the domain and label of task t . Let $\mathcal{X}_t = \bigcup_j \mathcal{X}_{t,j}$ and $\mathcal{Y}_t = \{\mathcal{Y}_{t,j}\}$, where $j \in \{1, \dots, |\mathcal{Y}_t|\}$ indicates the j -th class in task t . Given a pre-trained model f_{θ} , CIL aims to learn $P(\mathbf{x} \in \mathcal{X}_{i,j} | D, \theta)$ for the sample $\mathbf{x} \in \bigcup_{k=1}^t \mathcal{X}_k$, where $j \in \{1, \dots, |\mathcal{Y}_t|\}$ indicates the j -th class in task t . Based on theorem of Bayes, the goal can be decomposed as:

$$P(\mathbf{x} \in \mathcal{X}_{i,j} | D, \theta) = P(\mathbf{x} \in \mathcal{X}_{i,j} | \mathbf{x} \in \mathcal{X}_i, D, \theta) P(\mathbf{x} \in \mathcal{X}_i | D, \theta). \quad (3)$$

where $P(\mathbf{x} \in \mathcal{X}_i | D, \theta)$ is the task identity inference probability, $P(\mathbf{x} \in \mathcal{X}_{i,j} | \mathbf{x} \in \mathcal{X}_i, D, \theta)$ is the within-task prediction probability. Hence, when the performance of task identity inference is improved, the performance of CIL will be enhanced. For more proof details, we recommend referring to HiDe-Prompt.

1.3 More reasonable framework

Framing re-matching within a probabilistic framework is inspiring and reasonable. Assume the input is \mathbf{x} . The posterior probability that it belongs to class $y = c$ is $P(y = c | \mathbf{x}, D, \theta)$. We can further

Table 1: Notations in the paper.

| Notation | Description |
|-----------------|---|
| D_t | The set of training samples and labels for the t -th task |
| D | The set of training samples and labels for all tasks |
| x | The input image |
| x_i^t | The i -th input image in t -th task |
| y_i^t | The label of i -th input image in t -th task |
| \hat{d} | The output probability distribution in TII |
| \tilde{d} | The final output probability distribution in WTP |
| \hat{y} | The final predicted class in WTP |
| \hat{y}_{CRM} | The final predicted class after CRM in WTP |
| Γ | The top-N identities in initial matching |
| \hat{t}_f | The task identity in initial matching |
| \hat{t}_s | The task identity after re-matching |
| h_{ptm} | The pre-trained model |
| g_ω | The classifier for task identity in TII |
| g_{θ_g} | The classifier for class prediction in WTP |
| θ_{ptm} | Parameters of the pre-trained model |
| ω | Parameters of classifier for task identity in TII |
| θ_g | Parameters of classifier for class prediction in WTP |

decompose it into task identity inference and within-task prediction:

$$P(y = (i, j) \mid \mathbf{x}, D, \theta) = P(y = (i, j) \mid \mathbf{x}, \text{task} = i, D, \theta) \cdot P(\text{task} = i \mid \mathbf{x}, D, \theta),$$

where $y = (i, j)$ is the j -th class in task t . After training with PET, we obtain a parameter pool $\mathcal{P} = \{p_1, p_2, \dots, p_T\}$, where p_i denotes the learned parameters specific to task i . Assume that all p in \mathcal{P} are considered to confirm the final prediction. Rematching involves inference with multiple p . To aggregate these predictions in a principled manner, we draw upon Bayesian Model Averaging (BMA), which supports combining models under uncertainty about model selection. Applying BMA, the final posterior is expressed as:

$$P(y = (i, j) \mid \mathbf{x}, D, \theta, \mathcal{P}) = \sum_{i=1}^T P(y = j \mid \mathbf{x}, \text{task} = i, \theta, p_i) \cdot P(\text{task} = i \mid \mathbf{x}, D, \theta).$$

However, not all task parameters can be weighted equally. We hope that the correct task identity has a higher weight. Therefore, we calculate the weights ϕ_t by the conditions in our strategy, such as the confidence of CRM:

$$P(y = (i, j) \mid \mathbf{x}, D, \theta, \mathcal{P}) = \sum_{i=1}^T P(y = j \mid \mathbf{x}, \text{task} = i, \theta, p_i) \cdot P(\text{task} = i \mid \mathbf{x}, D, \theta) \cdot \phi_i.$$

In our paper, ϕ_i acts as a binary gate (i.e., $\phi_i \in \{0, 1\}$). Our rematching is uniformly implemented in the above equation.

2 Experiments

2.1 Experimental details

We employ the same supervised or self-supervised pre-trained ViT-B/16 as the backbone. The LoRA modules are integrated into the initial five layers of the ViT. The image inputs of all experiments are resized to 224x224 and then normalized. The means and standard deviations of experimental results under 3 different random seeds are reported. For the confidence function on the Split CIFAR-100 dataset, the MaxLogit function is employed, whereas for other datasets, the Generalized Entropy (GEN) function is utilized. γ and M in GEN are set to 0.01 and 20. We grid search for an appropriate threshold τ . For Split CIFAR-100 with $\tau \in \{0.5, 0.6, 0.7, 0.8\}$ of MaxLogit, we set $\tau = 0.8$ for better performance. For other datasets $\tau \in \{-12, -10, -8\}$ of GEN, we set $\tau = -10$ for Split ImageNet-R and 5-Datasets, $\tau = -8$ for ImageNet-A. We find that for the same dataset, the best

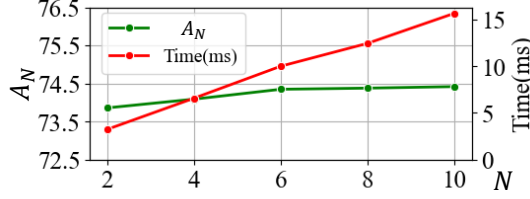


Figure 1: The influence of N on A_N and inference time.

τ under the settings of the self-supervised and the unsupervised pre-trained models are consistent. Similarly, we also conduct a grid search for more suitable learning rates l and training epochs E . For learning rates $l \in \{0.005, 0.01, 0.03\}$, all datasets are set to 0.03 under the Sup-21K pre-trained model, and others are set to 0.01. For fairer comparison and better performance, we use LoRA as PET in our method and HiDe-Prompt while setting the same learning rate and training epochs. For the supervised pre-trained model, the rank of LoRA is 5, and for others, it is 8. The class statistic distribution calculation method in TAP is multi-centroid. For other hyperparameters in HiDe-Prompt, we follow the settings in [1, 2]. Furthermore, our implementation on the 5-Datasets does not assume that images of the same batch come from the same task during inference, which would result in 100% matching accuracy. In fact, the matching accuracy of 5-dataset does not reach 100%. All experiments are performed on RTX3090.

2.2 Details of pre-trained weights

We observe some methods have different pre-trained weights for Sup-21K. For example, CODA-Prompt and Cprompt utilize a checkpoint that was first pretrained on ImageNet-21K and then fine-tuned under supervision on ImageNet-1K (**Sup21K/1K**¹). On the other hand, SLCA and InfLoRA claim to use weights pretrained on ImageNet-21K, but its reported weights (**Sup21K**^{†2}) differ from the **Sup21K**³ weights used in [1].

To ensure fairness and consistency across different methods, we conduct all evaluations using **Sup21K** as the standard pretrained weight.

2.3 Ablation of N and Limitation

In the Fig. 1, we present the influence of matching number N on A_N and inference time in ImageNet-R under Sup-21K. While A_N improves as N increases, the improvements become marginal. When N increases from 0 to 2, A_N improves by 1.26, whereas the increase from 2 to 10 improves by 0.65, with $5\times$ inference time. As a potential limitation, how to significantly improve A_N while increasing N is worth exploring in the future. Considering computational efficiency, N is set to 2. Moreover, N is robust across datasets and is 2 for all datasets and pre-trained weights.

2.4 The ablation of the confidence measure

To explore the effect of the confidence measure, we conduct ablation experiments on different post-hoc confidence measures. 1) Overall, our method is not sensitive to different confidence calculation methods. 2) GEN performs best across most settings. This is attributed to its design for semantic shift scenarios, i.e., detecting inputs with semantic categories that are absent from the training set. In CL, the PET parameters for task t are only trained on the semantic categories of task t . When predicted task identity \hat{t} does not match the true task t for sample x from task t , the semantic class of x is absent from the training set in task \hat{t} . Therefore, GEN is more advantageous in detecting mismatched samples. 3) On CIFAR-100, performance is near the upper bound with few mismatched samples. MaxLogit with a high threshold effectively filters mismatches, so we adopt it for better performance.

¹B_16-i21k-300ep-lr_0.001-aug_medium1-wd_0.1-do_0.0-sd_0.0-imagenet2012-steps_20k-lr_0.01-res_224.npz

²B_16-i21k-300ep-lr_0.001-aug_medium1-wd_0.1-do_0.0-sd_0.0.npz

³ViT-B_16.npz

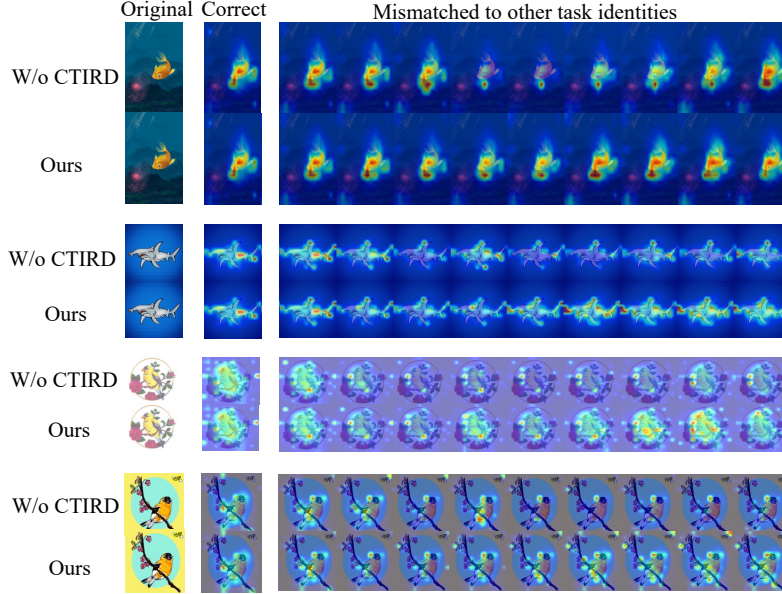


Figure 2: More visualization of attention regions with (Ours) and without CTIRD (W/o CTIRD). "Correct" means using the parameters corresponding to the correct task identity. "Mismatched to other task identities" means using parameters corresponding to other incorrect task identities.

Table 2: Comparison of different confidence calculation methods on Split CIFAR-100 and Split ImageNet-R under Sup-21K and iBOT-21K pre-trained model.

| Confidence | Split CIFAR-100 | | Split ImageNet-R | |
|------------|-----------------|--------------|------------------|--------------|
| | Sup-21K | iBOT-21K | Sup-21K | iBOT-21K |
| MSP | 89.23 | 89.20 | 73.73 | 74.65 |
| MaxLogit | 89.45 | 89.70 | 73.77 | 74.97 |
| Energy | 89.24 | 89.29 | 73.60 | 75.12 |
| GEN | 89.35 | 89.55 | 73.86 | 75.23 |

2.5 Discussion of scalability

To discuss the resource consumption on longer tasks, we conduct extra experiments on 20 tasks on ImageNet-R as shown in the Tab. 3. Since N in CRM is set to 2, the inference cost of our method only depends on the number of mismatched samples, e.g., the number of samples that meet the confidence threshold. When the number of tasks increases to 20, the matching error rate is higher, which inevitably leads to an increase in inference time. **However, we improve the utilization of inference resources.** Our method achieves a higher gain in accuracy per millisecond of increased inference time. The inference time per image only increases by 0.57 ms, but the accuracy has increased by 3.1%. Certainly, further reducing absolute inference cost remains a promising direction.

Table 3: The resource consumption on longer tasks.

| Method | 10 Tasks | | 20 Tasks | |
|----------|----------------|------------------------|----------------|------------------------|
| | $A_N \uparrow$ | Time (ms) \downarrow | $A_N \uparrow$ | Time (ms) \downarrow |
| Baseline | 71.60 | 2.81 | 68.03 | 2.81 |
| HRM-PET | 73.86 | 3.24 | 71.13 | 3.38 |

2.6 More Visualization

In order to verify the effectiveness of our CTIRD, we visualized the attention area after using parameters corresponding to different task identities on the test samples in Fig. 2.

References

- [1] L. Wang, J. Xie, X. Zhang, M. Huang, H. Su, and J. Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. In *NeurIPS*, 2024.
- [2] L. Wang, J. Xie, X. Zhang, H. Su, and J. Zhu. Towards a general framework for continual learning with pre-training. *arXiv preprint arXiv:2310.13888*, 2023.