Random Path Selection for Incremental Learning

Jathushan Rajasegaran **Munawar Hayat** **Salman Khan**

Fahad Shahbaz Khan **Ling Shao**

Inception Institute of Artificial Intelligence
`first.last@inceptioniai.org`

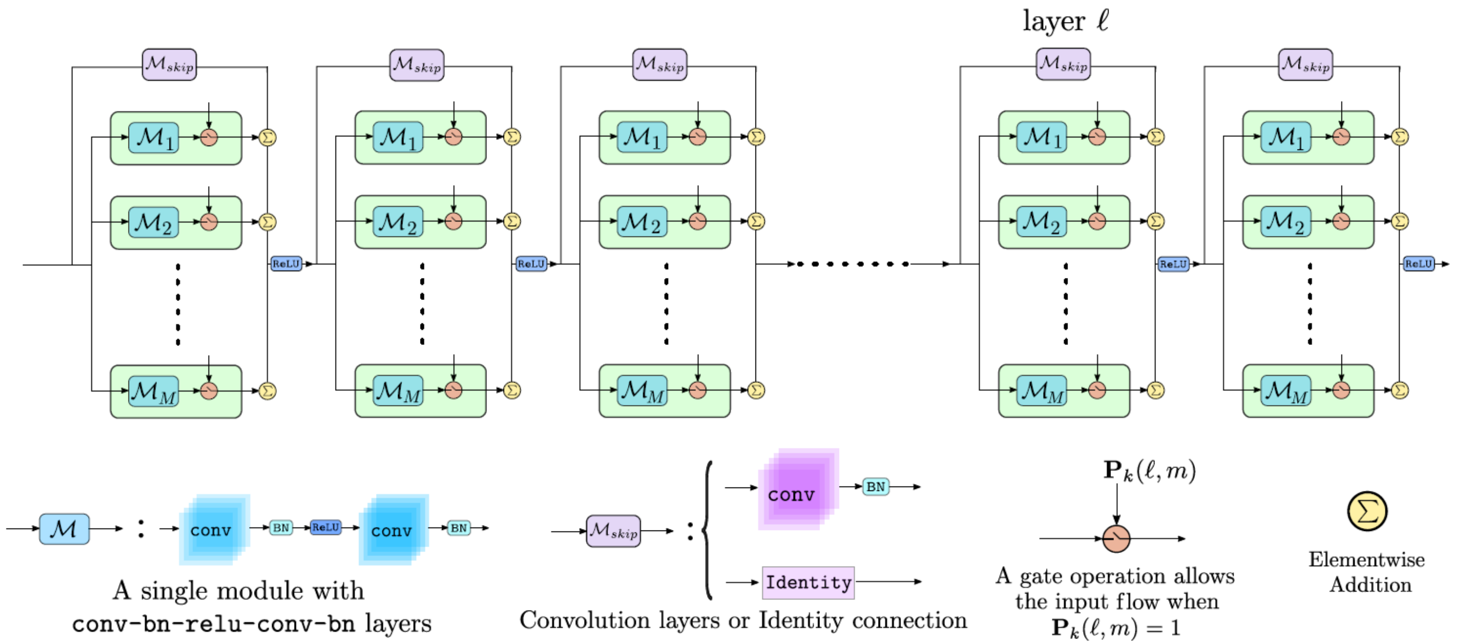


Figure 1: *An overview of our RPS-Net:* The network architecture utilizes a parallel residual design where the optimal path is selected among a set of randomly sampled candidate paths for new tasks. The residual design allows forward knowledge transfer and faster convergence for later tasks. The random path selection approach is trained with a hybrid objective function that ensures the right trade-off between network stability and plasticity, thus avoiding catastrophic forgetting.

路径选择:

- 每 J 个任务选择一次路径 （不合理）
- 训练时，只有一条路径被训练 （合理）
- 测试时，以往选中的路径同时使用 （不合理）

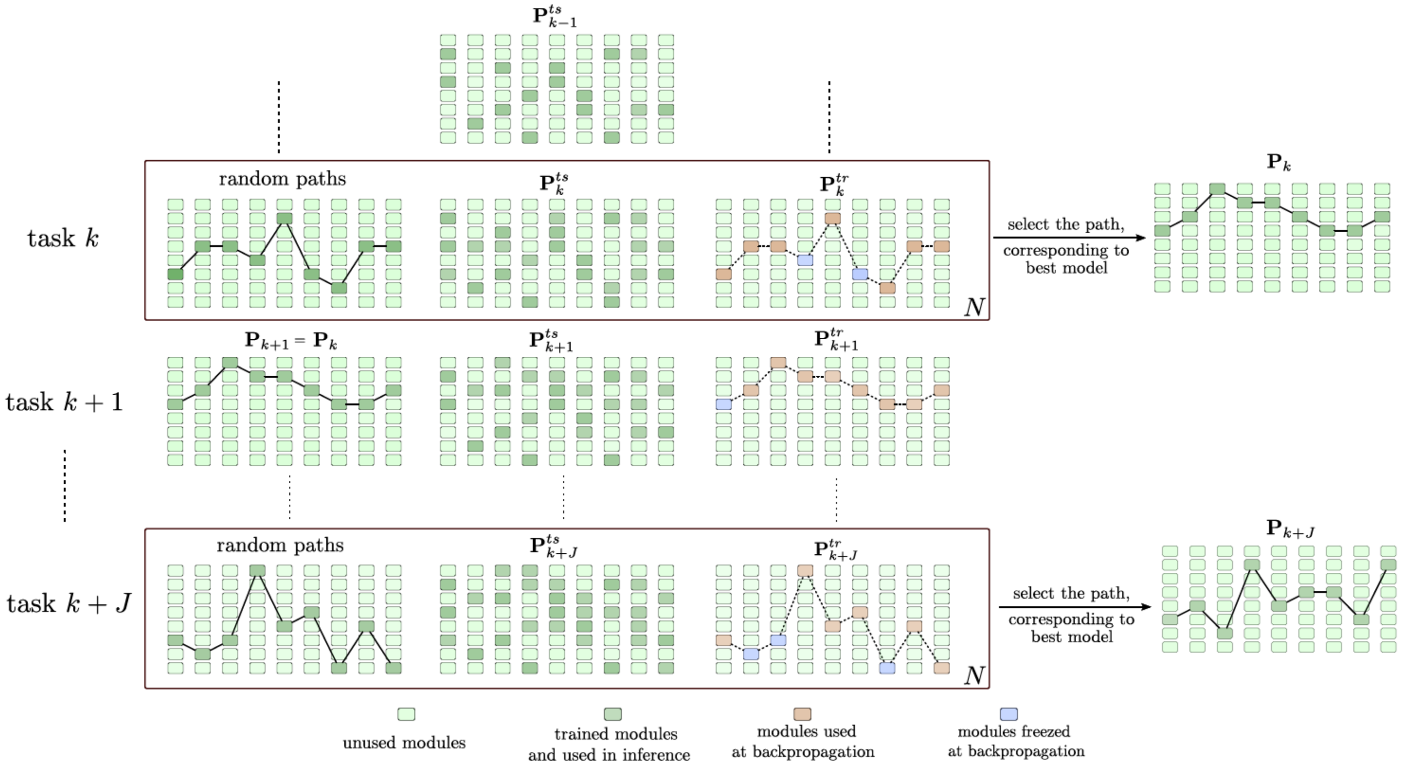
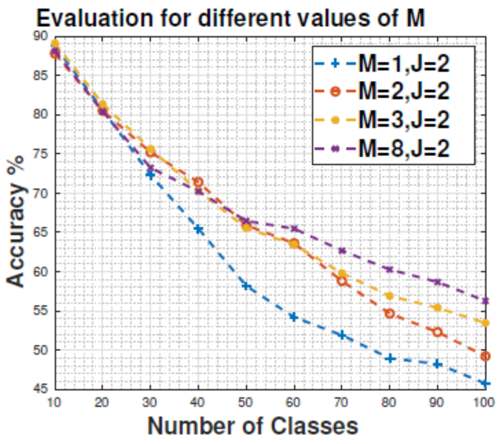


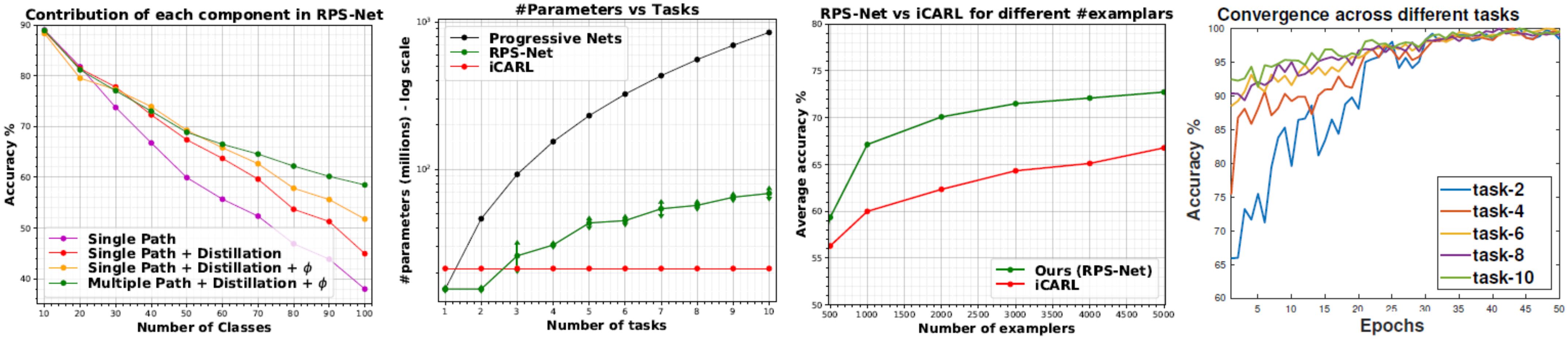
Figure 2: *Path Selection Approach:* Given a task k, N random paths are initialized. For each path, only the modules different from the previous inference path $\mathbf{P}_{k-1}^{ts}$ are used to form the training path $\mathbf{P}_k^{tr}$. Among N such paths, the optimal $\mathbf{P}_k$ is selected and combined with the $\mathbf{P}_{k-1}^{ts}$ to obtain $\mathbf{P}_k^{ts}$. Notably, the path selection is only performed after J tasks. During training, the complexity remains bounded by a standard single path network and the resources are shared between tasks.

$$\mathcal{L}_{ce} = -\frac{1}{n}\sum_i \mathbf{t}_i[1:k*U]\log(\mathrm{softmax}(\mathbf{q}_i[1:k*U])),$$ 学习新知识

$$\mathcal{L}_{dist} = \frac{1}{n}\sum_i \mathrm{KL}\left(\log\left(\sigma\left(\frac{\mathbf{q}_i[1:(k-1)*U]}{t_e}\right)\right), \sigma\left(\frac{\mathbf{q'}_i[1:(k-1)*U]}{t_e}\right)\right).$$ 保留旧知识

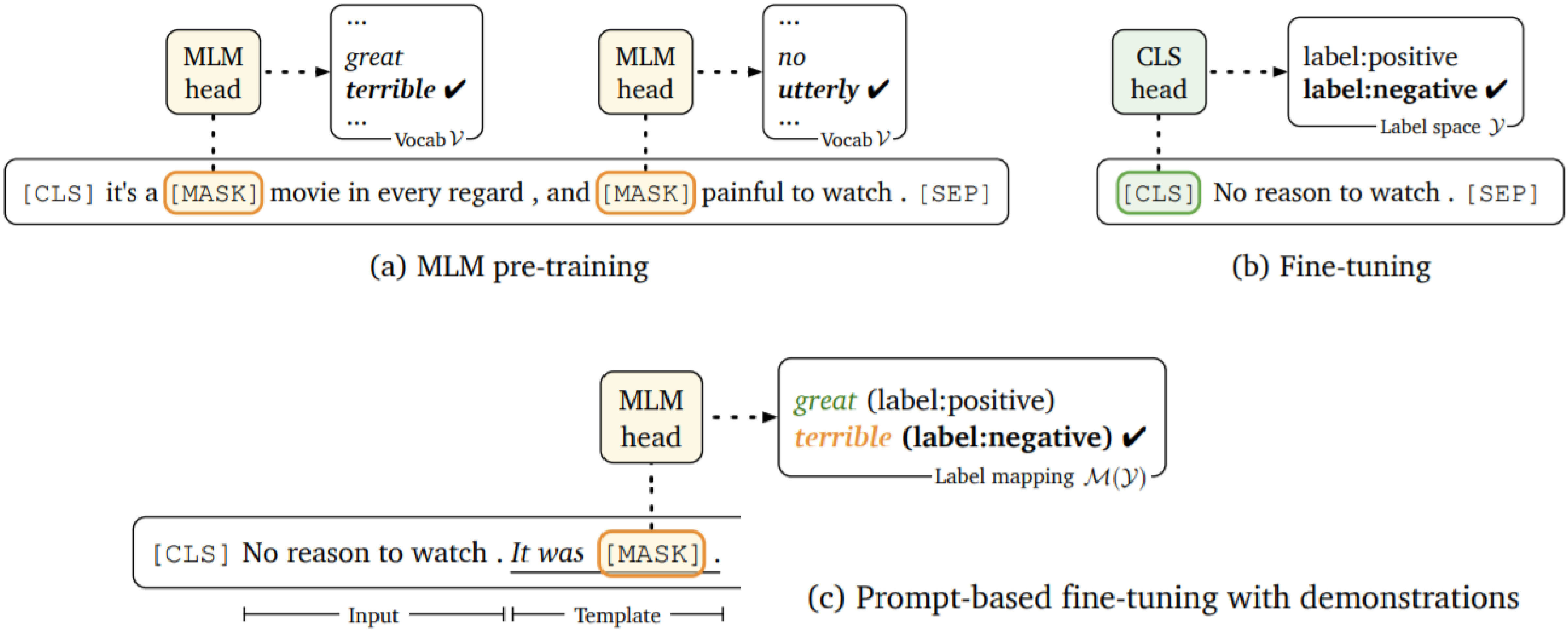
$$\mathcal{L} = \mathcal{L}_{ce} + \phi(k,\gamma)\cdot\mathcal{L}_{dist},$$


Evaluation for different values of M
M=1,J=2
M=2,J=2
M=3,J=2
M=8,J=2
Accuracy %
Number of Classes
10
20
30
40
50
60
70
80
90
100
45
50
55
60
65
70
75
80
85
90


Contribution of each component in RPS-Net
Single Path
Single Path + Distillation
Single Path + Distillation + ϕ
Multiple Path + Distillation + ϕ
Accuracy %
Number of Classes
#Parameters vs Tasks
Progressive Nets
RPS-Net
iCARL
#parameters (millions) - log scale
Number of tasks
RPS-Net vs iCARL for different #examplars
Ours (RPS-Net)
iCARL
Average accuracy %
Number of examplers
Convergence across different tasks
task-2
task-4
task-6
task-8
task-10
Accuracy %
Epochs

(a) (b) (c) (d)

Making Pre-trained Language Models Better Few-shot Learners

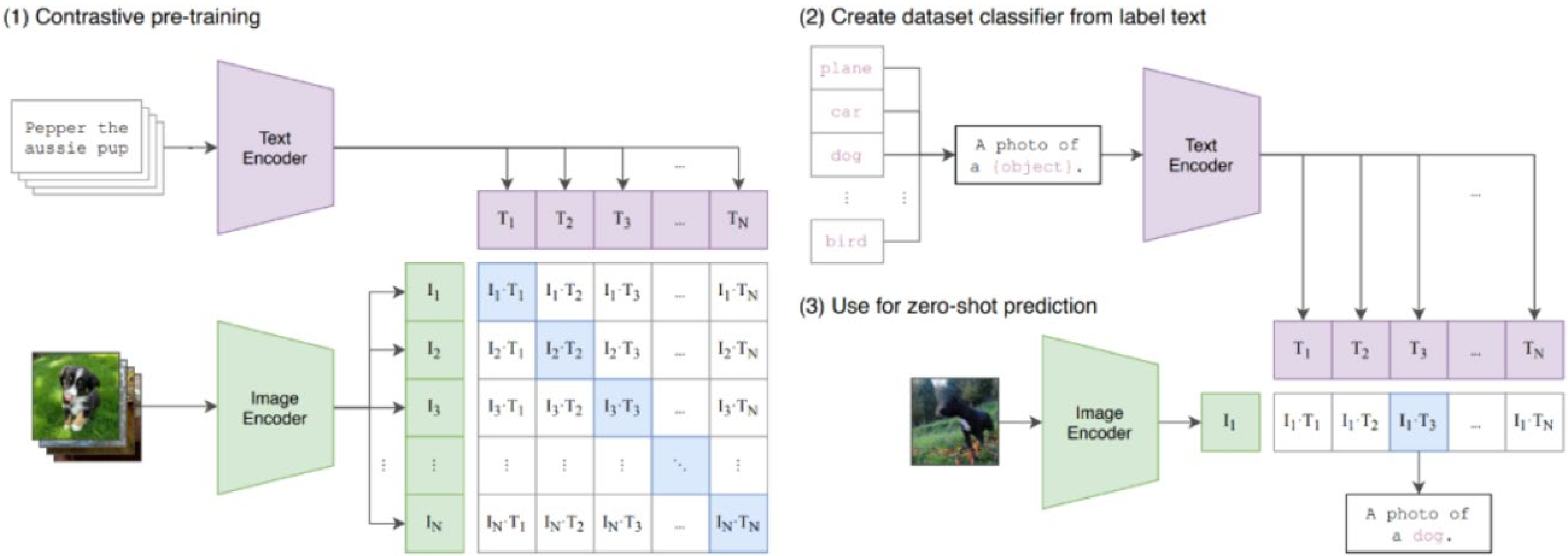


(a) MLM pre-training

(b) Fine-tuning

(c) Prompt-based fine-tuning with demonstrations

Learning Transferable Visual Models From Natural Language Supervision



Caltech101

Prompt	Accuracy
a [CLASS].	80.77
a photo of [CLASS].	78.99
a photo of a [CLASS].	84.42
$[V]_1 [V]_2 \ldots [V]_M$ [CLASS].	**92.00**

(a)

Flowers102

Prompt	Accuracy
a photo of a [CLASS].	56.68
a flower photo of a [CLASS].	61.23
a photo of a [CLASS], a type of flower.	62.32
$[V]_1 [V]_2 \ldots [V]_M$ [CLASS].	**93.22**

(b)

Describable Textures (DTD)

Prompt	Accuracy
a photo of a [CLASS].	38.24
a photo of a [CLASS] texture.	37.71
[CLASS] texture.	40.72
$[V]_1 [V]_2 \ldots [V]_M$ [CLASS].	**62.55**

(c)

EuroSAT

Prompt	Accuracy
a photo of a [CLASS].	22.30
a satellite photo of [CLASS].	31.12
a centered satellite photo of [CLASS].	31.53
$[V]_1 [V]_2 \ldots [V]_M$ [CLASS].	**81.60**

(d)

Figure 1: **Prompt engineering vs. context optimization (CoOp)**. The latter uses only 16 shots for learning in these examples.

Learning to Prompt for Vision-Language Models

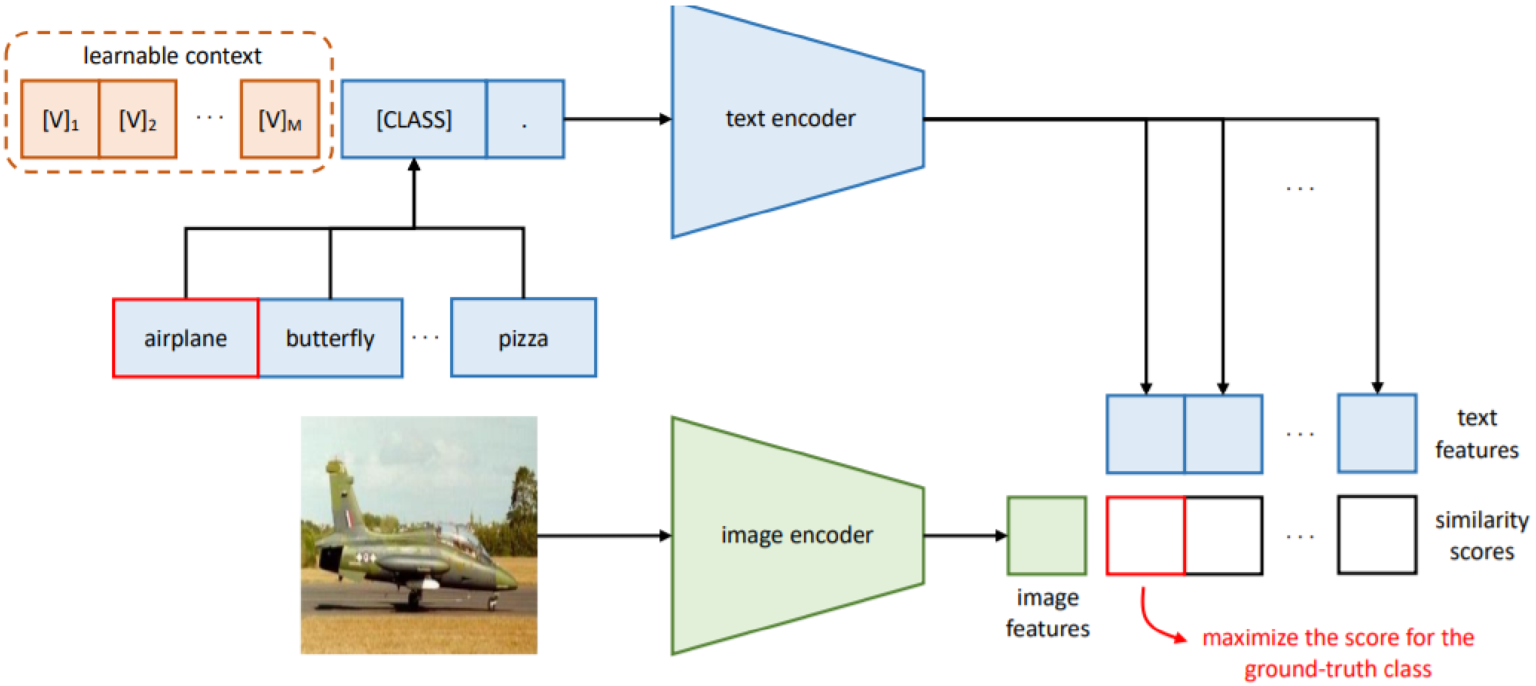


Figure 2: Overview of context optimization (CoOp).

LEARNING TO PROMPT FOR CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

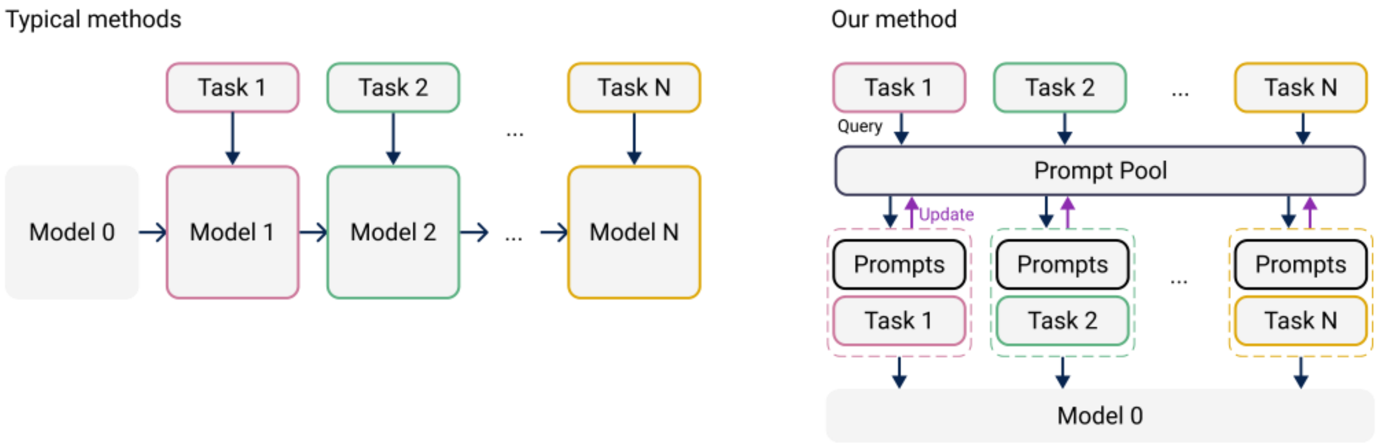


Figure 1: Overview of the L2P framework. Compared with typical continual learning methods (left) that adapt model weights to tasks sequentially, L2P (right) uses a single backbone model and learns a prompt pool to adapt tasks.

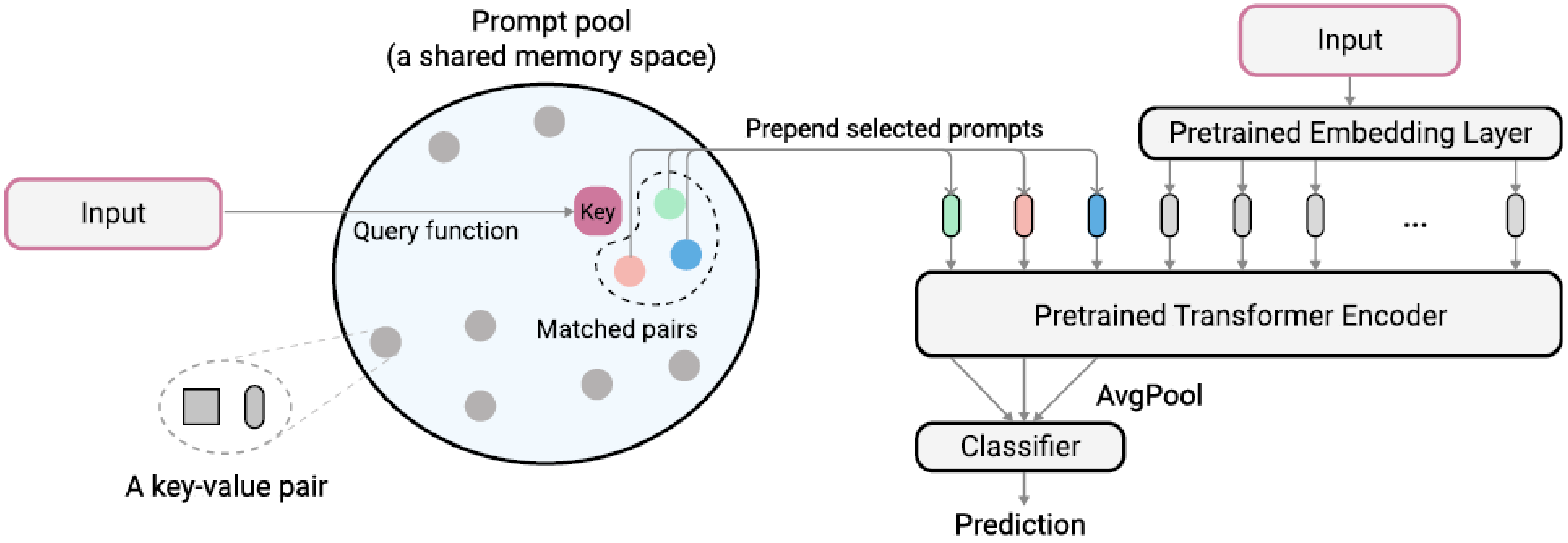


Figure 2: The illustration of L2P at test time. During training time, we follow the same procedure and optimize the model as described in Section 4.3.

$$\mathbf{P}_{\boldsymbol{x}} = \underset{\{s_i\}_{i=1}^{N} \subseteq [1,M]}{\operatorname{argmin}} \sum_{i=1}^{N} \gamma\left(q(\boldsymbol{x}), \boldsymbol{k}_{s_i}\right) \cdot h_{s_i},$$

Table 1: Results on class-incremental learning. Accuracy and forgetting are reported. All methods start from the same pre-trained ViTB/16 model and train on each task for 5 epochs. Methods are separated based on whether rehearsal is applied. All results are shown in percentage (%) and are averaged over 3 runs.

Method	**Split CIFAR-100**		**5-datasets**	
	Average Acc (↑)	Forgetting (↓)	Average Acc (↑)	Forgetting (↓)
Upper bound:				
FT-iid	90.85±0.12	-	93.93±0.18	-
Non-rehearsal based methods:				
FT-seq-frozen	17.72±0.34	59.09±0.25	39.49±0.12	42.62±0.20
FT-seq	33.61±0.85	86.87±0.20	20.12±0.42	94.63±0.68
EWC	47.01±0.29	33.27±1.17	50.93±0.09	34.94±0.07
LwF	60.69±0.63	27.77±2.17	47.91±0.33	38.01±0.28
L2P (ours)	**83.83±0.04**	**7.63±0.30**	**81.14 ±0.93**	**4.64 ±0.52**
Rehearsal based methods:				
ER	82.53±0.17	16.46±0.25	89.30±0.94	8.08±0.53
GDumb	81.67±0.02	-	70.76±0.12	-
L2P-R (ours)	**86.31±0.59**	**5.83±0.61**	**91.92±0.78**	**3.34±0.71**

Table 4: Ablation study on 5-datasets. All results are shown in percentage (%).

Method	**5-datasets** Average Acc (↑)	Forgetting (↓)
L2P without prompt pool	51.96	26.60
L2P without key-value pair	58.33	20.45
L2P without diversified prompt selection	62.26	17.84
L2P	**81.14**	**4.64**

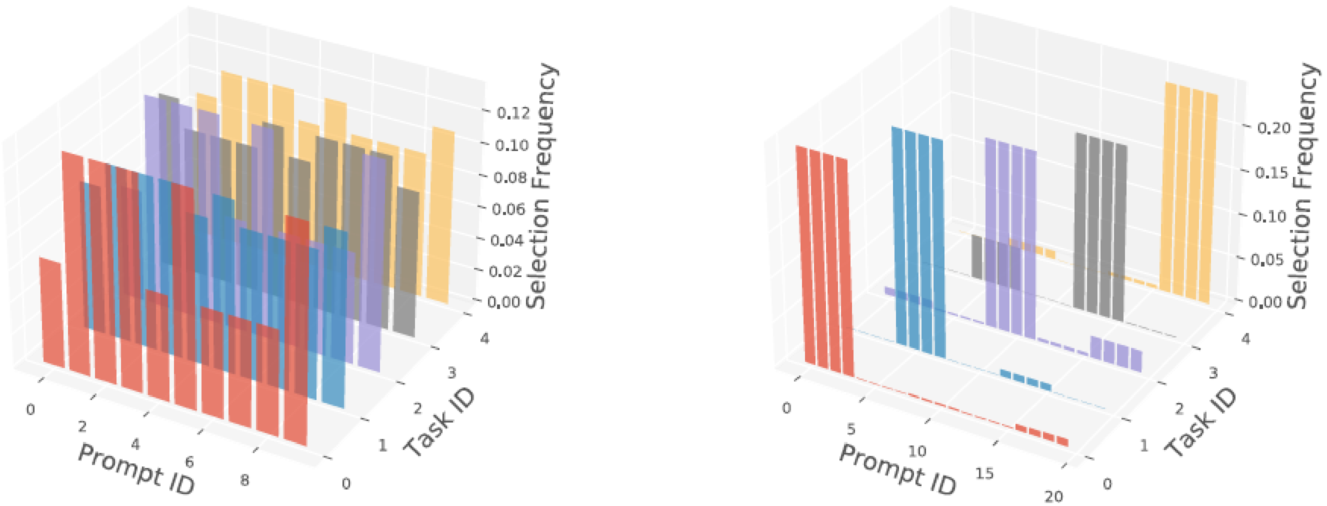


Figure 3: Prompt selection histograms for (left) Split CIFAR-100 and (right) 5-datasets. Note that we only show the first 5 tasks for Split CIFAR-100 for better readability.