Instruction-Grounded Visual Projectors for Continual Learning of Generative Vision-Language Models

Supplementary Material

A. Additional Implementation Details

Details of Router. We concatenated the image feature and instruction token and used them as input to the router. To align the dimension of the image feature with that of the instruction token, we first applied a pre-trained visual projector to the image feature. To more effectively extract contextual information from the given instructions, we employed a simple linguistic preprocessing technique, following [10]. Specifically, we applied Part-of-Speech (POS) tagging to categorize key components, such as WH-words, nouns, and verbs, while filtering out irrelevant terms. The filtered instructions were then processed through the tokenizer and embedding function of a pre-trained model to extract text tokens for the router. Note that this process was not applied to the instruction tokens for the pre-trained large language model. Finally, the projected image feature was concatenated with the filtered instruction token and used as input to the router.

To activate an expert by considering both the given image and instruction, we implemented the router consisting of a single multi-head self-attention block followed by a linear layer. We set the number of translation experts K engaged in translation to two for all experiments, following previous practices [36, 50]. We pruned experts by applying a threshold to E^t, removing experts whose corresponding values in E^t were smaller than 10^{-3}. After identifying redundant experts, we reinitialized those that had not been activated across all tasks learned thus far (i.e., zero-valued entries in the cumulative activation vector $\mathcal{M}^{1:t}$) using the parameters of the visual projector from the pre-trained model.

Details of the Language Instructions. The language instructions used as the default setting in our experiments are presented in Table B. For question answering tasks, the instruction varies depending on the sample.

Measuring Accuracy of Generated Responses. For the classification and question answering tasks, the performance of the generated responses is evaluated in terms of accuracy. To measure accuracy for classification tasks, we compute the similarity in the embedding space between the textual representations of the classification categories and the generated responses. The predicted class is the category with the highest similarity to the generated response; if it matches the ground truth, the response is considered correct. To compute textual similarity, we extract embeddings using the text encoder of CLIP [40], following [4]. For question answering tasks, the accuracy is determined by whether the answer text appears in the generated response, following previous practices [7, 33].

Table A. Results on a different task order.

Method	Classification		Captioning		Question Answering	
	Average	Δ (↑)	Average	Δ (↑)	Average	Δ (↑)
Vicuna						
Zero-shot	64.41	0.00	75.00	0.00	51.62	0.00
Last						
LwF [30]	70.78	+6.37	70.46	-4.54	57.18	+5.56
EWC [25]	47.57	-16.84	72.37	-2.63	52.42	+0.80
GMM [4]	60.62	-3.79	73.83	-1.17	54.93	+3.31
EProj [16]	59.26	-5.15	71.41	-3.59	58.97	+7.35
MoEAdapter [50]	66.19	+1.78	75.53	+0.53	47.60	-4.02
MVP (Ours)	**86.68**	**+22.27**	**77.55**	**+2.55**	**70.93**	**+19.31**
Avg						
LwF [30]	62.88	-1.53	65.36	-9.64	52.72	+1.10
EWC [25]	47.43	-16.98	63.03	-11.97	40.43	-11.19
GMM [4]	46.76	-17.65	72.54	-2.46	45.89	-5.73
EProj [16]	47.24	-17.17	72.41	-2.59	45.54	-6.08
MoEAdapter [50]	67.84	+3.43	75.53	+0.53	51.91	+0.29
MVP (Ours)	**81.67**	**+17.26**	**77.12**	**+2.12**	**60.87**	**+9.25**
Transfer						
LwF [30]	70.70	+6.29	63.21	-11.79	49.87	-1.75
EWC [25]	57.77	-6.64	61.92	-13.08	39.85	-11.77
GMM [4]	55.01	-9.40	72.43	-2.57	44.74	-6.88
EProj [16]	50.81	-13.60	72.67	-2.33	43.72	-7.90
MoEAdapter [50]	69.07	+4.66	75.92	+0.92	49.86	-1.76
MVP (Ours)	**76.09**	**+11.68**	**77.13**	**+2.13**	**55.98**	**+4.36**

B. Additional Experimental Results

Results on a Different Task Order. The task order used in Tables 1 and 2 of the main paper follows the sequence of ten classification tasks, four captioning tasks and four question answering tasks. Since captioning and question answering tasks do not have explicit classes, experiments on different class orders are omitted. To evaluate the robustness of the proposed method to a different task order, we conducted additional experiments. We randomly shuffled the task order used in Table 1 of the main paper while keeping all other implementation details the same. Table A reports the experimental results of learning with a random task order. Overall, the proposed method outperforms other continual learning methods. Most competitors show unsatisfactory performance for the three metrics compared to Zero-Shot. Additionally, we report the task-specific performance measured at each time step in Figure A. While other continual learning methods exhibit large performance variations depending on the type of tasks being learned, the proposed method demonstrates robustness with respect to the task order.

Analysis on Language Instructions. Additionally, to analyze the robustness of the proposed instruction-grounded

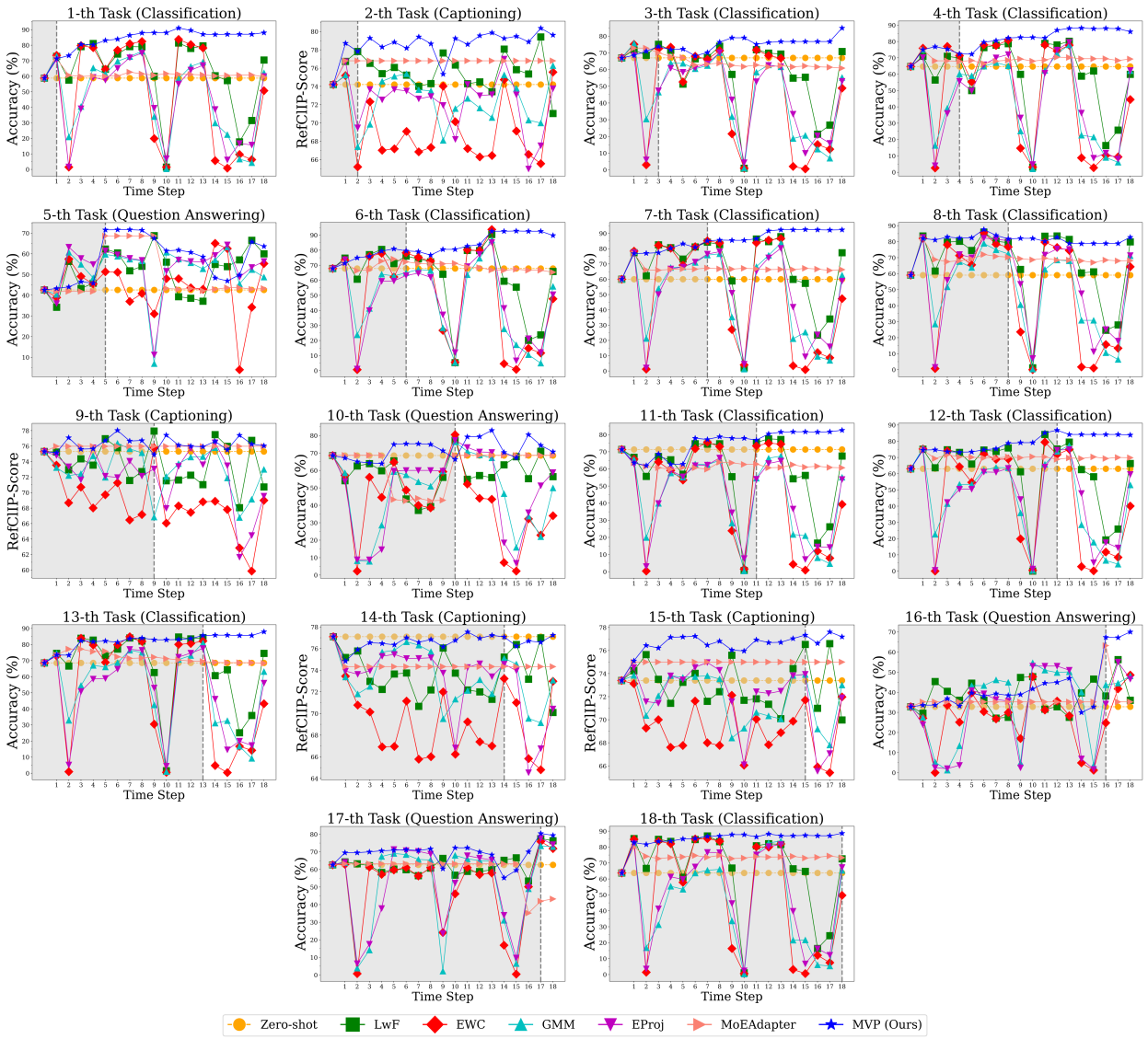

Figure A. Results of training with a random task order, showing the performance of each task at each time step. The yellow horizontal line represents the zero-shot performance of the pre-trained model. The gray area represents the time steps preceding training for specific tasks, and the vertical line indicates the time step at which the corresponding task is learned. Best viewed in color.

visual projector when multiple textual instructions are provided per task, we conducted an additional experiment. We generated ten language instructions using GPT-4o [1] for image classification and captioning tasks. The generated instructions for each task category are shown in the generated response in Table B. For tasks belonging to the corresponding category, one of the ten generated instructions was randomly selected for each sample during training. The experimental results are reported in Table C. The results indicate that the proposed method exhibits robustness, as its performance remains stable regardless of variations in instructions, provided that the instructions maintain contextual consistency. Although an increase in the number of instructions leads to a slight decline in the transfer performance of the captioning and question answering tasks, the performance measured in *Last* is comparable to those achieved using the default instructions.

Analysis on Computational Cost. We analyzed the computational cost of MVP by measuring the wall-clock time and VRAM usage for the experiments in Table 1 of the main paper. The results are summarized in Table D. During its main training phase, MVP requires 20 hours and 48 minutes, a negligible time overhead (1.08×) compared to GMM. This efficiency stems from freezing the computationally expensive vision and language backbones of the VLM while updating only the lightweight components: the

Table B. Language instructions for each type of task.

Task	**Classification**
Default	· What is this photo of?
Generated	· Analyze the image to identify its most detailed category. · The most detailed category should be determined based on the image's visual features. · Assign the image to the most detailed category by examining its characteristics. · Determine which detailed category best fits the given image and identify it. · Based on the primary subject, identify the detailed category of the image. · The image should be assessed to determine the most detailed category. · Identify the correct detailed category for the given image. · Assign the image to its respective detailed category after carefully identifying its features. · Give the image a detailed category by examining its characteristics. · Given the image's appearance, identify the appropriate detailed category.
Task	**Captioning**
Default	· Describe the given image with a short sentence
Generated	· Describe the scene in detail by capturing its key visual elements, interactions, and overall atmosphere. · The scene contains various elements, and it is essential to describe them clearly while maintaining a concise yet informative structure. · Identify and describe the key subjects, their actions, and the interactions that define the scene comprehensively. · A meaningful explanation should effectively describe the scene by considering its composition, significant events, and contextual details. · The scene consists of numerous visual details, so it is important to describe them with clarity to ensure a precise understanding. · Observe the scene carefully and describe the relationships between objects, people, and their interactions within the given context. · Given the scene in the image, provide a well-structured statement that naturally describes its key features and dynamics. · A well-formed caption must thoroughly describe the significant aspects of the scene, ensuring an accurate depiction of its core elements. · As the scene unfolds dynamically, it is necessary to describe its core features with well-structured and concise sentences. · Carefully examine the objects and contextual details in the scene to naturally describe its composition, interactions, and overall atmosphere.
Task	**Question Answering**
Default	{Sample-wise Question}
Generated	N/A

Table C. Results using generated instructions.

Method	Classification		Captioning		Question Answering	
	Average	Δ (↑)	Average	Δ (↑)	Average	Δ (↑)
Vicuna						
Zero-shot	64.41	0.00	75.00	0.00	51.62	0.00
Last						
MVP w/ Default	85.87	+21.46	77.75	+2.75	76.34	+24.72
MVP w/ Generated	84.99	+20.58	78.25	+3.25	75.77	+24.15
Avg						
MVP w/ Default	83.28	+18.87	76.94	+1.94	57.68	+6.06
MVP w/ Generated	84.19	+19.78	74.44	-0.56	54.28	+2.66
Transfer						
MVP w/ Default	78.45	+14.04	76.16	+1.16	50.89	-0.73
MVP w/ Generated	79.38	+14.97	72.64	-2.36	45.51	-6.11

Table D. Analysis of computational cost.

Vicuna	GMM (train)	EProj (train)	MVP (train / prune / finetune)
Wall-clock time	19h 58min	20h 02min	20h 48min / 24min / 22min
Memory (VRAM)	20.5GB	20.6GB	23.9GB / 4.0GB / 3.1GB

Table E. Results on SEED-Bench after training all tasks.

LLaMa-2	Zero-Shot	GMM	EProj	MoEAdapter	MVP
SEED-Bench [28]	42.66	32.08	29.69	34.72	42.59

router, the set of expert visual projectors, and the learnable pruning vector E^t. In terms of memory, MVP requires 23.9 GB of VRAM, a marginal increase (1.16×) over GMM from the use of multiple experts. Notably, the subsequent prune and finetune stages are highly efficient, each completing in under 25 minutes and requiring less than 4.0 GB of VRAM. This low cost demonstrates that subsequent learning phase within our framework is highly practical.

Results on Standard VLM Benchmark. To measure generalization capability, we evaluated the performance on SEED-Bench [28] after their training on all tasks. As shown in Table E, the compared methods exhibit a significant performance degradation from the Zero-Shot score, with a performance gap ranging from 7.94 to 12.97. In contrast, MVP achieves a score of 42.59, which is the closest to the Zero-Shot performance. This performance retention is attributed to our adaptive knowledge aggregation strategy. For unseen data, this strategy minimizes the influence of the trained experts and relies on the retained knowledge of the original pre-trained projector, thus preventing performance degradation.