INTERLACE: Interleaved Layer Pruning and Efficient Adaptation in Large Vision-Language Models

Supplementary Material

A. Appendix

A.1. Per-Benchmark Inference Speedup Analysis

To provide a comprehensive understanding of the computational benefits achieved by our method, we report detailed Time-To-First-Token (TTFT) speedup measurements across all twelve evaluation benchmarks in Table 4. These measurements were obtained by averaging TTFT across all samples in each benchmark using Qwen3-VL 8B and 4B models with 10 to 25% layer pruning ratios. As the pruning ratio increases to 15% and beyond, all benchmarks exhibit clear acceleration benefits. Fine-grained high-resolution benchmarks such as HRBench4K and HRBench8K show more modest speedup gains across pruning ratios compared to other tasks, which we attribute to the computational bottleneck shifting toward vision encoding and projection operations rather than language model inference in these vision-intensive scenarios. Conversely, text-heavy benchmarks like AI2D, ChartQA, and ScienceQA demonstrate the highest speedup ratios at 25% pruning, reaching up to $1.22\times$ acceleration, as their longer token sequences amplify the benefits of reduced layer computations.

A.2. Comprehensive Ablation Study Results

Table 5 presents the complete performance breakdown of our ablation studies across all twelve benchmarks, extending the aggregated results shown in Table 3. Each ablation variant was evaluated under identical conditions with the 8B model at 25% pruning ratio and chain-of-thought reasoning enabled. The detailed absolute performance scores reveal several critical patterns in how different layer selection strategies affect specific task categories. The consecutive dropping baseline exhibits catastrophic degradation on general visual question answering tasks, achieving only 8.3% on MMBench compared to the 70.5% achieved by INTERLACE, which represents a striking 62.2 percentage point gap. This severe collapse suggests that vision-language alignment and cross-modal reasoning are especially vulnerable to large blocks of consecutive architectural modifications. Perception-intensive tasks such as HRBench4K, HRBench8K, and VStar also suffer dramatically under consecutive pruning, losing over 30 absolute percentage points compared to INTERLACE. Random layer selection substantially improves upon consecutive dropping, with MMBench performance jumping to 53.0%, but still exhibits significant gaps across all benchmarks. The Interlace-OA variant, which uses fixed positional assignment within triplets, achieves competitive performance across most tasks, reaching 65.9% on AI2D and 85.9% on POPE, demonstrating that the triplet structure alone provides substantial benefits. Interlace-TN performs remarkably close to the full INTERLACE method across nearly all benchmarks, with particularly strong results on perception tasks where it achieves 68.5% on HRBench4K and 64.9% on VStar. This validates the effectiveness of individual layer importance scoring while highlighting that our complete triplet-based framework with explicit anchor placement provides the most consistent and predictable performance across the diverse benchmark suite.

A.3. Layer and Triplet Similarity Distribution Analysis

Figure 4 visualizes the distribution of cosine similarity scores for both individual layers and triplets across the entire depth of the Qwen3-VL-8B and 4B models. The similarity scores were computed using 10% of the fine-tuning dataset (24,000 samples), measuring the average cosine similarity between hidden states before and after each layer or triplet. Both models exhibit higher redundancy in middle layers, with similarity scores peaking around layers 15-25, suggesting that these regions contain more functionally overlapping transformations that can be safely removed. The deeper layers beyond layer 30 show lower similarity scores, indicating more specialized and less redundant functionality that is critical for final output generation. The triplet similarity scores generally follow similar trends to individual layer scores but with smoother transitions.

A.4. Fine-Tuning Effects on Unpruned Baseline Models

Table 6 provides quantitative analysis of how fine-tuning affects the unpruned baseline model both with and without chain-of-thought reasoning, expanding upon the visualization presented in Figure 3 of the main paper. We fine-tuned the last 25% of layers in the Qwen3-VL-8B model on 1% of FineVision for one epoch, matching the training configuration used for our pruned models. With chain-of-thought reasoning enabled by setting the maximum token generation to 16384, the unpruned dense baseline achieves an average of 84.3% performance, while fine-tuning degrades this to 81.6%, representing a 2.7 percentage point drop. When chain-of-thought reasoning is limited to 50 tokens, the pattern becomes more complex. The baseline performance drops to 78.7% on average without extended

Table 4. Time-To-First-Token (TTFT) speedup for INTERLACE across different pruning ratios and model sizes. All speedup factors are relative to the unpruned baseline models.

		Qwen3-VL-8B				Qwen3-VL-4B			
Category	Benchmark	10%	15%	20%	25%	10%	15%	20%	25%
Text/Chart	AI2D	0.98	1.11	1.15	1.21	0.96	1.10	1.14	1.19
	ChartQA	0.97	1.11	1.16	1.21	0.95	1.09	1.14	1.19
	OCRBench	1.02	1.08	1.11	1.15	1.00	1.08	1.11	1.14
	TextVQA	1.02	1.10	1.15	1.20	0.99	1.09	1.14	1.19
	Average	1.00	1.10	1.14	1.19	0.98	1.09	1.13	1.18
GVQA	MMBench	0.96	1.11	1.16	1.22	0.95	1.10	1.14	1.19
	POPE	0.95	1.11	1.16	1.22	0.95	1.10	1.14	1.20
	RealWorldQA	1.01	1.09	1.14	1.18	0.98	1.09	1.13	1.17
	Average	0.97	1.10	1.15	1.21	0.96	1.10	1.14	1.19
Perception	HRBench4K	1.03	1.05	1.08	1.10	1.02	1.06	1.08	1.10
	HRBench8K	1.03	1.05	1.08	1.10	1.03	1.05	1.08	1.10
	VStar	1.06	1.10	1.13	1.17	1.00	1.08	1.11	1.14
	Average	1.04	1.07	1.10	1.12	1.02	1.06	1.09	1.11
Inst&Sci	MIABench	1.02	1.07	1.10	1.13	1.00	1.07	1.10	1.13
	ScienceQA	0.96	1.12	1.16	1.22	0.95	1.10	1.15	1.20
	Average	0.99	1.09	1.13	1.18	0.98	1.08	1.12	1.17
Overall		**1.00**	**1.09**	**1.14**	**1.18**	**0.99**	**1.08**	**1.12**	**1.17**

Table 5. Complete ablation study results for Qwen3-VL 8B with 25% pruning rate. Results show absolute percentage performance values for each benchmark task with chain-of-thought reasoning enabled. Results are after one epoch of training on 1% of FineVision dataset.

Benchmark	Consecutive	Random	Interlace-OA	Interlace-TN	INTERLACE	Dense
AI2D	33.8	52.8	65.9	69.8	**72.3**	85.7
ChartQA	62.2	64.2	67.7	**73.0**	72.0	83.3
OCRBench	67.1	75.0	76.7	77.1	**77.8**	89.6
TextVQA	64.9	69.4	74.1	**75.1**	74.5	82.9
MMBench	8.3	53.0	65.6	67.6	**70.5**	85.0
POPE	75.8	85.8	85.9	86.5	**87.7**	88.0
RealWorldQA	43.3	54.8	60.9	60.8	**62.4**	71.5
HRBench4K	35.3	53.6	61.9	**68.5**	66.3	78.9
HRBench8K	35.4	49.4	54.4	**63.0**	61.3	74.6
VStar	37.7	50.3	61.3	64.9	**70.7**	86.4
MIABench	62.5	73.7	**78.6**	76.8	77.6	91.1
ScienceQA	49.0	63.4	76.0	76.8	**78.7**	94.2
Average	47.9	62.1	69.1	71.7	**72.7**	84.3

reasoning capability, but fine-tuning actually improves performance to 81.2%, suggesting that the fine-tuning process helps the model adapt to generate more concise responses. Notably, AI2D shows a dramatic improvement from 69.9% to 83.0% under limited CoT conditions after fine-tuning, while ScienceQA jumps from 75.2% to 91.7%. However, this improvement comes at the cost of reduced performance on several other benchmarks.

A.5. Fine-Tuning Hyperparameter Configuration

In our fine-tuning procedure, we use AdamW optimizer with a learning rate of 1×10^{-5}. The learning rate follows a cosine annealing schedule with a warmup ratio of 0.03 (3% of total training steps), which allows the optimizer to gradually adapt to the modified architecture. We explicitly set weight decay to 0 to avoid constraining the model's ability to rapidly adapt its representations during the short one-

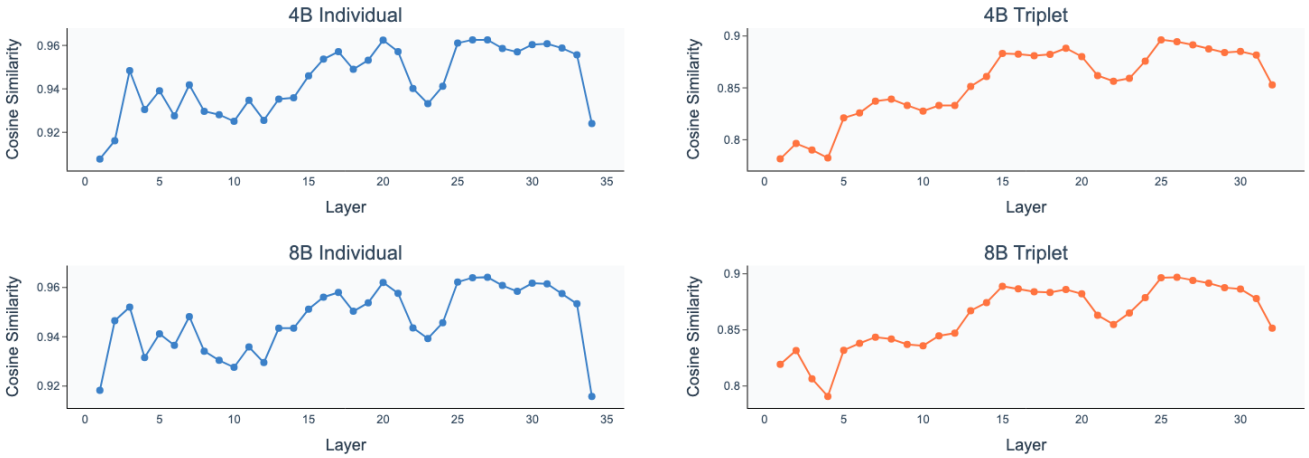

Figure 4. Distribution of cosine similarity scores for individual layers and triplets across the depth of Qwen3-VL-8B (blue) and 4B (orange) models. Higher scores indicate greater redundancy.

Table 6. Effect of fine-tuning 25% of layers in the unpruned dense baseline under enabled or limited Chain-of-Thought (CoT) reasoning. Dense-FT is fine-tuned on 1% of the FineVision dataset for one epoch. Values denote absolute performance scores (%).

Benchmark	CoT ✓		CoT ✗	
	Dense	Dense-FT	Dense	Dense-FT
AI2D	85.7	82.9	69.9	83.0
ChartQA	83.3	83.6	83.3	83.4
OCRBench	89.6	85.8	81.2	85.3
TextVQA	82.9	81.4	82.9	81.0
MMBench	85.0	82.7	77.9	82.1
POPE	88.0	88.1	88.0	88.3
RealWorldQA	71.5	68.6	71.4	70.1
HRBench4K	78.9	74.9	77.0	74.5
HRBench8K	74.6	69.8	70.8	70.8
VStar	86.4	82.7	81.7	82.2
MIABench	91.1	87.2	85.0	81.7
ScienceQA	94.2	92.1	75.2	91.7
Average	84.3	81.6	78.7	81.2

epoch training regime. We employ a per-device batch size of 16 with 2 gradient accumulation steps, yielding an effective batch size of 32, which fits within the memory constraints of our NVIDIA RTX PRO 6000 GPUs while maintaining stable gradient estimates. Mixed-precision training using bfloat16 arithmetic reduces memory consumption compared to full float32 training while preserving numerical stability. We use gradient clipping with a maximum norm of 1.0 to prevent occasional large gradients from destabilizing training, which is particularly important given that we are fine-tuning only a sparse subset of layers. We apply gradient checkpointing to all trainable layers. We utilize 4 dataloader workers and employ DeepSpeed ZeRO Stage 3 optimization with overlap communication and contiguous gradients enabled to further reduce memory footprint and enable training on consumer-grade hardware.

Table 7. End-to-end inference speedup and FLOPS reduction comparison on Qwen3-VL-8B-Instruct across all benchmarks with Chain-of-Thought (CoT) reasoning enabled. RP denotes average relative performance compared to the dense baseline.

Method	Sparsity	Speedup↑	FLOPS↓	RP (%)
Dense	0%	1.00×	1.00×	100.0
Wanda 2:4	50%	0.94×	1.05×	8.9
Magnitude 2:4	50%	0.95×	1.03×	9.5
SLEB	25%	1.14×	0.98×	60.5
INTERLACE	25%	**1.81**×	**0.95**×	**88.9**

A.6. End-to-End Inference Speedup and FLOPS Reduction

While Table 4 in the main appendix reports Time-To-First-Token (TTFT) speedup on a per-benchmark basis, we additionally measure end-to-end inference speedup and reduction in Floating-Point Operations (FLOPS) across all benchmarks for each pruning method on Qwen3-VL-8B-Instruct. Table 7 summarizes these results. INTERLACE achieves a 1.81× end-to-end speedup with a 0.95× FLOPS factor while retaining 88.9% relative performance.

A.7. Effect of Dataset Size and Domain-Specific Fine-Tuning

To examine the sensitivity of INTERLACE to the amount and composition of fine-tuning data, we conduct two sets of experiments on Qwen3-VL-8B at 25% layer pruning with chain-of-thought reasoning enabled.

Dataset size ablation. We vary the proportion of the

Table 8. Dataset ablation on Qwen3-VL-8B at 25% layer pruning with COT enabled. Top: effect of varying the FineVision subset size on overall average relative performance. Bottom: comparison of general-purpose (1% FineVision) vs. domain-specific fine-tuning on individual benchmarks.

Fine-Tuning Data	Test Benchmark	Rel. Perf. (%)
Dataset Size Ablation (Overall Average)		
0.5% FineVision	All Benchmarks	84.9
1% FineVision		86.1
3% FineVision		86.7
Domain-Specific Fine-Tuning		
1% FineVision	$\text{ChartQA}_{\text{Test}}$	**86.5**
ChartQA train		81.7
1% FineVision	$\text{TextVQA}_{\text{Val}}$	**89.7**
TextVQA train		82.9
1% FineVision	$\text{ScienceQA}_{\text{Test}}$	83.5
ScienceQA train		**85.4**

FineVision dataset used for post-pruning fine-tuning, comparing 0.5%, 1% (our default), and 3% subsets while keeping all other hyperparameters fixed. Table 8 reports the average relative performance across all twelve benchmarks. Reducing the data to 0.5% yields 84.9% relative performance, 1.2 percentage points below our default 1% setting (86.1%), while increasing to 3% provides a gain of 0.6 percentage points (86.7%).

Domain-specific fine-tuning. We additionally investigate whether task-specific training data can outperform our general-purpose FineVision fine-tuning on individual benchmarks. We fine-tune the pruned model separately on the training sets of ChartQA, TextVQA, and ScienceQA, then evaluate on their respective test or validation splits. As shown in Table 8, domain-specific fine-tuning on ChartQA and TextVQA underperforms our 1% FineVision setting (81.7% vs. 86.5% on $\text{ChartQA}_{\text{Test}}$, and 82.9% vs. 89.7% on $\text{TextVQA}_{\text{Val}}$), suggesting that the diverse, multi-task nature of FineVision provides broader representational recovery that benefits even specialized benchmarks. An exception is ScienceQA, where domain-specific training achieves 85.4% compared to 83.5% with FineVision, likely because science reasoning relies on specialized knowledge that benefits from concentrated in-domain exposure. These findings support our default strategy of using a small, diverse fine-tuning corpus, while suggesting that domain-specific data may offer complementary benefits for tasks with highly specialized reasoning requirements.