

GM-Skip: Metric-Guided Transformer Block Skipping for Efficient Vision-Language Models

Supplementary Material

7. Ablation Study

Impact of Forward Transformer Block Deletion: Table 6 show the results of simple classification tasks with the forward deletion strategy. The results demonstrate that the reverse deletion strategy consistently outperforms the forward deletion strategy in balancing performance and target sparsity. For instance, the average accuracy of Reverse (High Perf.) reaches **76.85%**, significantly higher than that of Forward (High Perf.), which averages **66.34%**. In terms of sparsity, Reverse (High Perf.) also achieves a higher average of **39.06%**, compared to **34.77%** for Forward (High Perf.). This performance gap is particularly pronounced in specific categories such as *Bird* and *Sheep*, where the accuracy differences between the two strategies reach **39.30%** and **29.80%**, respectively.

7.1. Captioning Performance under Varying Sparsity

Table 7 compares the performance of different block skipping strategies on the COCO image captioning task using CIDEr as the evaluation metric. The results demonstrate that **GM-Skip** consistently outperforms **Skip-MLLM** in both efficiency and accuracy across a wide range of sparsity levels.

At low sparsity (e.g., 3.13% and 6.25%), GM-Skip achieves CIDEr scores of **110.46** and **109.85**, which are significantly higher than the baseline score of **98.43**, indicating that minor layer removal can even enhance caption quality, potentially due to reduced overfitting or overprocessing. As sparsity increases, GM-Skip maintains strong performance: at 9.38% sparsity, it achieves a peak CIDEr of **111.07**, and even at 21.88%, it still preserves **97.22**, nearly matching the baseline.

In contrast, Skip-MLLM shows substantial degradation as sparsity increases. While its 9.38% configuration achieves a competitive CIDEr of **100.84**, performance drops rapidly at higher sparsity levels (e.g., **43.24** at 31.25%). This is attributed to its fixed-interval removal design, which lacks task-specific feedback and often removes semantically critical layers indiscriminately.

The selected removal lists also reflect the difference in strategy: GM-Skip prefers a greedy selection that avoids early foundational blocks, typically removing mid-to-late layers like 13, 19, 12, and 4. Meanwhile, Skip-MLLM includes blocks such as 7, 10, and 16 early in its removal, which may compromise core representation learning in captioning tasks.

Overall, these results confirm the effectiveness of GM-Skip in maintaining high-level semantic generation quality even under aggressive compression, making it suitable for real-time captioning scenarios.

7.1.1. Trade-off between Captioning Quality and Sparsity:

Table 7 demonstrates the trade-off between captioning performance and model sparsity on the COCO dataset, with CIDEr as the evaluation metric. **GM-Skip** achieves strong captioning quality even under varying levels of block removal, maintaining robustness across a wide sparsity range.

At 9.38% sparsity, GM-Skip reaches a peak CIDEr score of **111.07**, surpassing the full-layer baseline (**98.43**). As sparsity increases, CIDEr scores degrade gradually—remaining at **97.22** under 21.88% sparsity and **89.19** at 25%, indicating stable degradation behavior. In contrast, **Skip-MLLM** suffers significant performance drops when sparsity exceeds 15%, with CIDEr falling to **72.77** at 18.75% and as low as **43.24** at 31.25%.

These results highlight GM-Skip’s ability to balance compression and generation quality. Its metric-guided selection avoids early critical layers and dynamically adapts to semantic sensitivity in the captioning task, providing a controllable mechanism to trade inference efficiency for output fidelity.

7.2. Multi-Object Detection Performance under Varying Sparsity

Table 8 presents the results of different block skipping strategies on the COCO multi-object detection task. **GM-Skip** demonstrates clear advantages over both the baseline and **Skip-MLLM** in terms of accuracy, sparsity, and latency.

At 12.50% sparsity, GM-Skip reaches its best performance with an accuracy of **66.01**, surpassing the full-layer baseline (**60.27**) while also reducing latency from **1.0218s** to **0.7781s**. Even as sparsity increases to 25.00%, accuracy remains competitive (**60.09**) and latency drops further to **0.6881s**. At the highest evaluated sparsity level (31.25%), GM-Skip still retains **59.44** accuracy, comparable to the baseline but with **0.6708s** latency.

In contrast, **Skip-MLLM** shows much steeper accuracy degradation under higher sparsity. While its 9.38% configuration yields 53.41 accuracy, this drops to **34.51** at 31.25% sparsity, indicating the fragility of fixed-interval deletion in more complex multi-object settings.

The selected removal lists also differ noticeably. GM-Skip consistently avoids early blocks and prefers removing mid-to-deep layers like 29, 24, 8, and 15. This confirms that GM-Skip’s greedy selection is more aligned with maintaining spatially distributed object features critical to multi-object detection.

These results highlight GM-Skip’s robustness and adaptability in handling more complex visual grounding scenarios while maintaining efficiency.

7.3. Cross-domain validation.

Using COCO-derived skip configurations, GM-Skip transfers well to CODA, improving person and car accuracy over the full-layer baseline. This demonstrates that GM-Skip captures domain-robust structural redundancy rather than dataset-specific heuristics.

7.4. Beam-Search Variant vs. Greedy Selection

To further validate the efficiency of our greedy block-selection strategy, we additionally evaluated a beam-search variant with a beam width of 3. As shown in our experiments, the beam-search approach requires approximately $3\times$ the calibration cost of the greedy method, yet provides **less than 1%** performance improvement across captioning, multi-object detection, and single-object classification tasks.

Given the marginal gains relative to the significantly higher computational overhead, these results confirm that greedy selection is substantially more cost-effective than pursuing global optimality via beam-search. For completeness and reproducibility, we include the implementation of the global-optimal (beam-search) variant in our code.

8. Visualization of Block Removal

Table 9 presents the detailed removal orders of Transformer blocks across eight COCO categories under different skipping strategies. The results illustrate how GM-Skip dynamically adapts its block selection based on both task feedback and category semantics. For instance, in the “person” class, the removal sets of GM-Skip (High Perf.) and (High Sps.) share a common early deletion pattern involving blocks 4, 24, and 12, suggesting their relatively low contribution to person-level recognition. However, under the High Sps. setting, GM-Skip continues to remove deeper blocks such as 26 and 29, pushing for more aggressive sparsity. This progressive expansion reflects the model’s ability to balance between efficiency and accuracy depending on the target configuration.

In contrast, Skip-MLLM applies a fixed-interval-like deletion across all categories, removing blocks at positions 4, 7, 10, 13, and 16 regardless of semantic context. While simple and generalizable, this approach ignores the task-specific importance of each layer, likely leading to unneces-

sary degradation in some categories (e.g., “bird” or “boat”), where fine-grained visual features may reside in mid-depth layers. Notably, the “chair” and “cup” classes—both involving relatively small or static objects—exhibit more variation in block selection, particularly under high sparsity settings. GM-Skip (High Sps.) aggressively prunes blocks such as 8, 22, and 25, while still maintaining task-relevant performance, implying that VLMs tend to over-process in these simpler visual contexts. Across categories such as “boat”, GM-Skip tends to retain early blocks and skip more towards the tail, further reinforcing the effectiveness of reverse-order removal. Similarly, in the “bird” and “cow” classes, deeper blocks such as 28, 29, and 31 are frequently removed, suggesting that late-stage representations contribute less to category-level decisions in these cases. These patterns highlight GM-Skip’s adaptive nature—removing blocks that are semantically redundant in one class while preserving them in another. The variation in selected blocks across object types also supports the need for per-task or per-class skipping strategies rather than fixed pruning schemes.

Forward vs. Reverse Selection Process

Table 9 further highlights the contrast between forward and reverse block selection strategies in the single-object classification setting. Under the Forward (High Perf.) and Forward (High Sps.) configurations, block removal typically starts from the earlier layers—for example, blocks such as 4, 6, 8, and 12 are commonly removed across categories like person, bird, and boat. While this can yield high sparsity, it also increases the risk of disrupting low-level visual grounding, especially when deeper semantic reasoning relies on early-layer features.

In contrast, GM-Skip adopts a greedy strategy that favors reverse-order deletion, i.e., removing deeper blocks first. This behavior is evident in categories such as boat, sheep, and cow, where blocks like 29, 30, and 31 are consistently removed in both High Perf. and High Sps. settings. This depth-aware pattern helps preserve foundational early-layer representations, thereby maintaining better stability and overall task performance—even under aggressive sparsity constraints.

Additionally, while both forward and reverse strategies share some overlap in block indices, GM-Skip exhibits more task- and class-specific flexibility, adapting its removal decisions based on performance feedback rather than fixed layer positions. This adaptability is especially useful in heterogeneous categories, where semantic features may distribute differently across the network depth.

8.1. Skipped Blocks

Figures 7 visualize the block selection process under both **Forward** and **Reverse** deletion strategies over multiple it-

erative steps. At each step, we simulate the removal of each remaining block, evaluate the task-specific metric (e.g., accuracy), and greedily remove the block that causes the minimal performance degradation.

In the **Forward** strategy, deletion starts from the shallowest (early) blocks and proceeds layer by layer toward deeper ones. As seen in the visual logs, accuracy often drops sharply in the early steps, confirming that low-level layers are crucial for semantic grounding. By Step 3, multiple configurations already reach an accuracy of zero, highlighting the fragility of forward removal in early layers.

Conversely, the **Reverse** strategy begins by attempting to remove the deepest blocks first. Across the same steps, we observe that model performance remains much more stable, with accuracy staying near 1.0 for most steps. This suggests that deeper blocks are more redundant and safer to remove, supporting our design choice of depth-aware, reverse-preferred deletion.

Each screenshot displays the full simulation output at a given step: the tested block removal configurations, their associated accuracy scores, and the final block selected for removal (highlighted in yellow). This process is repeated iteratively to build the full skip configuration.

Together, these visualizations provide transparent insight into the dynamic block selection mechanism used in GM-Skip and support our core hypothesis that layer position significantly influences skipping safety.

9. Training Details and Runtime Overhead

Training requires:

- **3 hours** for captioning;
- **4 hours** for multi-object detection;
- **45 minutes** for single-object classification.

All calibration and analysis in this work were performed on a single NVIDIA A100 (80GB) GPU under an offline setting. The offline calibration stage is used only to determine task-specific block-removal configurations; no fine-tuning or parameter updates are involved.

During online inference, GM-Skip operates purely through a lightweight configuration file that specifies which Transformer blocks to bypass. Importantly, executing a skipped block does not require re-routing through any auxiliary head or additional computation. Instead, the system performs a simple logical check to determine whether a block should be executed or skipped. This conditional control introduces **negligible runtime overhead**—effectively **near-zero additional latency** compared with the original forward pass.

In summary, all computational cost is concentrated in the offline calibration stage, while the deployed skipping mechanism functions with virtually no online cost, enabling efficient real-time inference.

Table 6. Comparison of block skipping strategies on Single-object classification.

Order	Person			Car			Chair			Cup		
	Acc.	Sps.	Lat.	Acc.	Sps.	Lat.	Acc.	Sps.	Lat.	Acc.	Sps.	Lat.
GM-Skip (High Perf.)	87.30	40.63	0.1062	95.20	34.38	0.1277	98.40	40.63	0.1064	68.70	37.50	0.1066
GM-Skip (High Sps.)	37.30	56.25	0.0771	67.00	59.38	0.0889	88.20	62.50	0.0758	31.00	78.13	0.0528
Forward (High Perf.)	83.80	37.50	0.1256	94.30	28.13	0.1373	96.00	31.25	0.1194	60.90	28.13	0.1187
Forward (High Sps.)	30.60	59.38	0.0737	63.10	37.50	0.1041	66.20	53.13	0.0897	21.10	81.25	0.0488
Order	Bird			Boat			Sheep			Cow		
	Acc.	Sps.	Lat.	Acc.	Sps.	Lat.	Acc.	Sps.	Lat.	Acc.	Sps.	Lat.
GM-Skip (High Perf.)	94.30	43.75	0.1079	50.50	46.88	0.1058	51.30	34.38	0.1225	69.10	34.38	0.1020
GM-Skip (High Sps.)	50.60	53.13	0.0936	26.60	87.50	0.0436	15.80	71.88	0.0663	59.70	40.63	0.1157
Forward (High Perf.)	55.00	25.00	0.1352	50.00	50.00	0.1016	21.50	40.63	0.1134	69.20	37.50	0.1204
Forward (High Sps.)	47.10	81.25	0.0520	14.50	87.50	0.0436	0.50	56.25	0.0909	61.80	50.00	0.1235

Table 7. Comparison of block skipping strategies on COCO Captioning.

Method	CIDEr↑	Sparsity (%)	Latency (s)	Interval	Selected Removal List
Baseline	98.43	0.00	0.6546	N/A	[]
GM-Skip	110.46	3.13	0.5574	N/A	[13]
	109.85	6.25	0.5304	N/A	[13, 19]
	111.07	9.38	0.5029	N/A	[13, 19, 12]
	107.82	12.50	0.5119	N/A	[13, 19, 12, 4]
	104.61	15.63	0.4644	N/A	[13, 19, 12, 4, 10]
	102.38	18.75	0.4474	N/A	[13, 19, 12, 4, 10, 23]
	97.22	21.88	0.4413	N/A	[13, 19, 12, 4, 10, 23, 21]
	89.19	25.00	0.4186	N/A	[13, 19, 12, 4, 10, 23, 21, 27]
Skip-MLLM	93.19	3.13	0.6347	28	[4]
	84.69	6.25	0.6402	14	[4, 18]
	100.84	9.38	0.5490	10	[4, 14, 24]
	76.24	12.50	0.6010	7	[4, 11, 18, 25]
	97.09	15.63	0.5087	6	[4, 10, 16, 22, 28]
	72.77	18.75	0.5543	5	[4, 9, 14, 19, 24, 29]
	86.76	21.88	0.4418	4	[4, 8, 12, 16, 20, 24, 28]
	43.24	31.25	0.3821	3	[4, 7, 10, 13, 16, 19, 22, 25, 28, 31]

Table 8. Comparison of block skipping strategies on COCO Detection - Multi.

Method	Acc↑	Sparsity (%)	Latency (s)	Interval	Selected Removal List
Baseline	60.27	0.00	1.0218	N/A	[]
GM-Skip	63.41	3.13	0.8022	N/A	[29]
	63.29	6.25	0.7904	N/A	[29, 24]
	63.53	9.38	0.7672	N/A	[29, 24, 8]
	66.01	12.50	0.7781	N/A	[29, 24, 8, 15]
	64.02	15.63	0.7918	N/A	[29, 24, 8, 15, 11]
	63.44	18.75	0.7491	N/A	[29, 24, 8, 15, 11, 17]
	62.03	21.88	0.7363	N/A	[29, 24, 8, 15, 11, 17, 27]
	60.09	25.00	0.6881	N/A	[29, 24, 8, 15, 11, 17, 27, 28]
	57.81	28.13	0.6734	N/A	[29, 24, 8, 15, 11, 17, 27, 28, 14]
	59.44	31.25	0.6708	N/A	[29, 24, 8, 15, 11, 17, 27, 28, 14, 7]
Skip-MLLM	57.25	3.13	0.9032	28	[4]
	53.17	6.25	0.9032	14	[4, 18]
	53.41	9.38	0.9121	10	[4, 14, 24]
	51.06	12.50	0.8848	7	[4, 11, 18, 25]
	53.20	15.63	0.8061	6	[4, 10, 16, 22, 28]
	46.82	18.75	0.8193	5	[4, 9, 14, 19, 24, 29]
	48.42	21.88	0.8114	4	[4, 8, 12, 16, 20, 24, 28]
	34.51	31.25	0.7000	3	[4, 7, 10, 13, 16, 19, 22, 25, 28, 31]

Table 9. Removal list of Transformer blocks per class under different strategies.

Removal List	person	car	chair	cup
Baseline(Full-layer)	[]	[]	[]	[]
Forward (High Perf.)	[4, 20, 12, 15, 8, 2, 21, 24, 18, 9, 27, 30]	[24, 21, 9, 20, 27, 11, 2, 16, 14]	[31, 14, 6, 16, 4, 8, 30, 5, 11, 22]	[30, 28, 19, 14, 24, 5, 6, 4, 29]
GM-Skip (High Perf.)	[4, 24, 12, 15, 8, 21, 2, 20, 18, 27, 19, 28, 25]	[24, 21, 9, 20, 27, 11, 16, 13, 8, 15, 28]	[31, 14, 6, 16, 30, 21, 24, 28, 2, 8, 7, 18, 26]	[30, 28, 19, 14, 29, 8, 26, 13, 21, 22, 5, 25]
Forward (High Sps.)	[4, 20, 12, 15, 8, 2, 21, 24, 18, 9, 27, 30, 19, 25, 28, 7, 26, 23, 31]	[24, 21, 9, 20, 27, 11, 2, 16, 14, 25, 8, 29, 12, 17, 19, 10]	[31, 14, 6, 16, 4, 8, 30, 5, 11, 22, 25, 29, 21, 24, 28, 19, 12]	[30, 28, 19, 14, 24, 5, 6, 4, 29, 26, 22, 21, 25, 27, 16, 31, 15, 11, 23, 7, 18, 17, 12, 2, 8, 3]
GM-Skip (High Sps.)	[4, 24, 12, 15, 8, 21, 2, 20, 18, 27, 19, 28, 25, 26, 29, 23, 9, 7]	[24, 21, 9, 20, 27, 11, 16, 13, 8, 15, 28, 2, 10, 25, 18, 7, 23, 31, 30]	[31, 14, 6, 16, 30, 21, 24, 28, 2, 8, 7, 18, 26, 29, 19, 11, 25, 20, 27, 13]	[30, 28, 19, 14, 29, 8, 26, 13, 21, 22, 5, 25, 3, 27, 9, 31, 20, 15, 24, 18, 10, 23, 7, 11, 2]
Skip-MLLM	[4, 7, 10, 13, 16, 19, 22, 25, 28, 31]	[4, 7, 10, 13, 16, 19, 22, 25, 28, 31]	[4, 7, 10, 13, 16, 19, 22, 25, 28, 31]	[4, 7, 10, 13, 16, 19, 22, 25, 28, 31]
Removal List	bird	boat	sheep	cow
Baseline(Full-layer)	[]	[]	[]	[]
Forward (High Perf.)	[6, 21, 19, 28, 29, 13, 2, 3]	[15, 24, 8, 4, 28, 31, 30, 12, 25, 19, 7, 29, 14, 23, 21, 26]	[2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 28, 19]	[10, 27, 8, 18, 7, 5, 17, 22, 24, 11, 13, 23]
GM-Skip (High Perf.)	[29, 28, 13, 24, 22, 26, 25, 17, 6, 2, 27, 18, 11, 10]	[29, 24, 28, 31, 30, 19, 25, 6, 26, 12, 23, 27, 14, 3, 21]	[28, 30, 31, 29, 27, 26, 23, 19, 18, 25, 16]	[10, 27, 24, 23, 17, 18, 5, 20, 13, 16, 9]
Forward (High Sps.)	[6, 21, 19, 28, 29, 13, 2, 3, 24, 17, 27, 14, 11, 23, 25, 31, 30, 9, 7, 26, 22, 15, 18, 10, 16, 8]	[15, 24, 8, 4, 28, 31, 30, 12, 25, 19, 7, 29, 14, 23, 21, 26, 27, 11, 18, 10, 5, 9, 6, 22, 17, 16, 13, 2]	[2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 28, 19, 25, 10, 27, 26, 29]	[10, 27, 8, 18, 7, 5, 17, 22, 24, 11, 13, 23, 12, 19, 26, 20]
GM-Skip (High Sps.)	[29, 28, 13, 24, 22, 26, 25, 17, 6, 2, 27, 18, 11, 10, 15, 23, 21]	[29, 24, 28, 31, 30, 19, 25, 6, 26, 12, 23, 27, 14, 3, 21, 13, 15, 2, 11, 17, 18, 16, 8, 22, 7, 5, 9, 4]	[28, 30, 31, 29, 27, 26, 23, 19, 18, 25, 16, 7, 12, 11, 10, 24, 9, 8, 22, 4, 21, 3, 13]	[10, 27, 24, 23, 17, 18, 5, 20, 13, 16, 9, 12, 21]
Skip-MLLM	[4, 7, 10, 13, 16, 19, 22, 25, 28, 31]	[4, 7, 10, 13, 16, 19, 22, 25, 28, 31]	[4, 7, 10, 13, 16, 19, 22, 25, 28, 31]	[4, 7, 10, 13, 16, 19, 22, 25, 28, 31]

Forward Block Deletion - Step 5

Reverse Block Deletion - Step 5

Forward Block Deletion - Step 6

Reverse Block Deletion - Step 6

Forward Block Deletion - Step 7

Reverse Block Deletion - Step 7

Forward Block Deletion - Step 8

Reverse Block Deletion - Step 8


```
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 0] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 1] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 5] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 6] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 10] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 12] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 13] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 14] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 15] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 16] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 18] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 20] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 21] → Accuracy: 1.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 22] → Accuracy: 1.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 23] → Accuracy: 1.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 24] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 25] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 26] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 27] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 28] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 29] → Accuracy: 0.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 30] → Accuracy: 1.00
Test removal [2, 4, 3, 7, 9, 17, 18, 11, 31] → Accuracy: 0.00
Selected blocks: 20, Accuracy: 1.00
Current removal List: [[2], [3], [7], [9], [17], [18], [11]]
```

Forward Block Deletion - Step 9

Reverse Block Deletion - Step 9

[illegible]

Forward Block Deletion - Step 10

Reverse Block Deletion - Step 10

[illegible]

Forward Block Deletion - Step 11

Reverse Block Deletion - Step 11

Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 0] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 0] → Accuracy: 8.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 1] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 1] → Accuracy: 8.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 5] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 5] → Accuracy: 30.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 6] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 6] → Accuracy: 30.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 7] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 7] → Accuracy: 34.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 10] → Accuracy: 1.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 10] → Accuracy: 37.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 12] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 12] → Accuracy: 30.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 13] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 13] → Accuracy: 30.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 14] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 14] → Accuracy: 32.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 15] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 15] → Accuracy: 26.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 16] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 16] → Accuracy: 30.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 19] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 19] → Accuracy: 37.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 21] → Accuracy: 2.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 21] → Accuracy: 30.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 22] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 22] → Accuracy: 30.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 23] → Accuracy: 4.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 23] → Accuracy: 8.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 25] → Accuracy: 4.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 25] → Accuracy: 27.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 26] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 26] → Accuracy: 8.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 28] → Accuracy: 7.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 28] → Accuracy: 34.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 29] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 29] → Accuracy: 30.00
Test removal [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 30] → Accuracy: 8.00	Test removal [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 30] → Accuracy: 30.00
Selected block: [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 7, 0]	Selected block: [30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 7]
Current removal list: [2, 4, 3, 7, 8, 17, 18, 11, 20, 22, 24, 7, 0]	Current removal list: [20, 30, 31, 29, 27, 26, 22, 19, 18, 25, 16, 7]

Forward Block Deletion - Step 12

Reverse Block Deletion - Step 12

Figure 7. Visualization of selected Transformer blocks on the COCO sheep class.