Turbo: Informativity-Driven Acceleration Plug-In for Vision-Language Models

Anonymous ECCV 2024 Submission

Paper ID #06301

In the supplementary material, we first provide more details about our Turbo plug-in. Then more ablation studies on different fusion strategies are provided. Then we offer the theoretical demonstration of the proposed proposition in Section 3.1. Finally we demonstrate some visualization results to justify our arguments, including the generated images from stable diffusion and intermediate merging process of understanding tasks.

1 Implementation Details

1.1 Turbo Architecture for Understanding Tasks

Turbo could be easily plugged in almost any pre-trained, attention-based VLMs to reduce the total sequence length block by block, with no need for further training or adjustment. In practice, we replace all the attention blocks by Turbo. For Turbo, we merge tokens progressively based on their information degree:

$$\mathcal{E} = \mathcal{R} - \alpha\mathcal{A}, \quad \mathcal{E} = \mathcal{R}/\mathcal{A}. \tag{1}$$

Information degree takes both mutual redundancy and semantic values into account, encouraging insignificant, similar tokens to be merged preferentially. This merging strategy reduces the amount of tokens with duplicated or low informativity, compressing the token sequence with minor information loss. Inspired by [1, 2], we follow the bipartite soft matching to calculate mutual redundancy and apply the merging strategy to aggregate tokens. Specifically, due to the over-parameterized problem for token sequence [4], we leverage keys (K) or queries (Q) in QKV attention and the cosine similarity metric between tokens to measure the similarity between tokens. We define mutual redundancy of a token to be the maximum cosine similarity with the other tokens. By adding the quantity of semantic value, which is the attention proportion of each token, the information degree is obtained.

To avoid excessive computational cost for calculating similarity matrix of the whole token sequence, we leverage bipartite soft matching to speed up the merging process. Suppose the drop ratio is Υ, which means we will reduce the amount of tokens by number Υ in each block. In every block, we divide the tokens into two partitions B and C of the same size (if the number of tokens is odd, one partition has 1 more tokens than the other). Then we calculate the mutual redundancy between the two partitions B and C. Specifically, for each token in partition B, we keep the highest cosine similarity with respect to partition

Table 1: Ablation Study on Fusion Strategies. We adopt several fusion strategies and test their performance under different coefficients. Some complex fusion strategies can achieve slightly better results, but in order to keep the form easy to apply, we adopt the simple weighted average in other experiments.

Strategy	α	β	γ	B@4	CIDEr	Throughput
S1	6	-	-	38.2	130.0	67.6
S2	-	-	-	38.2	129.9	62.8
S3	1	-	-	37.8	128.5	61.9
S3	2	-	-	37.9	128.4	61.9
S3	3	-	-	36.9	125.7	61.9
S3	4	-	-	36.1	123.1	61.9
S4	6	0.9	1	38.2	129.5	67.4
S4	6	0.9	2	38.3	129.5	67.4
S4	6	0.9	3	38.1	129.5	67.4
S4	6	0.9	4	38.2	129.5	67.4
S4	6	0.9	5	38.1	129.7	67.4
S4	6	0.9	6	38.0	129.2	67.4
S4	6	0.9	7	38.0	129.5	67.4
S4	6	0.9	8	38.4	130.2	67.4
S4	6	0.9	9	38.2	129.9	67.4
S4	6	0.9	10	38.2	129.7	67.4
S4	6	1.1	1	38.3	130.0	67.4
S4	6	1.1	2	38.3	130.3	67.4
S4	6	1.1	3	38.4	130.1	67.4
S4	6	1.1	4	38.3	129.9	67.4
S4	6	1.1	5	38.5	129.9	67.4
S4	6	1.1	6	38.1	129.3	67.4
S4	6	1.1	7	38.2	129.6	67.4
S4	6	1.1	8	38.5	130.3	67.4
S4	6	1.1	9	38.2	129.8	67.4
S4	6	1.1	10	38.2	129.6	67.4

C as its mutual redundancy. After adding the semantic value of each token on partition B, we sort the information degree of B and merge the top Υ tokens into C, by averaging merging the Υ tokens in B into the corresponding tokens in C with the highest cosine similarity. At last we concatenate the sequence back to continue the forwarding process. In this way, we reduce the length of token sequence by Υ in each block after the attention layer and before the MLP layer. Notice that We call Υ the drop ratio, but it is in fact the number of tokens we reduce every attention block, which is a real ratio by dividing the length of the token sequence. We also note that the semantic value are naturally contained in uni-modal cls self-attention map or cross-modal cls cross-attention map depending on the model structure, so we don't need to add additional calculation for the semantic value.

When merging process is finished, some tokens can represent several different patches. This can change the outcome of softmax attention and thus influence

the attention calculation. So we fix this with a minor modification:

$$A = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D}} + \log s\right), \tag{2}$$

where s is the number of patches/tokens represented by the merged tokens all along the merging procedure.

In order to avoid excessive merging, *i.e.*, merging too many tokens and leave only a few tokens in final blocks. This will cause insufficient expression ability problem and we observe a sharp performance drop at certain drop ratios. So we set up a threshold r for the least number of tokens in the final stage, to mitigate the dramatic drop on large Υ. Table 2 shows the effectiveness of this restriction for preventing a sudden performance decline.

1.2 Turbo Architecture for Generation Tasks

For generative tasks, Stable Diffusion [5] is one popular backbone. Here, Turbo contains one merging module and one inverse-sampling module. For the merging module, we attach Turbo acceleration on UNet of Stable Diffusion, as UNet consumes the most computation. More specifically, UNet usually consists of three key components: self-attention, cross-attention and FFN. We add Turbo merging/restoring before/ after each component. For self-attention and FFN, Turbo merging is calculated by one visual modality; while for cross-attention, Turbo merging is calculated by visual-textual modalities. We evaluate by generating 2000 images, each resolution is $512 * 512$. The text classes used are from ImageNet-1k. We use FID scores to metric the generation quality.

2 More Experiments

2.1 Ablation Study on Fusion Strategy

We design four types of fusion strategies to combine mutual redundancy ($\mathcal{R}$) with semantic value ($\mathcal{A}$):

$$\text{S1} : \mathcal{E} = \mathcal{R} - \alpha\mathcal{A}, \tag{3}$$

$$\text{S2} : \mathcal{E} = \mathcal{R}/\mathcal{A}, \tag{4}$$

$$\text{S3} : \mathcal{E} = \mathcal{R}(1 - \alpha\mathcal{A}), \tag{5}$$

$$\text{S4} : \mathcal{E} = \mathcal{R} - \beta^{|\gamma - \text{block\_id}|}\alpha\mathcal{A}, \tag{6}$$

where S4 is designed to allow dynamic scales between $\mathcal{R}$ and $\mathcal{A}$ on different blocks. For example, if $\beta > 1$ and $\gamma = 6$, then the scale of $\mathcal{A}$ will reach its maximum on block 6 and attain the minimum value on two-end blocks.

As shown in table 1, we have done extensive experiments on the four fusion strategies with different coefficients. Though S4 attains the best result, due to its complexity and such slight performance gain, *i.e.*, three hyper-parameters to be determined with a gain of only 0.3 on CIDEr, we thus adopt the simple weighted average fusion strategy (S1) on our turbo module.

Table 2: Ablation Study on Threshold. We validate the effectiveness of threshold to prevent dramatic performance drop on large Υ. With slight decline on speed, model performance with threshold on minimum token length can maintain a smoother decrease when Υ getting larger.

Υ	Threshold	B@4	CIDEr	Throughput
65	0	28.8	95.5	107.6
65	70	31.1	104.3	103.3
65	130	34.2	115.3	100.8
70	0	27.3	89.4	113.0
70	70	31.0	103.7	110.7

2.2 Ablation Study on Threshold

When applying large drop ratios Υ on Turbo module, we witness a sharp drop of model performance. We argue that this phenomenon is due to the insufficient expression ability of token sequence length below a certain threshold. So we append a threshold on minimum number of tokens left in the final block. Specifically, we stop the token merging process once the token sequence length is below the threshold. Results in Table 2 demonstrates the effectiveness of such a threshold on large Υ. By introducing a threshold to large drop ratio, we improve the model performance by over 10 points on CIDEr with slight acceleration declines.

3 Theoretical Interpretation of the Proposition

We here detail the theoretical deduction of proposition in the main paper (Eq. (9)).

Preliminary. We first remind the definitions/propositions concerning open ball, neighborhood and continuity in topology.

Definition 1. Given a metric space (E, d), where E is a set and $d : E \times E \to \mathbb{R}$ is a metric on E, the open ball centered at a point $a \in E$ with radius $\epsilon > 0$ is defined as the set of all points in E whose distance to a is less than ϵ. Mathematically, this is expressed as:

$$B(a, \epsilon) = \{x \in E : d(a, x) < \epsilon\} \tag{7}$$

Definition 2. Let (E, d) be a metric space, $a \in E$, we say that V is a neighborhood of a in E, and we write $V = \mathcal{V}(a)$ if there exists $\eta > 0$ such that $B(a, \eta) \subseteq V$.

Proposition 8. Suppose $f : (E, d) \to (E', d')$, then

$$[f \text{ continue on point } a \in E] \iff [\forall V \in \mathcal{V}(f(a)),\ f^{-1}(V) \in \mathcal{V}(a)] \tag{8}$$

Demonstration. We define $y_1 \in R^n$ as a semantic-rich vector if y_1 is in the set of all possible *cls* tokens Λ. Inspired by the success of quantization methods [3,6], y_1 can be replaced by the most similar discrete vector in the code-book

Fig. 1: Visualizations of Text-To-Image Generation. Left is no acceleration, and Right is acceleration by our Turbo module.

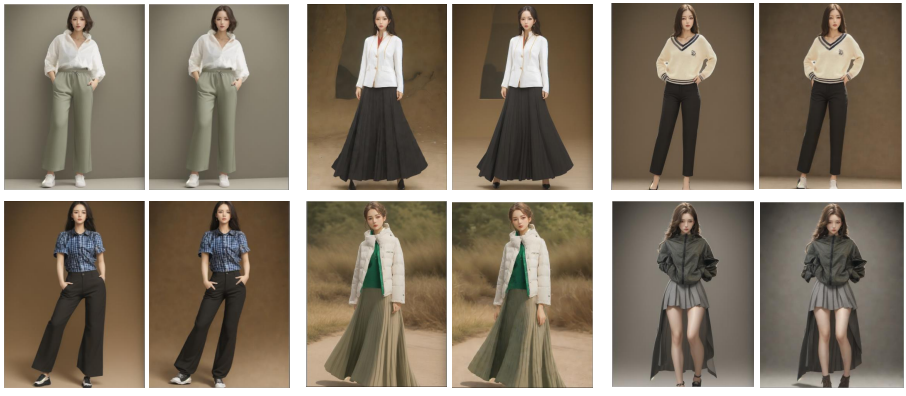

Fig. 2: Visualizations of Image-To-Image Generation. Left is no acceleration, and Right is acceleration by Turbo. Generally speaking, the generation quality is very similar before and after acceleration.

without losing its informativity. Based on this observation, we make an assumption for the local continuity for informativity $\mathcal{I}$ on y_1, which can be formulated mathematically as follows.

Assumption. Under the metric spaces (R^n, L_2) and (R, L_1), $\forall y_1 \in \Lambda$, $\mathcal{I} : (R^n, L_2) \to (R, L_1)$ is continue on y_1.

According to Proposition 8 and Definition 2, for all $\eta_0 \in R_+$, $B(\mathcal{I}(y_1), \eta_0) \in \mathcal{V}(\mathcal{I}(y_1))$, so $\mathcal{I}^{-1}(B(\mathcal{I}(y_1), \eta_0)) \in \mathcal{V}(y_1)$, therefore there exists $\epsilon > 0$ such that $B(y_1, \epsilon) \subseteq \mathcal{I}^{-1}(B(\mathcal{I}(y_1), \eta_0))$. By mapping the open ball $B(y_1, \epsilon)$ back using $\mathcal{I}$, we can find $\eta \leq \eta_0$ such that $\mathcal{I}(B(y_1, \epsilon)) \subseteq B(\mathcal{I}(y_1), \eta)$. The operator $\leq$ can be replaced by $<$ by choosing small ϵ, proving the proposition in the main paper.

4 Visualization Results

To intuitively demonstrate the superiority of our Turbo module, we here visualizes results from generative tasks. Figure 1 shows the results of text-to-image generation. The used text prompts are usually as: One female model wearing purple long sleeves and blue jeans stands on the coast. Although Stable Diffusion

still needs improvement in generation details (such as hands), Turbo acceleration has almost no side effects. Figure 2 shows the results of image-to-image generation. Generally speaking, the conclusion is similar: the generation quality is close before and after Turbo acceleration.

5 Limitations

Although our Turbo can accelerate VLMs without retraining, it slightly lowers the model performance, which may affect the performance on fine-grained scenarios like OCR. Besides, the drop ratio Υ is set the same on all batches for simplicity, and more complex designs can be investigated in the future. In this paper, we focus on exploring the data-perspective acceleration for VLMs, being orthogonal to existing model-perspective acceleration.

References

1. Ahmed, F., Dickerson, J.P., Fuge, M.: Diverse weighted bipartite b-matching. arXiv preprint arXiv:1702.07134 (2017) 1
2. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461 (2022) 1
3. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021) 4
4. Marin, D., Chang, J.H.R., Ranjan, A., Prabhu, A., Rastegari, M., Tuzel, O.: Token pooling in vision transformers. arXiv preprint arXiv:2110.03860 (2021) 1
5. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022) 3
6. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems **30** (2017) 4