TWEO: Transformers Without Extreme Outliers Enables FP8 Training And Quantization For Dummies

Supplementary Material

A. Empirical Analysis of Data-Independent Outliers

To provide a solid empirical foundation for the *Contradiction Stethoscope* presented in the main text (Sec. 1 of the main paper), i.e., that extreme outliers are *data-independent, mechanically-produced artifacts* of training, this appendix presents detailed ablation studies. Concretely, the stethoscope reports layer-wise Avg Max, Avg Min, and Mean-Abs statistics, averaged over the batch.

A.1. Ablation Study on Qwen2.5-0.5B (Language Model)

For the language model, we designed four experimental combinations (see Figure 1) to decouple the influence of pre-trained weights, the embedding layer, and the input data on outlier generation. The results clearly demonstrate:

- *Product of Training.* Comparing (a) and (c), extreme outliers are only absent (Avg Max $<$ 10) when both the model weights and the embedding layer are randomly initialized. The moment pre-trained weights are used, outliers emerge immediately.
- *Product of Weights.* Comparing (a) and (b), panel (b) shows that even if the embedding layer is re-initialized randomly, as long as the Transformer Block weights are pre-trained, extreme outliers ($\sim$1600) are still mechanically reproduced.
- *Data-Independence.* Comparing (c) and (d), the pre-trained model produces outliers of the same magnitude regardless of whether the input is real data or random tokens.

Conclusion. Extreme outliers are a *mechanical product* of the pre-trained Transformer Block weights and are independent of both the embedding layer and the input data.

A.2. Statistical Evidence on Data Independence

To further substantiate the data-independent nature of extreme outliers, we extend the analysis to Qwen2.5-0.5B by comparing real text inputs against random token inputs. We collect statistics from the per-sample maximum activation over $N = 10000$ samples. The results show that extreme outliers emerge consistently under both settings, indicating that the phenomenon persists even when semantic structure is removed from the input.

As shown in Table 1, Qwen2.5-0.5B produces extreme outliers far beyond the FP8 range under both real text and random token inputs. The comparable maxima and average maxima indicate that semantic structure is not required for the emergence of these outliers. This statistical evidence further supports our claim that extreme outliers are mechanically produced by the trained weight structure, rather than being triggered by specific data patterns.

Table 1. Statistical comparison of extreme outliers on Qwen2.5-0.5B under real text and random token inputs. Statistics are computed from the per-sample maximum activation over $N = 10000$ samples.

Input Type	Max	Min	Avg Max
Real text	1880.0	1464.0	1690.4
Random tokens	1928.0	1264.0	1622.0

A.3. Ablation Study on ViT-B (Vision Model)

To demonstrate that this finding holds true in the vision modality, we conducted parallel experiments on the ViT-B model. Figure 2 shows the input samples used, including real ImageNet images and randomly generated noise. Figure 3 displays the corresponding activation results.

The conclusions are identical to those from the Qwen model:

- *Product of Training.* Comparing (c) and (d), the randomly initialized ViT-B model (c) produces no extreme outliers (Avg Max $<$ 7).
- *Data-Independence.* The comparison between (d) and (e) proves that the pre-trained ViT-B model consistently reproduces extreme outliers (Avg Max $\sim$1500), regardless of whether the input is real images (Fig. 2(a), (b)) or random noise (Fig. 2(c), (d)).

Conclusion. Our core finding—that outliers are data-independent and a product of training—is cross-modally general.

B. Ablation Study on Key TWEO Hyperparameters

This appendix details the impact of TWEO's two key hyperparameters—the threshold τ and the penalty power p—on model performance, outlier suppression efficacy, and downstream quantization performance.

B.1. Impact of Scaling Factor (τ) on Outlier Suppression

We first investigate the effect of different scaling factor settings (which act as a soft threshold) on the suppression of outliers during training. As shown in Table 2, we fix the

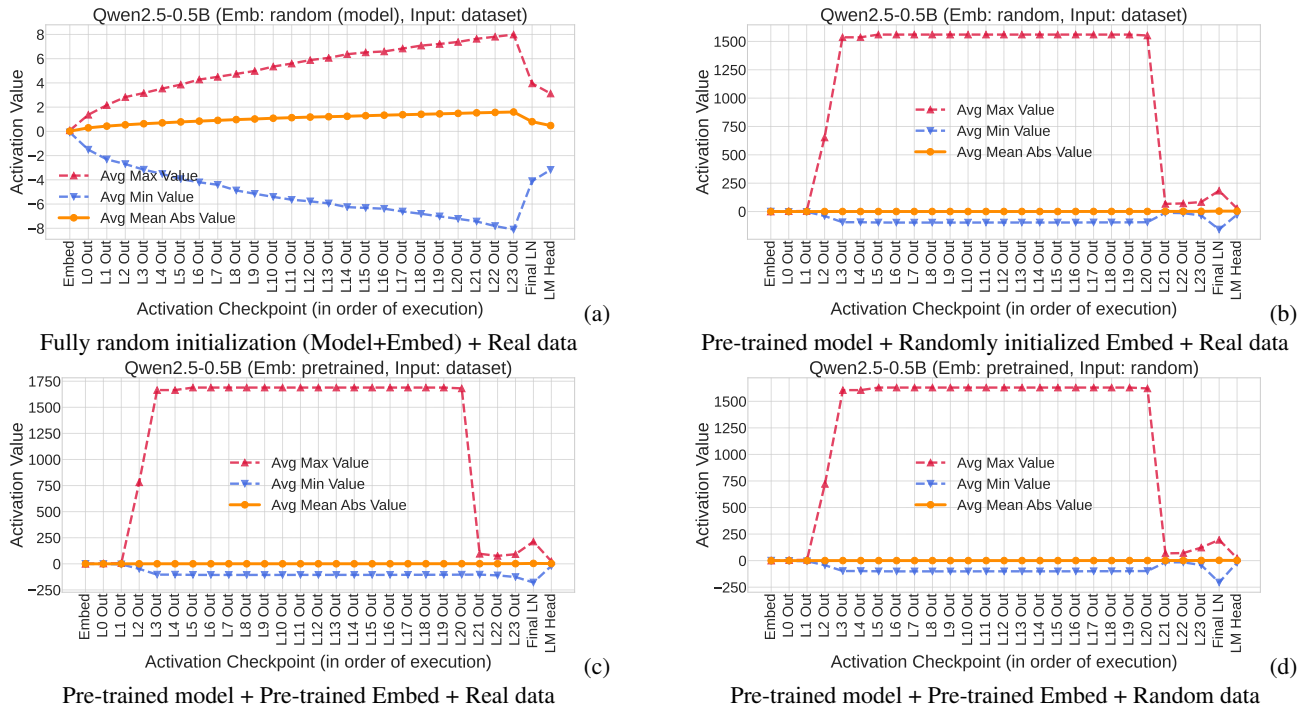


Figure 1. Overview of Qwen2.5-0.5B ablation study. (a) Fully random initialization (Model+Embed) + Real data. (b) Pre-trained model + Randomly initialized Embed + Real data. (c) Pre-trained model + Pre-trained Embed + Real data. (d) Pre-trained model + Pre-trained Embed + Random data.

(a) Real sample 1 (b) Real sample 2 (c) Random sample 1 (d) Random sample 2

Figure 2. ViT-B input samples. The samples are clearly divided into: real ImageNet images ((a), (b)) and randomly generated noise images ((c), (d)). These examples are used to demonstrate the data-independent nature of outlier generation.

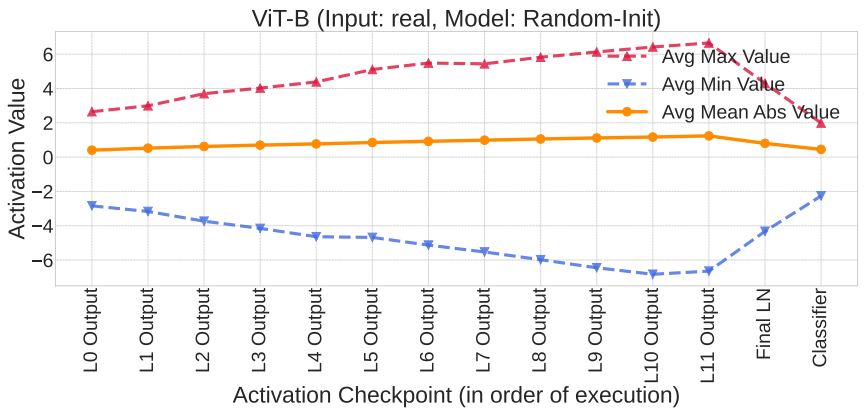
penalty power ($p = 4$, the value used in our main paper) and vary the scaling factor τ.

Table 2 clearly shows that a lower threshold τ results in stronger outlier suppression (i.e., smaller peak magnitudes during and after training). We found that for the GPT2 model, a lower threshold (e.g., $\tau = 1$) with its strong regularization had little impact on model performance (PPL). However, for the Swin Transformer series, an excessively low threshold ($\tau = 1$) slightly impaired model accuracy. Therefore, the experiments reported in our main text use $\tau = 3$ as a balanced choice.

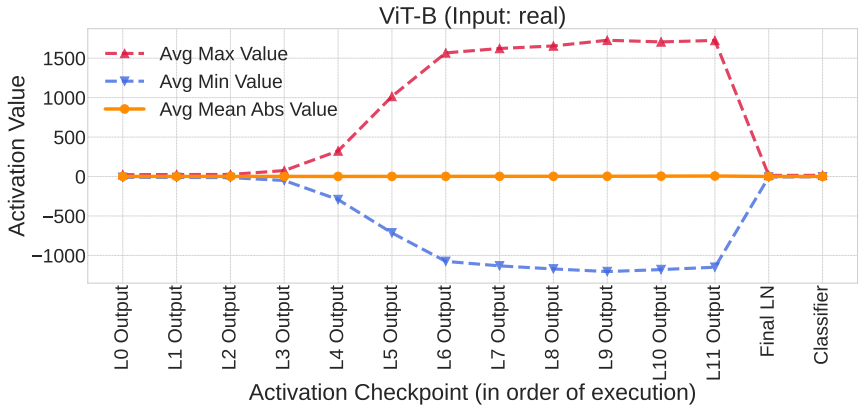
B.2. Impact of Penalty Power (p) on Outlier Suppression

Next, we investigate the impact of the penalty power p on model performance. As shown in Table 3, we fix the threshold ($\tau = 3$, the value used in our main paper) and vary the penalty power p.

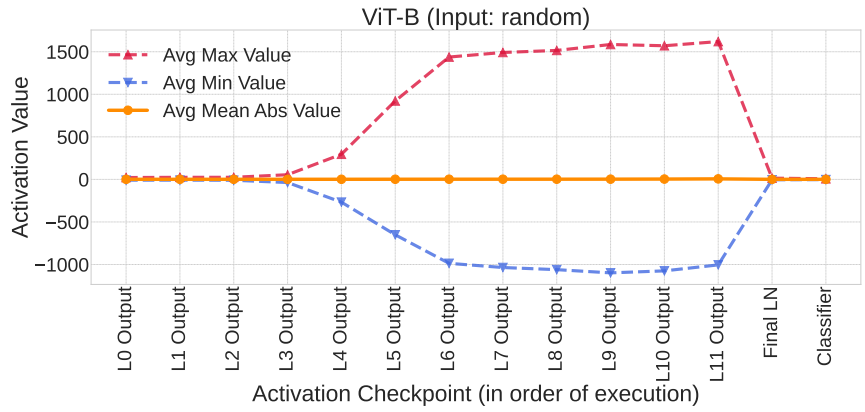
As seen in Table 3, a higher penalty power p leads to better outlier suppression. We found that for GPT2, stronger



(c) Randomly initialized model + Real images



(d) Pre-trained model + Real images (Input a/b)



(e) Pre-trained model + Random noise

Figure 3. Data-independence results for ViT-B. (c) Randomly initialized model + Real images. (d) Pre-trained model + Real images. (e) Pre-trained model + Random noise.

regularization ($p = 5$) did not significantly affect model performance. However, for the Swin series, excessively strong regularization ($p = 5$) slightly impaired performance. Therefore, the experiments reported in our main text uniformly use $p = 4$.

B.3. Combined Impact of Threshold and Power on Quantization Performance

Finally, we conducted a more detailed ablation study on the GPT2 model to investigate the combined impact of differ-

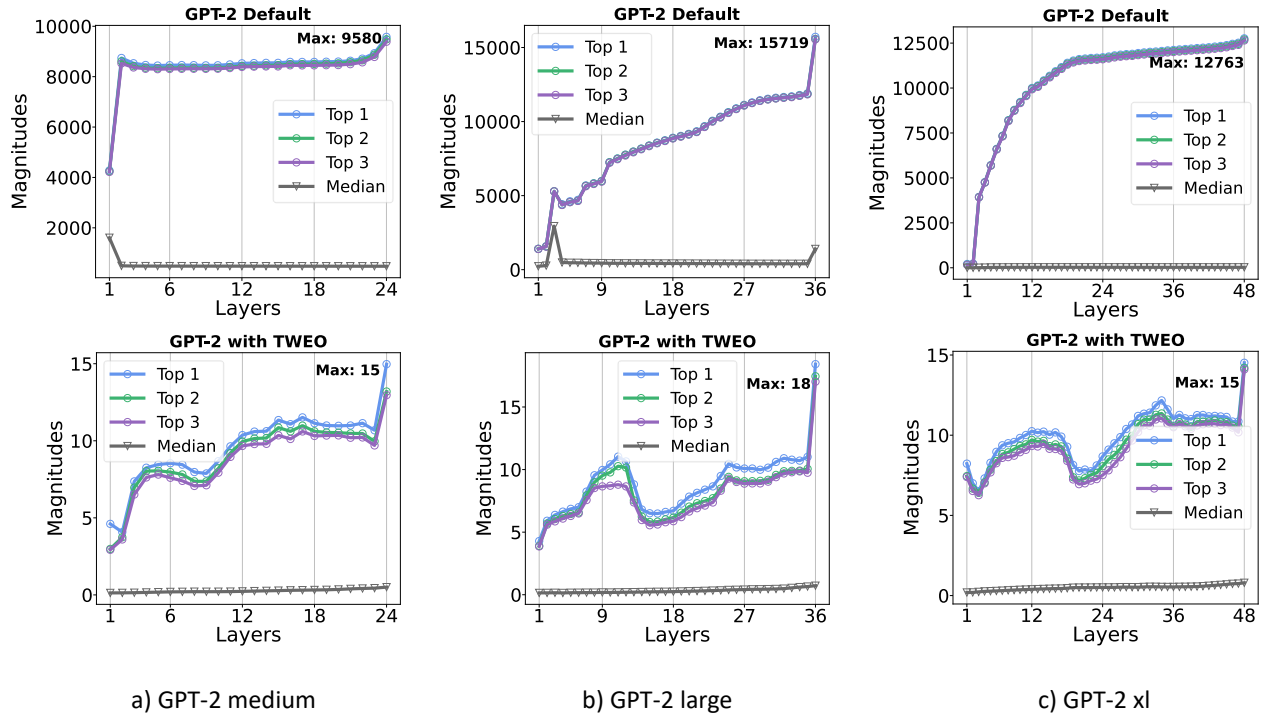


Figure 4. Comparison of activation magnitudes during GPT-2 training. Top 1 refers to the largest activation value in each layer. Top 2, Top 3 refer to the second and third largest. Median is the median of all activations.

Table 2. Ablation study on the effect of different TWEO scaling factors (τ) on outlier suppression.

Model	Threshold (τ)	Accuracy (%)	Peak	Final
	ImageNet Accuracy			
Swin-T †	–	81.2	1556	534
Swin-T	1	80.6	6	6
Swin-T	2	81.3	11	8
Swin-T #	3	81.4	22	15
Swin-T	5	81.4	34	24
	ImageNet Accuracy			
Swin-S †	–	82.7	6402	1758
Swin-S	1	82.5	10	5
Swin-S	2	82.6	15	7
Swin-S #	3	82.8	35	18
Swin-S	5	83.1	48	25
Swin-S	10	83.1	62	31
	Wikitext-2 Perplexity (PPL)			
GPT2 †	–	20.04	823	563
GPT2	1	18.68	6	5
GPT2 #	3	18.68	16	15
GPT2	5	18.89	28	28

† Baseline model trained without TWEO.
# Hyperparameter setting used in the main paper.

Table 3. Impact of different TWEO penalty powers (p) on model performance and outlier suppression.

Model	Power (p)	Accuracy (%)	Peak	Final
	ImageNet Accuracy			
Swin-T †	–	81.2	1556	534
Swin-T	3	81.2	48	24
Swin-T #	4	81.4	22	15
Swin-T	5	81.1	12	10
	ImageNet Accuracy			
Swin-S †	–	82.7	6402	1758
Swin-S	2	82.3	292	127
Swin-S	3	82.4	80	48
Swin-S #	4	82.8	22	10
Swin-S	5	82.7	15	8
	Wikitext-2 Perplexity (PPL)			
GPT2 †	–	20.04	823	563
GPT2	2	18.77	158	113
GPT2 #	4	18.68	16	15
GPT2	5	19.07	10	10

† Baseline model trained without TWEO.
# Hyperparameter setting used in the main paper.

ent threshold and power combinations on downstream 8-bit quantization performance.

The ablation study in Table 4 shows that stronger outlier suppression from TWEO (e.g., $\tau = 1, p = 4$) leads to better downstream quantization performance (especially for per-tensor quantization). It is worth noting that the hyperparameters adopted in our main text ($\tau = 3, p = 4$) are not the optimal choice for downstream quantization (compared to $\tau = 1, p = 4$). This indicates that the choice of TWEO hyperparameters involves a trade-off between model pretraining performance and downstream quantization performance. Future work focused on quantization can leverage our TWEO method with more aggressive (i.e., stronger suppression) hyperparameters, such as $\tau = 1, p = 4$, to achieve superior quantization results.

C. OpenWebText Pre-training Hyperparameters

We pre-trained a series of GPT-2 [3] models from scratch on the OpenWebText dataset. The training used a total batch size simulating 1M tokens and was run for 100k steps, equivalent to 100B tokens of training. The training hyperparameters largely follow those of GPT-3 [1]: for the 3B and 7B models, we adopted configurations similar to GPT-3; the other smaller models (124M to 1.5B) maintain configurations similar to the standard GPT-2. For GPT-2 124M to Large, a 0.5M batch size for 100k steps is equivalent to 50B tokens; for XL, 3B, and 7B, a 1M batch size for 100k steps is equivalent to 100B tokens. Detailed configurations are shown in Table 5.

D. Comparison with SmoothQuant: Quantizing the Residual Stream

As discussed in the main text (Sec. 4.3 of the main paper), a core advantage of the TWEO paradigm is its resolution of the fundamental bottleneck in the old paradigm: the inability to quantize the residual stream ($x = x + f(x)$). In old paradigm models, extreme outliers are passed and accumulated layer by layer in the residual stream x. Consequently, previous methods (like SmoothQuant) are forced to add the quantized $f(x)$ to a high-precision x, incurring significant inference overhead. Due to space constraints, the main text (Table 4 of the main paper) only presented results for GPT-2 XL. This section provides the complete experimental data (Table 6), comparing the full performance of Default and TWEO models when the residual stream is quantized.

Full Analysis The complete results in Table 6 are decisive and validate the claims in the main text:

- *SmoothQuant Fails in a Fair Comparison.* When SmoothQuant is forced to quantize the residual stream (Yes), its performance on all Default models *completely collapses* (e.g., Medium PPL 1510.17, Large PPL 3751.66). This confirms that outliers accumulated in the residual stream are the fundamental obstacle to true full-model quantization.
- *TWEO Makes Residual Quantization Possible.* Conversely, the TWEO models, even when using the simplest AbsMax and quantizing the residual stream, show only a minor drop in PPL. This performance far exceeds the Default + SmoothQuant combination that *does not* quantize the residual.
- *TWEO Renders Complexity Obsolete.* More interestingly, on the outlier-free models created by TWEO, applying the complex SmoothQuant method yields marginal gains compared to simple AbsMax, making the trade-off in complexity unjustifiable. This proves that TWEO has leveled the quantization difficulty of activations and weights, rendering complex difficulty transfer methods like SmoothQuant unnecessary.

E. Training Overhead and FP8 Acceleration Analysis

To evaluate the additional computational overhead introduced by TWEO and its resulting performance advantages, we recorded the training resource consumption for vision and language models under different settings.

As shown in Table 7, the additional training overhead from TWEO is minimal for vision models (Swin Transformer). On Swin-S (50M) and Swin-B (88M) models, the total training time increased by only 3.3% and 3.0%, respectively, while VRAM usage remained unchanged.

E.1. Vision FP8 Training Results

To complement the main-text results on outlier suppression and quantization, we further evaluate whether TWEO enables stable native FP8 training in vision Transformers. We conduct experiments on Swin models under the same native FP8 setup used in our language experiments. The results show that standard FP8 training suffers severe collapse, whereas TWEO stabilizes optimization and recovers near-BF16 accuracy.

As shown in Table 8, native FP8 training without TWEO leads to substantial accuracy degradation on both Swin-T and Swin-B. In contrast, TWEO reduces the peak activation magnitude to a small and stable range, allowing FP8 training to achieve accuracy nearly identical to the BF16 baseline. This result confirms that TWEO is effective not only for language models, but also for vision Transformers under native FP8 training.

Table 4. Combined impact of different TWEO threshold (τ) and power (p) combinations on GPT2 quantization performance (PPL ↓).

			Full Precision	Activation Only 8-bit (A8)	Weight Only 8-bit (W8)	Full 8-bit Quant (W8A8)	
Method	τ	p	(BF16) PPL	per-tensor / per-token	per-tensor / per-channel	W8(T)A8(T) / W8(T)A8(K)	W8(C)A8(T) / W8(C)A8(K)
Default †	-	-	20.04	82.94 / 21.19	20.71 / 20.32	86.60 / 22.02	82.94 / 21.49
TWEO	1	4	18.68	19.64 / 18.81	19.17 / 18.94	20.19 / 19.31	19.93 / 18.81
TWEO #	3	4	18.83	20.21 / 19.00	19.32 / 19.11	20.82 / 19.51	20.54 / 19.00
TWEO	3	2	18.77	137.62 / 19.35	19.73 / 19.32	158.15 / 20.42	151.52 / 19.95
TWEO	3	3	18.74	21.48 / 18.95	19.35 / 19.10	22.37 / 19.57	22.03 / 19.31
TWEO #	3	4	18.83	20.21 / 19.00	19.32 / 19.11	20.82 / 19.51	20.54 / 19.00

† Baseline model trained without TWEO.
# Hyperparameter setting used in the main paper.

Table 5. Detailed OpenWebText pre-training hyperparameters. We set the corresponding number of layers (n_{layers}), model dimension (d_{model}), number of attention heads (n_{head}), head dimension (d_{head}), Batch Size, and Learning Rate for GPT-2 models of different scales (124M to 7B).

Model	**Params**	n_{layers}	d_{model}	n_{head}	d_{head}	**Batch Size**	**Learning Rate**
GPT-2	124M	12	768	12	64	0.5M	6.0e-4
GPT-2 Medium	350M	24	1024	16	64	0.5M	6.0e-4
GPT-2 Large	774M	36	1280	20	64	0.5M	3.0e-4
GPT-2 XL	1.5B	48	1600	25	64	1M	2.0e-4
GPT-2 3B	2.7B	32	2560	32	80	1M	1.6e-4
GPT-2 7B	6.7B	32	4096	32	128	1M	1.2e-4

Table 6. Comparison with SmoothQuant on quantizing the Residual Stream ($x = x + f(x)$). This experiment demonstrates that TWEO enables effective quantization of the residual stream for the first time, whereas old paradigm models collapse completely under this setting.

Model PPL (BF16)	**Method**	**Res.?**	**W8(C)A8(T) / W8(C)A8(K)**
GPT-2-M Default	SmoothQuant	No	19.76 / 17.28
(PPL: 16.77)	SmoothQuant	Yes	1510.17 / 19.23
GPT-2-M TWEO	**AbsMax**	Yes	16.32 / 15.40
(PPL: 15.18)	**SmoothQuant**	Yes	16.09 / 15.34
GPT-2-L Default	SmoothQuant	No	15.70 / 15.06
(PPL: 14.92)	SmoothQuant	Yes	3751.66 / 20.57
GPT-2-L TWEO	**AbsMax**	Yes	18.39 / 14.07
(PPL: 13.79)	**SmoothQuant**	Yes	18.24 / 14.03
GPT-2-XL Default	SmoothQuant	No	14.81 / 14.01
(PPL: 13.84)	SmoothQuant	Yes	1876.70 / 21.93
GPT-2-XL TWEO	**AbsMax**	Yes	**13.06** / 12.63
(PPL: 12.77)	**SmoothQuant**	Yes	**12.89 / 12.51**

W8(C)/A8(T): Weights per-channel, Activations per-tensor
W8(C)/A8(K): Weights per-channel, Activations per-token

Table 7. Training resource consumption comparison for vision models (Swin Transformer) on 4090 GPUs.

Model	**Method**	**VRAM (GB)**	**Total Time (300epoch)**	**8-GPU Time**
Swin-S (50M)	Default	18.0G	240 H	30H
	TWEO	18.0G	248H (+3.3%)	31H
Swin-B (88M)	Default	23.5G	300 H	37.5H
	TWEO	23.5G	309H (+3.0%)	38.6H

Table 8. Vision FP8 training results on ImageNet. Standard native FP8 training collapses, while TWEO recovers near-BF16 performance.

Model	**Top-1 (BF16)**	**Top-1 (FP8)**	**Peak**	**Gap**
Swin-T	81.2	50.2 (*Collapse*)	>1500	-30.9
Swin-T + TWEO	**81.4**	**81.2**	**32**	**-0.2**
Swin-B	83.5	61.9 (*Collapse*)	>7000	-21.6
Swin-B + TWEO	**83.5**	**83.4**	**47**	**-0.1**

E.2. Comparison with a DeepSeek-Style FP8 Strategy

We further compare TWEO with a DeepSeek-style FP8 strategy from the perspective of training efficiency. The DeepSeek-style setup adopts mixed precision and fine-grained per-tile scaling to bypass outlier-sensitive components, whereas TWEO enables a much simpler full-layer per-tensor FP8 configuration by structurally suppressing outliers. For a fair comparison, both methods are implemented with the same Transformer Engine backend and

Table 9. Comparison with a DeepSeek-style FP8 strategy on Swin models. TWEO achieves higher throughput and lower memory usage under the same Transformer Engine backend.

Model	Method	Top-1 (%)	Memory	Speed	Speedup
Swin-T	DeepSeek-style FP8	81.2	20G	1640	+2.5%
	TWEO FP8	**81.2**	**15G**	**2508**	**+39%**
Swin-S	DeepSeek-style FP8	82.7	34G	1302	+5.0%
	TWEO FP8	**82.8**	**27G**	**1654**	**+34%**

Table 10. Training resource consumption and FP8 acceleration effects for language models (GPT-2) on H800 GPUs. XL and 3B models were trained on 8 NVIDIA H800s; the 7B model was trained on 16 NVIDIA H800s. All experiments used DeepSpeed ZeRO-3 optimization.

Model	Method	VRAM (GB)	Total Time (H)	iter/H
GPT-2 XL	Default	57	1210	661
	TWEO	57	1219 (+1%)	656 (-1%)
	FP8	55	1000 (-17%)	800 (+21%)
GPT-2 3B	Default	63	1597	500
	TWEO	63	1628 (+2%)	490 (-2%)
	FP8	60	1228 (-23%)	650 (+30%)
GPT-2 7B	Default	56	3696	432
	TWEO	56	3774 (+2%)	423 (-2%)
	FP8	52	2715 (-27%)	588 (+36%)

evaluated on 8 NVIDIA H800 GPUs.

Table 9 shows that TWEO consistently matches or slightly improves accuracy while substantially reducing memory usage and increasing training throughput. We attribute this advantage to TWEO's ability to remove extreme outliers at the source, thereby avoiding the block-wise switching and frequent cast operations required by fine-grained mixed-precision strategies. These results further support TWEO as a simple yet hardware-friendly solution for practical FP8 training.

As shown in Table 10, which records the total training time (in hours) for **100k iterations**, for large language models (GPT-2), the computational overhead of the TWEO method itself is essentially negligible (showing a minor 1-2% increase in training time compared to the default BF16 baseline). More importantly, by fundamentally resolving extreme outliers, TWEO makes the previously highly unstable FP8 training both feasible and efficient. By directly enabling the NVIDIA Transformer Engine without any additional optimization, FP8 training achieved significant speedups on the GPT-2 XL, 3B, and 7B models: training time was reduced by 17%, 23%, and 27%, respectively, and training throughput (iter/H) correspondingly increased by 21%, 30%, and 36%.

Table 11. TWEO's impact on Vision Transformer PTQ. All quantization uses *static AbsMax quantization with per-tensor activations (W(C)A(T))*. The table clearly shows baseline models collapsed at W6A6, while TWEO successfully recovered accuracy to highly competitive levels.

Model	Method	W32A32 (%)	W8A8 (%)	W6A6 (%)
ViT-B (87M)	Default	81.32	79.86	7.41
	+TWEO (Ours)	**81.22**	**80.29**	**66.37**
Swin-T (28M)	Default	81.24	77.69	0.19
	+TWEO (Ours)	**81.40**	**80.94**	**51.69**
Swin-S (50M)	Default	82.74	80.22	0.13
	+TWEO (Ours)	**82.79**	**82.55**	**77.27**
Swin-B (88M)	Default	**83.53**	40.16	0.10
	+TWEO (Ours)	83.41	**83.08**	**80.77**

F. Vision Transformer PTQ Results

To complement the main-text language quantization experiments, we report PTQ results on vision Transformers here. We quantize ViT and Swin models with static AbsMax quantization, using per-channel quantization for weights and the most outlier-sensitive *per-tensor* scheme for activations (W(C)A(T)).

Results in Table 11 again validate TWEO's generality. In the W8A8 setting, TWEO-trained models significantly outperformed the baselines. In the more aggressive W6A6 setting, the baseline models completely collapsed (e.g., Swin-T at 0.19%, ViT-B at 7.41% accuracy), whereas TWEO-trained models successfully recovered the accuracy to highly competitive levels (Swin-S at 77.27%, Swin-B at 80.77%). This shows that TWEO's outlier suppression benefits downstream quantization not only for language models, but also for vision Transformers.

G. Vision Generation Models: DiT Experiments

To further verify TWEO's cross-modal generality mentioned in the abstract, we applied it to a vision generation task using the DiT (Diffusion Transformer) [2] model. We evaluated TWEO's impact on generation quality (FID).

As shown in Table 13, we applied TWEO to a series of DiT models (from DiT-S/2 to DiT-XL/2). The training setup was uniformly $\tau = 30, p = 4, \lambda = 0.01$. The results show that compared to the baseline (Baseline), the FID scores of the TWEO-applied models (+TWEO) increased slightly (e.g., from 9.62 to 9.88 for DiT-XL/2 at 256x256 resolution). This suggests a slight trade-off may exist between suppressing outliers with TWEO and the final generation quality in complex generative tasks. Nonetheless, this validates that the TWEO loss function can be plug-and-play

Table 12. Comparison of 8-bit quantization performance (PPL ↓) on large-scale GPT-2 models (3B and 7B). **A8**: Activation Quantization, **W8**: Weight Quantization. (T): per-tensor, (C): per-channel, (K): per-token.

		Full Precision	Activation Only 8-bit (A8)	Weight Only 8-bit (W8)	Full 8-bit Quant (W8A8)	
Model	Method	(BF16) PPL	per-tensor / per-token	per-tensor / per-channel	W8(T)A8(T) / W8(T)A8(K)	W8(C)A8(T) / W8(C)A8(K)
GPT-2 3B	Default	12.67	72674.75 / 17.54	12.79 / 12.70	55252.88 / 17.38	75394.51 / 17.56
	TWEO	**12.24**	**12.71 / 12.28**	**12.29 / 12.27**	**12.76 / 12.33**	**12.73 / 12.32**
GPT-2 7B	Default	12.49	6117.04 / 414.28	12.77 / 12.67	5463.33 / 386.15	5821.13 / 417.67
	TWEO	**12.02**	**12.93 / 12.18**	**12.03 / 12.01**	**12.93 / 12.18**	**12.92 / 12.17**

Table 13. Generative task performance comparison on DiT models. Training setup: $\tau = 30, p = 4, \lambda = 0.01$. Max Value is the maximum outlier encountered during 250-step generation (no CFG).

Model	**# param (M)**	**Resolution**	**FID ↓**	**Max Value ↓**
DiT-S/2	33	256	68.40	8632
+TWEO	33	256	73.21	569
DiT-B/2	130	256	43.47	26782
+TWEO	130	256	44.58	1223
DiT-L/2	458	256	23.33	56364
+TWEO	458	256	23.92	1856
DiT-XL/2	675	256	9.62	40991
+TWEO	675	256	9.88	2203
DiT-XL/2	675	512	11.93	16641
+TWEO	675	512	12.35	1859

applied to generative Transformers and complete training successfully, opening up possibilities for exploring stable low-bit training in generative models in the future (see also memory-efficient generative quantization [4]).

H. Extended Quantization Results on Larger Models

Due to space limitations in the main text, we only reported quantization results for smaller models or focused on specific ablation studies. In this section, we present the comprehensive post-training quantization (PTQ) results for larger scale models, specifically GPT-2 3B and GPT-2 7B.

Table 12 compares the performance of the Default baseline and our TWEO model across various 8-bit quantization settings. Consistent with our findings on smaller models, the Default 3B and 7B models suffer from catastrophic collapse (e.g., PPL $> 10^4$) under activation quantization due to the presence of extreme outliers. In contrast, TWEO models successfully suppress these outliers during pre-training, enabling the models to maintain near-BF16 performance even with naive 8-bit quantization methods, without requiring complex post-training compensation techniques.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020. 5

[2] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 7

[3] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9, 2019. 5

[4] Jie Shao, Hanxiao Zhang, Hao Yu, and Jianxin Wu. Memory-efficient generative models via product quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16871–16881, 2025. 8