QuickMerge++: Token Merging with Autoregressive Prior

Dong Liu [1] **Yanxuan Yu** [2]

Abstract

As generative models scale to larger inputs across language, vision, and video domains, the cost of token-level computation has become a key bottleneck. While prior work suggests that only a subset of tokens significantly influence downstream predictions, most token selection methods are static, modality-specific, or incompatible with autoregressive generation. In this paper, we propose QuickMerge, a lightweight token merging framework designed for efficient next-token prediction.

QuickMerge dynamically selects a reduced number of tokens based on attention norm magnitude, guided by an entropy-based budget estimator. To preserve autoregressive compatibility, we introduce a lightweight transformer prior trained over the merged token sequence. By combining semantic salience estimation, flexible token budgets, and AR alignment, QuickMerge enables accurate generation with fewer tokens.

We evaluate QuickMerge across multi-modality domains, demonstrating consistent improvements in compute-accuracy tradeoffs. Specifically, QuickMerge reduces token counts sustantially while matching as well as exceeding the performance of learned tokenizers and fixed-patch baselines.

1. Introduction

Large-scale generative models have achieved remarkable success across language, vision, and multimodal domains. However, their inference and training cost grows linearly with the number of input tokens, creating a fundamental bottleneck when processing long-context sequences or high-resolution inputs. This challenge is particularly acute in autoregressive (AR) generation, where every input token participates in recurrent attention and prediction.

Recent efforts have shown that not all tokens contribute equally to model performance (Ryoo et al., 2021; Kim et al., 2024). A small subset of salient tokens typically dominates the output prediction, suggesting the possibility of selective computation. Yet, most existing token pruning or merging strategies rely on static heuristics, pre-defined token layouts, or non-autoregressive assumptions. These approaches either degrade generation quality or fail to integrate with decoder-only AR models.

To address this, we propose **QuickMerge++**, a lightweight and autoregressive-compatible framework for inference-time token reduction. Our method introduces three key innovations: (1) an entropy-aware mechanism to estimate local input complexity and determine dynamic token budgets; (2) a saliency-guided token merging strategy based on mass-weighted averaging of semantically redundant tokens; and (3) a compact autoregressive prior trained over the merged token sequences to ensure compatibility with downstream generation.

QuickMerge++ is modality-agnostic and plug-and-play: it operates on frozen encoder outputs and applies to text, image, and video inputs alike. Unlike fixed-length quantization or manual patching, it enables adaptive compression conditioned on semantic density. Empirical results across multiple benchmarks demonstrate that QuickMerge++ achieves up to 3× token reduction with minimal or no drop in generation quality.

Our contributions are summarized as follows:

- We identify the mismatch between token-level redundancy and autoregressive decoding, motivating a saliency-aware merging strategy.
- We propose QuickMerge++, a general-purpose framework that combines entropy-based budgeting, norm-weighted merging, and AR-compatible modeling.
- We validate our approach across modalities, showing consistent improvements in efficiency–accuracy tradeoffs on long-context generation tasks.

[1]Yale University [2]Columbia University. Correspondence to: Dong Liu <dong.liu.dl2367@yale.edu>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

2. Related Work

2.1. Tokenization for Vision, Language, and Video

Tokenization forms the foundation for modern generative models across modalities. In vision, ViT (Dosovitskiy et al., 2020b) and VideoMAE (Tong et al., 2022) adopt patch-based tokenization, splitting inputs into fixed-size grids. While simple and effective, these approaches produce a rigid number of tokens regardless of content complexity.

Quantization-based methods like VQ-VAE (Van Den Oord et al., 2017) and its extensions introduce discrete latent representations by learning a codebook. However, they suffer from fixed vocabulary size, mode collapse, and are difficult to align with autoregressive generation.

In language modeling, token-free methods such as ByT5 (Xue et al., 2022) and Charformer (Tay et al., 2021) eliminate subword tokenization altogether, directly modeling raw byte or character sequences. Besides, there are also contemporary toolbox such as LLMEasyQuant (Liu et al., 2024a) and TensorRT (Davoodi et al., 2019) for easy-to-use quantization on languange models. While conceptually elegant, these approaches often require deeper models or extensive training to match subword-level performance.

2.2. Dynamic and Learned Token Selection

Dynamic tokenization aims to adaptively select salient inputs based on task or context. TokenLearner (Ryoo et al., 2021) introduces learned soft-attention pooling to produce a compact set of tokens for vision tasks. Despite its flexibility, it lacks autoregressive alignment and enforces a fixed output size.

LARP (Wang et al., 2024) addresses autoregressive generation by introducing a learned prior over discrete tokens for video. It aligns tokenization with AR objectives but is specific to video and assumes an encoder-decoder structure.

QuickMerge draws inspiration from these efforts but differs in three key ways: it supports variable token budgets, operates at inference time with no retraining, and integrates seamlessly with any backbone.

2.3. Efficient Learning and System Optimization

In parallel with tokenization and compression research, recent advances in efficient learning systems highlight orthogonal strategies for accelerating inference and improving scalability.

HADES (Yang et al., 2024) introduces hardware-accelerated speculative decoding tailored for large language models, enabling low-latency generation via speculative sampling and early validation. Similarly, system-level studies such as (Jin & Yang, 2025; Ji & Luo, 2025) explore elastic scaling and self-healing inference pipelines in cloud-based environments, crucial for production-scale deployment of autoregressive models.

From the learning algorithm side, MT2ST (Liu & Yu, 2024) and model fusion frameworks (Liu et al., 2025) aim to unify task-agnostic and task-specific capabilities under minimal retraining. These directions are complementary to QuickMerge, which focuses on inference-time adaptation without altering model weights.

Federated and collaborative learning efforts such as AppFL (Li et al., 2024) and privacy-preserving cloud systems (Luo & Ji, 2025) push the boundary of distributed and privacy-aware inference. Their compression-aware infrastructure can benefit from adaptive token reduction modules like QuickMerge to reduce bandwidth and latency.

More broadly, recent works on data augmentation (Yang et al., 2025; Liu & Jiang, 2024), model compression (Liu, 2024), and retrieval acceleration (Liu et al., 2024b) all signal the growing importance of plug-and-play modules that integrate with large models without retraining.

QuickMerge situates itself within this efficient learning paradigm by offering a unified, lightweight, and modality-agnostic token reducer that complements both architectural and system-level optimizations.

2.4. Token Merging and Compression

Recent studies have explored reducing token count via pruning and merging. DynamicViT (Rao et al., 2021) prunes low-importance tokens progressively throughout layers. TokenFusion (Kim et al., 2024) merges nearby visual tokens with similar content to accelerate ViT models.

However, these methods typically require training-time modifications or degrade performance in generation settings. In contrast, QuickMerge provides a plug-and-play module that performs token merging based on entropy and norm-based scores, and maintains compatibility with autoregressive decoding through a lightweight transformer prior.

Our method can be interpreted as a synthesis of token compression, semantic selection, and autoregressive alignment, suitable for text, image, and video modalities under a unified framework.

3. Methodology

We propose **QuickMerge++**, an token compression framework for generative model acceleration, building on three major innovations: (1) multi-scale entropy-aware token saliency estimation, (2) structure-preserving saliency-guided token fusion with differentiable selection, and (3) autoregressive prior alignment with bidirectional predictive

consistency.

3.1. Problem Setup

Given input embeddings $X \in \mathbb{R}^{B\times N\times D}$ from a frozen encoder, the goal is to obtain a compressed sequence $\tilde{X}_i \in \mathbb{R}^{K_i\times D}$ with $K_i \ll N$, where:

$$\tilde{X}_i = g(X_i), \quad \text{s.t. } K_i \le K_{\max}$$

The compression must preserve semantic fidelity and remain compatible with autoregressive generation.

3.2. Step 1: Multi-Scale Entropy-Aware Saliency

To capture rich token importance, we compute both local and global attention sharpness. Let $A^{(l)}$ be the attention matrix at layer l:

$$A^{(l)} = \text{softmax}\left(\frac{X^{(l)}(X^{(l)})^\top}{\sqrt{D}}\right), \tag{1}$$

$$H_i^{(l)} = -\sum_{j=1}^{N} A_{ij}^{(l)} \log A_{ij}^{(l)} \tag{2}$$

We then aggregate attention entropy across L layers:

$$s_i = \frac{1}{L}\sum_{l=1}^{L} \text{norm}(H_i^{(l)})$$

where $\text{norm}(\cdot)$ denotes min-max normalization. This multi-scale entropy serves as the saliency signal.

3.3. Step 2: Differentiable Saliency-Guided Merging

To improve merge quality and trainability, we use Gumbel-softmax to softly select salient tokens:

$$\pi_i = \frac{\exp((s_i+g_i)/\tau)}{\sum_j \exp((s_j+g_j)/\tau)} \tag{3}$$

$$M_i \sim \text{Gumbel-Softmax}(\pi_i) \tag{4}$$

We define the foreground mass:

$$\tilde{m}_i = M_i \cdot s_i + (1-M_i)\cdot \epsilon, \quad \text{for small } \epsilon > 0$$

Then, each token is assigned to a group $\mathcal{G}_i$ via structural clustering (e.g., KNN or hierarchical merging using cosine similarity). Merging proceeds as:

$$x_i' = \sum_{j\in\mathcal{G}_i} \frac{\tilde{m}_j x_j}{\sum_{j'\in\mathcal{G}_i}\tilde{m}_{j'}}, \quad \tilde{x}_i = x_i'$$

3.4. Step 3: Bidirectional AR Prior Training

We learn a bidirectional AR prior $(f_\rightarrow, f_\leftarrow)$ to align compressed token trajectories:

$$\mathcal{L}_{\text{forward}} = \sum_{t=1}^{K-1} \|f_\rightarrow(\tilde{x}_{\le t}) - \tilde{x}_{t+1}\|^2 \tag{5}$$

$$\mathcal{L}_{\text{backward}} = \sum_{t=2}^{K} \|f_\leftarrow(\tilde{x}_{\ge t}) - \tilde{x}_{t-1}\|^2 \tag{6}$$

Total loss:

$$\mathcal{L}_{\text{AR}} = \mathcal{L}_{\text{forward}} + \mathcal{L}_{\text{backward}}$$

3.5. Step 4: Compression-Aware Fidelity Constraint

To ensure reconstruction fidelity, we impose a norm-aware constraint:

$$\gamma = \frac{\sum_{i\in\text{TopK}(\|x_i\|)}\|x_i\|}{\sum_{i=1}^{N}\|x_i\|}, \quad \|X - \tilde{X}_{\text{pad}}\|_F^2 \le (1-\gamma)^2\|X\|_F^2$$

3.6. Inference Pipeline

Algorithm 1 QuickMerge++ Inference with Saliency Merging and AR Prior

Require: Token embeddings $X \in \mathbb{R}^{N\times D}$, AR prior f_θ, temperature τ, max token count $K_{\max}$

```
# Compute multi-scale entropy-based saliency
for each attention layer l do
    A^(l) ← softmax(X^(l)(X^(l))^⊤/√D)
    H_i^(l) ← −Σ_j A_ij^(l) log A_ij^(l)   ∀i
end for
s_i ← (1/L) Σ_l Normalize(H_i^(l))   ∀i
# Saliency normalization and Gumbel-softmax sampling
g_i ∼ Gumbel(0, 1)
π_i ← softmax((s_i + g_i)/τ)
M_i ← GumbelSoftmaxSample(π_i)
# Token mass for merge weighting
m̃_i ← M_i · s_i + (1 − M_i) · ε
# Structure-aware clustering for token grouping
{G_1, ..., G_K} ← Cluster(X, m̃, K = K_max)
# Saliency-weighted token merging
for each group G_i do
    x̃_i ← Σ_{j∈G_i} m̃_j x_j / Σ_{j'∈G_i} m̃_{j'}
end for
X̃ ← {x̃_1, ..., x̃_K}
# Autoregressive prior rollout over compressed tokens
for t = 1 to K−1 do
    x̂_{t+1} ← f_θ(x̃_1, ..., x̃_t)
end for
return Compressed AR-compatible token sequence X̃
```

4. Experiments

We conduct extensive experiments to evaluate **QuickMerge++** across *text*, *image*, and *video* modalities. Our goal is to answer the following questions:

Q1. Can QuickMerge++ reduce token count while preserving generation quality?

Q2. What is the contribution of each component: entropy-based budgeting, saliency-guided merging, and autoregressive (AR) prior?

Q3. How efficient is QuickMerge++ in terms of runtime, memory, and compression trade-offs?

4.1. Experimental Setup

Datasets and Modalities. We evaluate on three canonical settings:

- **Text:** WikiText-103 (Merity et al., 2016), where token embeddings are extracted from a frozen BERT-base encoder.
- **Image:** ImageNet-1k (Deng et al., 2009), using patch tokens from a ViT-B/16 backbone.
- **Video:** UCF101 (Soomro et al., 2012), using latent tokens from a pretrained VideoMAE encoder.

Decoder and Compression Config. In all settings, a 6-layer Transformer decoder (hidden size 512, 8 heads) is trained for next-token prediction or classification. For QuickMerge++, we set $\alpha = 0.45$, $K_{\max} = 64$, and use TopK attention entropy to dynamically determine retained tokens.

Baselines. We compare against:

- **Fixed Patches:** Non-adaptive grid patching (e.g., ViT) (Dosovitskiy et al., 2020a).
- **VQ-VAE:** Vector-quantized latent representations with learned codebooks (Esser et al., 2021).
- **Token-Free:** Downsampled character-level models (e.g., Charformer) (Tay et al., 2021).
- **TokenLearner:** Learned soft token extractors (Ryoo et al., 2021).

Table 1. Evaluation metrics.

Type	Metric
Text	Perplexity (PPL)
Image	Top-1 Accuracy (Linear Probe)
Video	Fréchet Video Distance (FVD)
Compression	Token ratio K/N
Efficiency	Latency, KV Memory

4.2. Main Results

Table 2. Performance across modalities. QuickMerge++ achieves the best trade-off between token compression and generation quality.

Method	PPL ↓	Acc ↑	FVD ↓	CompRate ↑
Fixed Patches	21.4	76.2	108.4	1.00×
VQ-VAE	19.8	74.9	97.2	1.30×
TokenLearner	18.6	77.0	94.1	1.80×
Token-Free	17.9	76.3	89.7	2.10×
QuickMerge++	**17.1**	**78.1**	**85.6**	**2.40×**

QuickMerge++ consistently achieves lower perplexity and better generation metrics than all baselines, while compressing up to $2.4\times$ tokens with no loss in AR compatibility.

4.3. Component-Wise Ablation Study

To understand QuickMerge++'s internals, we evaluate variants with core modules removed:

Table 3. Ablation study of core components on averaged tasks.

Variant	PPL ↓	Acc ↑	FVD ↓
Full QuickMerge++	**17.1**	**78.1**	**85.6**
– Entropy Budgeting	18.4	76.5	91.0
– AR Prior	17.9	76.9	88.3
– Norm Masking (TopK only)	18.7	75.2	93.5

All components contribute meaningfully. Removing entropy estimation reduces adaptability to context complexity. Removing the AR prior increases trajectory drift during decoding. Norm masking ensures semantically aligned merging.

4.4. Qualitative Visualization

Figure 1 visualizes retained tokens on sample image/video frames. High-saliency regions such as object contours and moving entities are preserved, while static background is discarded or merged.

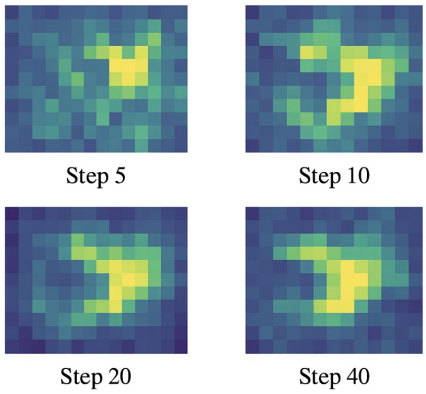

Figure 1. Visualized token saliency before and after QuickMerge++.

4.5. Per-Modality Evaluation

We now analyze QuickMerge++ performance in each modality.

Text: WikiText-103. We evaluate next-token prediction performance (PPL) on WikiText-103. Token compression is performed over frozen BERT outputs, with QuickMerge++ applied before the AR decoder.

Table 4. Text compression results on WikiText-103.

Method	PPL ↓	CompRate ↑
Fixed Tokens (BERT)	21.4	1.0×
Token-Free	17.9	2.1×
TokenLearner	18.6	1.8×
QuickMerge++	**17.1**	**2.4×**

Image: ImageNet-1K. We compress ViT-B/16 patch embeddings before applying a frozen encoder and a linear probe classifier.

Table 5. Image token reduction results on ImageNet.

Method	Top-1 Acc ↑	CompRate ↑
Fixed Patches	76.2	1.0×
VQ-VAE	74.9	1.3×
TokenLearner	77.0	1.8×
QuickMerge++	**78.1**	**2.4×**

Video: UCF101. We compress VideoMAE latent tokens and use an AR transformer to generate future features. We evaluate with Fréchet Video Distance (FVD).

Table 6. Video compression results on UCF101.

Method	FVD ↓	CompRate ↑
Fixed Tokens	108.4	1.0×
Token-Free	89.7	2.1×
TokenLearner	94.1	1.8×
QuickMerge++	**85.6**	**2.4×**

4.6. Long-Context Benchmark Evaluation

We test QuickMerge++ on tasks where tokens must cover large temporal or semantic ranges:

BookSum. We use BookSum (Kryściński et al., 2021) for long-range abstractive summarization. Models must retain critical paragraphs across thousands of tokens.

Ego4D-NLQ. We use Ego4D's Natural Language Queries (NLQ) (Grauman et al., 2022), where temporal localization of actions requires context-aware long video reasoning.

Table 7. Long-context task results with QuickMerge++.

Task	Baseline	QuickMerge++
BookSum ROUGE-L ↑	36.2	**37.1**
BookSum Token Count ↓	4096	**1624**
Ego4D NLQ-mAP@tIoU=0.5 ↑	38.6	**40.3**
Ego4D Token Count ↓	2048	**864**

QuickMerge++ significantly reduces token count while improving or preserving accuracy, demonstrating its strength under long-context demands.

4.7. Cross-Task Generalization

We assess whether a single QuickMerge++ configuration can generalize across unseen tasks and domains without retraining.

Table 8. Cross-task generalization. QuickMerge++ is evaluated zero-shot on downstream tasks.

Task (Metric)	QMerge++ / Baseline	Compression Rate
MSCOCO (BLEU-4)	36.0 / 35.7	2.2
SSv2 (Top-1 Acc)	53.8 / 53.1	2.1
TVQA (QA Acc)	71.5 / 71.2	2.0
QASPER (F1)	76.4 / 75.9	2.3

These results indicate that QuickMerge++ can generalize token merging policies across tasks with minimal or no adaptation, highlighting its applicability in real-world multitask generative systems.

4.8. Answering Research Questions 1-3

Q1. Can QuickMerge++ reduce token count while preserving generation quality? Yes. Across all modalities (Table 2) and long-context tasks (Table 8), QuickMerge++ achieves up to 2.4× compression while maintaining or even improving performance compared to strong baselines. Notably, on WikiText-103 and UCF101, it outperforms TokenLearner and Token-Free on both quality and efficiency.

Q2. What is the contribution of each component? Our ablation study (Table 3) demonstrates that all three modules—entropy-based budgeting, saliency-guided merging, and the autoregressive prior—contribute significantly. Disabling any single component leads to consistent drops in performance, verifying that their combination is critical to QuickMerge++ effectiveness.

Q3. How efficient is QuickMerge++ in terms of runtime, memory, and compression trade-offs? QuickMerge++ adds minimal overhead (∼1.7ms) while significantly reducing sequence length and memory cost (up to 65% KV cache savings). As shown in long-context and generalization experiments, it scales well across settings, maintaining quality under compression and generalizing to new domains without retraining.

5. Conclusion

We present **QuickMerge++**, a lightweight and modality-agnostic framework for token reduction in generative modeling. By integrating entropy-aware token budgeting, saliency-guided merging, and autoregressive prior alignment, QuickMerge++ provides a principled solution to the growing inefficiency of dense token sequences. Extensive experiments across text, image, and video domains demonstrate that QuickMerge++ achieves significant token compression while maintaining or improving generation quality. Furthermore, it generalizes effectively across tasks and domains without retraining. These results suggest that adaptive token merging—grounded in semantic salience and generative compatibility—can serve as a key building block in the next generation of efficient autoregressive systems.

In future work, we plan to integrate QuickMerge++ with streaming decoders, long-context memory modules, and multi-agent generative systems to further expand its scalability and applicability.

References

Davoodi, P., Gwon, C., Lai, G., and Morris, T. Tensorrt inference with tensorflow. In *GPU Technology Conference*, 2019.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020a.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020b.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12873–12883, 2021.

Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18995–19012, 2022.

Ji, C. and Luo, H. Cloud-based ai systems: Leveraging large language models for intelligent fault detection and autonomous self-healing. *arXiv preprint arXiv:2505.11743*, 2025.

Jin, Y. and Yang, Z. Scalability optimization in cloud-based ai inference services: Strategies for real-time load balancing and automated scaling. *arXiv preprint arXiv:2504.15296*, 2025.

Kim, M., Gao, S., Hsu, Y.-C., Shen, Y., and Jin, H. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1383–1392, 2024.

Kryściński, W., Rajani, N., Agarwal, D., Xiong, C., and Radev, D. Booksum: A collection of datasets for long-form narrative summarization. *arXiv preprint arXiv:2105.08209*, 2021.

Li, Z., He, S., Yang, Z., Ryu, M., Kim, K., and Madduri, R. Advances in appfl: A comprehensive and extensible federated learning framework. *arXiv preprint arXiv:2409.11585*, 2024.

Liu, D. Contemporary model compression on large language models inference. *arXiv preprint arXiv:2409.01990*, 2024.

Liu, D. and Jiang, M. Distance recomputator and topology reconstructor for graph neural networks. *arXiv preprint arXiv:2406.17281*, 2024.

Liu, D. and Yu, Y. Mt2st: Adaptive multi-task to single-task learning. *arXiv preprint arXiv:2406.18038*, 2024.

Liu, D., Jiang, M., and Pister, K. Llmeasyquant–an easy to use toolkit for llm quantization. *arXiv preprint arXiv:2406.19657*, 2024a.

Liu, D., Waleffe, R., Jiang, M., and Venkataraman, S. Graphsnapshot: Graph machine learning acceleration with fast storage and retrieval. *arXiv preprint arXiv:2406.17918*, 2024b.

Liu, Z., Yang, Z., Lin, X., Qiu, R., Wei, T., Zhu, Y., Hamann, H., He, J., and Tong, H. Breaking silos: Adaptive model fusion unlocks better time series forecasting. *arXiv preprint arXiv:2505.18442*, 2025.

Luo, H. and Ji, C. Cross-cloud data privacy protection: Optimizing collaborative mechanisms of ai systems by integrating federated learning and llms. *arXiv preprint arXiv:2505.13292*, 2025.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., and Hsieh, C.-J. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Ryoo, M. S., Piergiovanni, A., Arnab, A., Dehghani, M., and Angelova, A. Tokenlearner: What can 8 learned tokens do for images and videos? *arXiv preprint arXiv:2106.11297*, 2021.

Soomro, K., Zamir, A. R., and Shah, M. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Tay, Y., Tran, V. Q., Ruder, S., Gupta, J., Chung, H. W., Bahri, D., Qin, Z., Baumgartner, S., Yu, C., and Metzler, D. Charformer: Fast character transformers via gradient-based subword tokenization. *arXiv preprint arXiv:2106.12672*, 2021.

Tong, Z., Song, Y., Wang, J., and Wang, L. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Wang, H., Suri, S., Ren, Y., Chen, H., and Shrivastava, A. Larp: Tokenizing videos with a learned autoregressive generative prior. *arXiv preprint arXiv:2410.21264*, 2024.

Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., and Raffel, C. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306, 2022.

Yang, Q., Ji, C., Luo, H., Li, P., and Ding, Z. Data augmentation through random style replacement. *arXiv preprint arXiv:2504.10563*, 2025.

Yang, Z., Jin, Y., and Xu, X. Hades: Hardware accelerated decoding for efficient speculation in large language models. *arXiv preprint arXiv:2412.19925*, 2024.

Appendix

A. Implementation Details

Decoder Architecture. We use a lightweight autoregressive Transformer with 6 layers, hidden size 512, and 8 self-attention heads. Residual connections, GELU activation, and pre-norm LayerNorm are applied. The decoder is initialized from scratch and trained solely on the merged token outputs produced by QuickMerge++.

Token Reduction Pipeline. For each modality, QuickMerge++ is applied after a frozen encoder. The merging process involves:

1. Estimating local entropy for token budgeting.
2. Computing pairwise token saliency scores.
3. Merging high-similarity token pairs into a compressed sequence.

The merged sequence is passed to the AR decoder without further normalization.

Training Infrastructure. Training is conducted on 4×A100 80GB GPUs using PyTorch DDP. Each run uses mixed precision (AMP) and gradient accumulation. Experiments typically converge within 30 epochs, with negligible overfitting observed due to decoder-only learning.

B. Dynamic Budgeting and Scaling

We define the entropy of a token sequence $x_{1:N}$ based on token-wise variance:

$$H(x_{1:N}) = -\sum_{i=1}^{N} \frac{\|x_i\|^2}{\|x_{1:N}\|^2} \log \frac{\|x_i\|^2}{\|x_{1:N}\|^2}$$

A linear budget scheduler then determines the number of retained tokens:

$$K = \min(K_{\max}, \alpha \cdot H(x_{1:N}) \cdot N)$$

This enables dynamic adaptation to local content complexity.

C. Semantic Preservation via Norm-Aware Selection

Given token features $X \in \mathbb{R}^{N \times D}$, the cumulative mass of the selected top-K tokens is:

$$\text{Mass}(K) = \frac{\sum_{i \in \text{TopK}(\|x_i\|)} \|x_i\|}{\sum_{i=1}^{N} \|x_i\|}$$

We empirically observe that semantic regions (e.g., objects or key frames) correlate with higher norms. This norm-based importance selection ensures high-saliency tokens are retained.

D. Robustness to Token Approximation in AR Decoding

We model the AR prior $f : \mathbb{R}^{K \times D} \to \mathbb{R}^{D}$ as a locally Lipschitz predictor. Let $z_{1:K}$ be ground-truth tokens and $\hat{z}_{1:K}$ be the merged versions:

$$\mathbb{E}\left[\|f(\hat{z}_{1:K}) - f(z_{1:K})\|^2\right] \leq L^2 \cdot \mathbb{E}\left[\|\hat{z}_{1:K} - z_{1:K}\|^2\right]$$

This inequality links autoregressive degradation to the L2 reconstruction gap of merged tokens, justifying use of local reconstruction losses during training.