

SUPPLEMENTARY MATERIALS FOR KNAPFORMER: AN ONLINE LOAD BALANCER FOR EFFICIENT DIFFUSION TRANSFORMERS TRAINING

Anonymous authors

Paper under double-blind review

1 SOURCE CODE

We attach the source code of our KnapFormer in the supplementary. We will also open-source it to the community.

2 MODIFICATIONS TO MM-DiT

We extend FLUX—a multimodal mixture-of-experts (MoE) transformer with handcrafted modality-specific routing—to support efficient large-scale training on tightly packed, interleaved sequences. Our contributions focus on maximizing training throughput while preserving compatibility with distributed training infrastructure:

Removing T5 padding tokens. FLUX originally tokenizes text using a fixed-length, padded T5 encoder. These padded tokens are not masked out in the encoder’s attention blocks and are also used as conditioning tokens during generation. While this design simplifies implementation, it introduces several drawbacks: (1) it deviates from the intended T5 usage, where padded tokens are masked out to avoid meaningless attention; (2) it reduces training efficiency, especially in low-resolution image pretraining where the padded tokens incur non-negligible computational overhead; and (3) it can subtly affect generation quality, as shown in prior analysis (Toker et al., 2025) and Fig. 1. To address these issues, we remove all padded T5 tokens in MM-DiT training, resulting in variable-length text tokens per sample.

Packed Sequence with Interleaved Modalities. We organize inputs as tightly packed sequences where text and image tokens are interleaved per sample (e.g., [txt_1, img_1, txt_2, img_2, ...]), maximizing GPU utilization and minimizing padding. FLUX routes modalities through separate expert MLPs in the DoubleStreamBlock blocks, and we implement efficient routing by precomputing txt_indices and img_indices to dispatch tokens to their respective MLPs. A 1D seq_ids tensor enables each token to retrieve its corresponding per-sample latent vector (vec) for conditional modulation.

All-Gathered Modulation with Global Sequence IDs. To support distributed training with our KnapFormer sequence balancing, we address two key challenges: (1) after routing, a sample’s tokens and its latent vector vec may reside on different GPUs, and (2) tokens from the same sample may be split across multiple devices. Rather than duplicating vectors across tokens (which would increase communication and redundant computation), we all-gather per-sample vectors across GPUs once, and use a global seq_ids tensor to let each token retrieve its modulation parameters (scale, shift, gate) efficiently. This strategy preserves correctness while minimizing communication and compute cost.

FSDP-Compatible Conditional Execution for Sparse MoE Routing. FLUX routes text and image tokens to separate expert MLPs in each DoubleStreamBlock. However, sequence parallelism may cause some GPUs to receive only a subset of modalities (e.g., only image tokens). To prevent Fully Sharded Data Parallel (FSDP) from hanging in the backward pass, we ensure that all expert branches participate in the forward pass by applying dummy gradient-preserving operations on unused parameters.

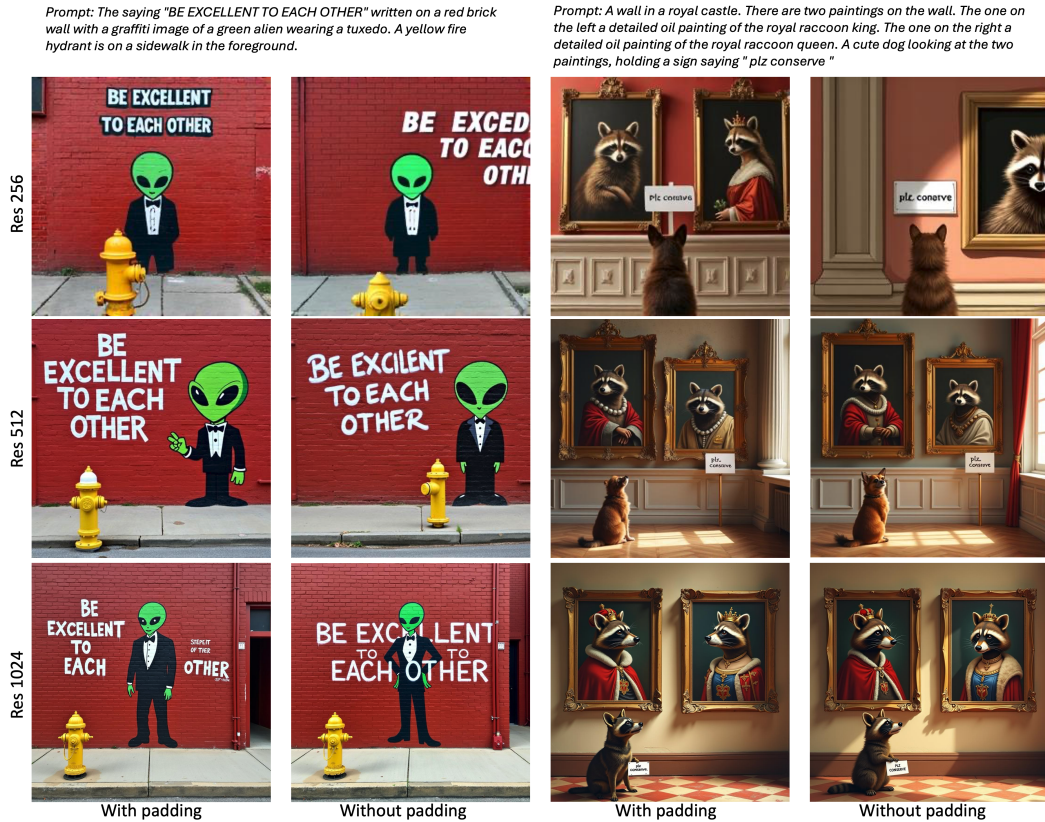


Figure 1: Effect of removing T5 padding tokens in FLUX inference: using the original FLUX.1-dev checkpoint (without additional finetuning), in the low-resolution setting (256), removing text padding significantly affects the image layout, often resulting in a noticeable “right shift” of the content. This effect is less pronounced at higher resolutions, but still it can cause text rendering accuracy to degrade. Such nuances can be mitigated by excluding T5 padding tokens in the FLUX training stage.

3 USE OF LOAD BALANCER FOR SEQUENCE PARALLELISM

Our load balancer is a super set of sequence parallelism and has the benefits of not requiring synchronized data loaders among sequence parallel groups, as shown in Fig. 2. In this regard, our load balancer offers a unified compute balancing angle into the problem of handling long sequences and heterogeneous data sources.

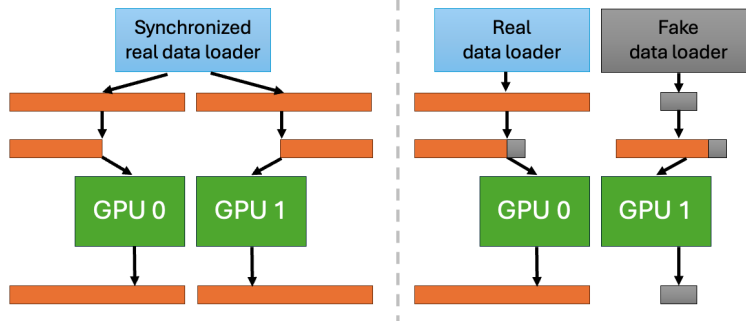


Figure 2: Our load balancer can also be used in a way similar to sequence parallelism, but we don’t require data loader synchronization. Instead, we can use a dummy data loader spitting very short sequences to help offload the compute burden on the other GPU.

4 ONLINE T5 AND VAE BALANCERS

Diffusion-based text-to-image models (Labs, 2024; Esser et al., 2024) commonly use a T5 encoder (Raffel et al., 2020) to process text prompts. In distributed training, GPUs may receive different numbers of (text, image) pairs due to dataloader variability—such as resolution-dependent batch sizes or sampling imbalance. As a result, the T5 encoding workload can become imbalanced across GPUs, introducing inefficiencies in pipeline-parallel or multi-stage training setups.

Since the T5 tokenizer pads inputs to a fixed length, the per-string compute cost is effectively uniform. Therefore, balancing this stage is straightforward: we evenly redistribute the text strings across GPUs, perform the T5 encoding in parallel, and then redistribute the encoded embeddings back to their original GPUs. This ensures balanced utilization during the T5 stage with minimal communication overhead.

A similar online balancing strategy applies to VAE encoders, which are used to process image or video data into latents. In this case, inputs can be split into spatial or spatiotemporal tiles and distributed across GPUs in a balanced manner. This is particularly beneficial in mixed-resolution or joint image-video training setups, where the cost of VAE encoding varies significantly across samples.

Together, these modular balancers complement the main KnapFormer algorithm and extend the benefits of online load balancing to encoder-heavy stages of multimodal pipelines.

REFERENCES

- Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.
- Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- Michael Toker, Ido Galil, Hadas Orgad, Rinon Gal, Yoad Tewel, Gal Chechik, and Yonatan Belinkov. Padding tone: A mechanistic analysis of padding tokens in t2i models, 2025. URL <https://arxiv.org/abs/2501.06751>.