A Notations used in paper

Table 4. Notations

Symbol	Definition
L	Number of layers in backbone model
$\mathbf{B}$, B, b	Training batch size, micro-batch size and number of samples in a partial-batch
$\mathbf{S}$, s	Set of model stages and model stage
$\mathbf{P}_l^f(B)$, $\mathbf{P}_l^b(B)$	Forward and backward computation time of layer l given batch size B
$\mathbf{C}_{l,l+1}^f(B)$, $\mathbf{C}_{l+1,l}^b(B)$	Data size of communication in forward and backward pass between layers l and $l+1$ given batch size B
$\mathbf{R}_x$, $\mathbf{L}_x$	Bandwidth and latency of communication type x (e.g., allreduce (ar), point-to-point (p2p))
$\mathbf{G}_l(B)$	Gradient size of layer l given batch size B
$\mathbf{O}_l(B)$	Output size of layer l given batch size B
$T_S(s)$	Synchronization time of stage s
$T_C(s)$	Compensation time of stage s
T_0	Maximum micro-batch execution time per stage or inter-stage communication time
T_0^{S-C}	Maximum gap between synchronization time and compensation time per stage
T_B	Length of a pipeline bubble (idle time)

B Partial-batch layer processing

The total number of samples processed by a partial-batch layer on all devices in a pipeline bubble is smaller than the batch size (as otherwise it would be a full-batch layer instead). A partial-batch layer is scheduled in multiple pipeline bubbles, in order to fully process the training batch. Especially, after introducing a partial-batch layer (h, u_h+k_h, b) in one pipeline bubble (line 5 in Alg. 1), the layer u_h+k_h of component h is the first ready layer of that component to be considered when filling the following pipeline bubbles, treated as a full-batch layer on the remaining batch in Alg. 1. In this way, the layer can be scheduled to process all or part of the remaining batch in a subsequent pipeline bubble.

As a partial-batch layer is executed in multiple pipeline bubbles, inputs to and outputs from the layer's execution in the bubbles should be properly partitioned and concatenated, and sent to the correct consumers. As illustrated in Fig. 15, we split the input batch of the partial-batch layer and dispatch the partial batches to devices in the pipeline bubbles where the layer is scheduled in advance. We concatenate the outputs of the partial-batch layer from the pipeline bubbles after the last partial batch is processed.

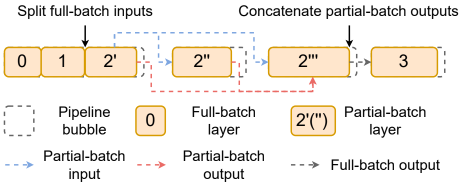

Figure 15. Input split and output concatenation of partial-batch layer's processing among pipeline bubbles. The partial-batch layer 2 of a non-trainable component is scheduled in 3 consecutive pipeline bubbles.

C Optimization time

In Table 5 we present the solution time of the dynamic programming (DP) approach to decide the backbone partitioning and the pipeline bubble filling algorithm (Greedy) when training models on the largest batch size on 64 GPUs. We solve the sub-problems in DP algorithm in parallel on at most 64 CPU cores. For single backbone models, the solution time is less than 1 second. For cascaded diffusion models, the solution time of DP is longer, while we still consider it acceptable to spend one or two minutes to derive an optimal partitioning scheme offline. The complexity of the bubble filling algorithm is not high, and we run it on only 1 CPU core. The total solution time for filling all pipeline bubbles is less than 3 seconds.

Table 5. Solution time of backbone partitioning (DP) and pipeline bubble filling (Greedy) algorithms in seconds

Model	DP	Greedy
Stable Diffusion v2.1	0.5	0.7
ControlNet v1.0	0.5	0.5
CDM-LSUN	145	2.5
CDM-ImageNet	87	1.7