Masked Gated Linear Unit

Yukito Tajima, Nakamasa Inoue, Yusuke Sekikawa, Ikuro Sato, Rio Yokota

NeurIPS 2025 Poster

Institute of SCIENCE TOKYO

DENSO IT LAB

Motivation

- Transformer FFN layers (e.g., SwiGLU, GEGLU) require two full matrix multiplications per token.
- This doubles weight-memory bandwidth at inference which is a key latency bottleneck.
- Compute is cheap; memory access dominates on modern GPUs.

GLU Family

- GLU introduces element-wise gating: $\mathrm{GLU}(x) = g\left(xW_g\right) \odot \left(xW_v\right)$

  where $x \in \mathbb{R}^h$, $W_g, W_v \in \mathbb{R}^{h \times d}$ and g an arbitrary gating function.

GLU Family

- GLU introduces element-wise gating: $\mathrm{GLU}(x) = g\left(xW_g\right) \odot \left(xW_v\right)$

  where $x \in \mathbb{R}^h$, $W_g, W_v \in \mathbb{R}^{h\times d}$ and g an arbitrary gating function.

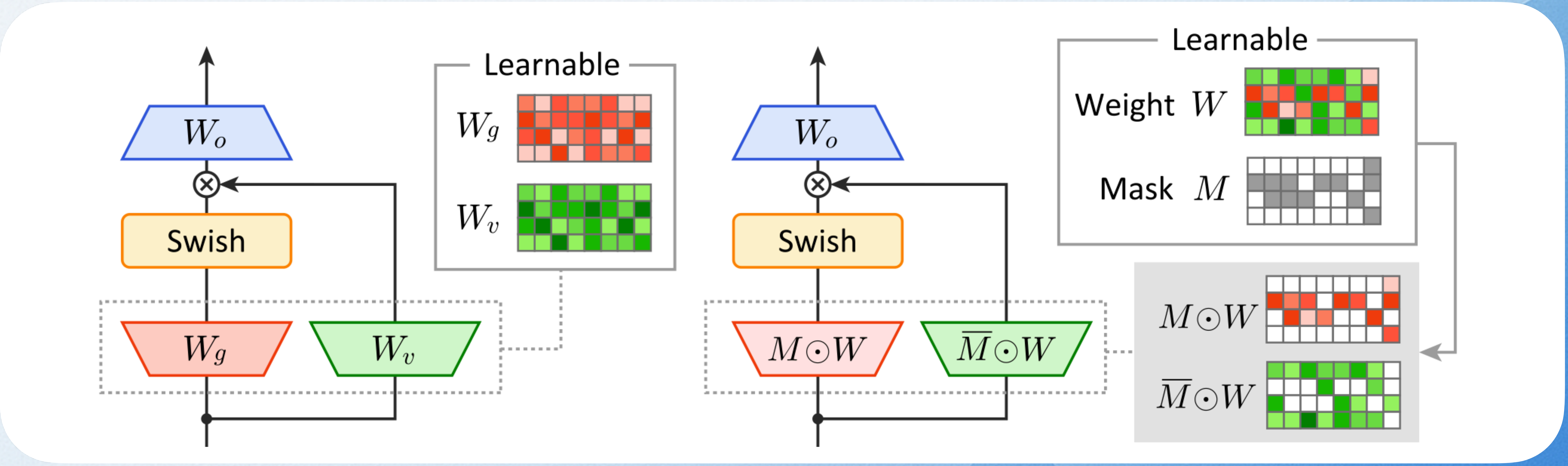
- Improves expressivity and gradient flow compared to other FFN structures.

- But requires two large projections (value + gate).

What is MGLU?

- Masked Gated Linear Unit (MGLU) replaces two projections (W_v, W_g) with one shared matrix W, where each neuron is masked to act as value or gate:

$$y = (x(1 - M) \odot W) \odot \sigma(x(M \odot W))$$

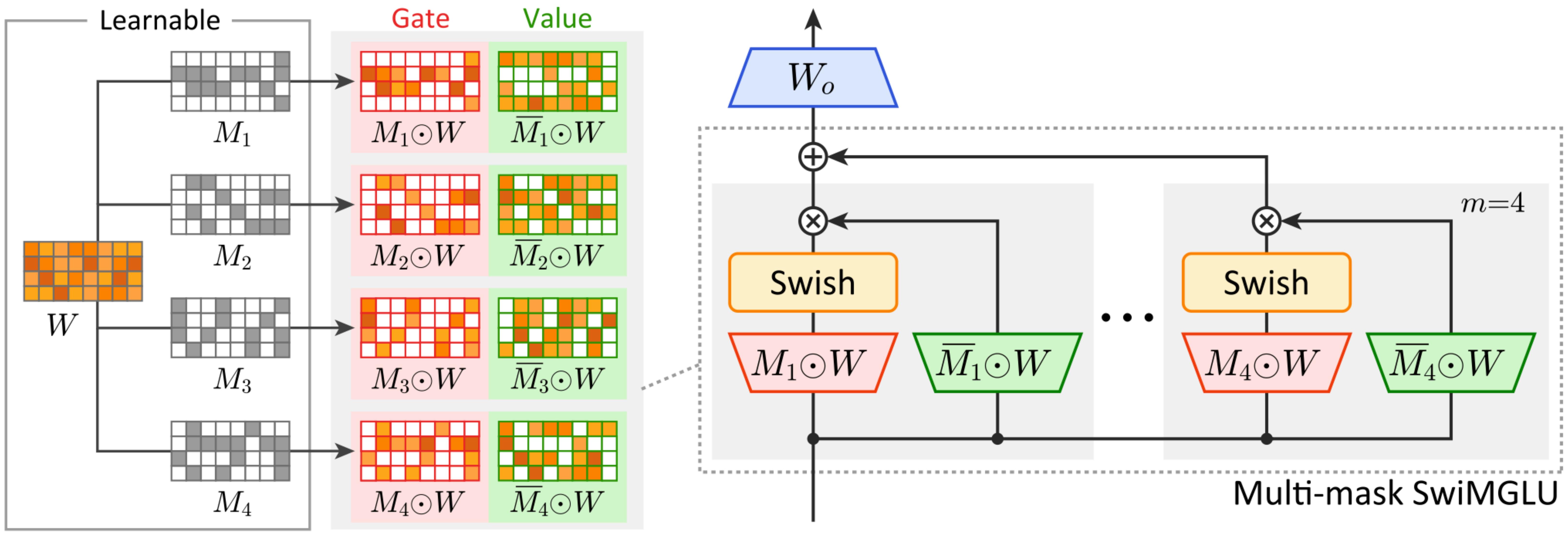


Mask Learning

- Binary mask $M \in \{0,1\}^{d_{in} \times d_{out}}$ is learned via STE (straight-through estimator)
- The mask determines sub-spaces for gating and value within the same weight.

Mask Learning

- Optional multi-mask extension (MoEG) improves expressivity using multiple M_i.

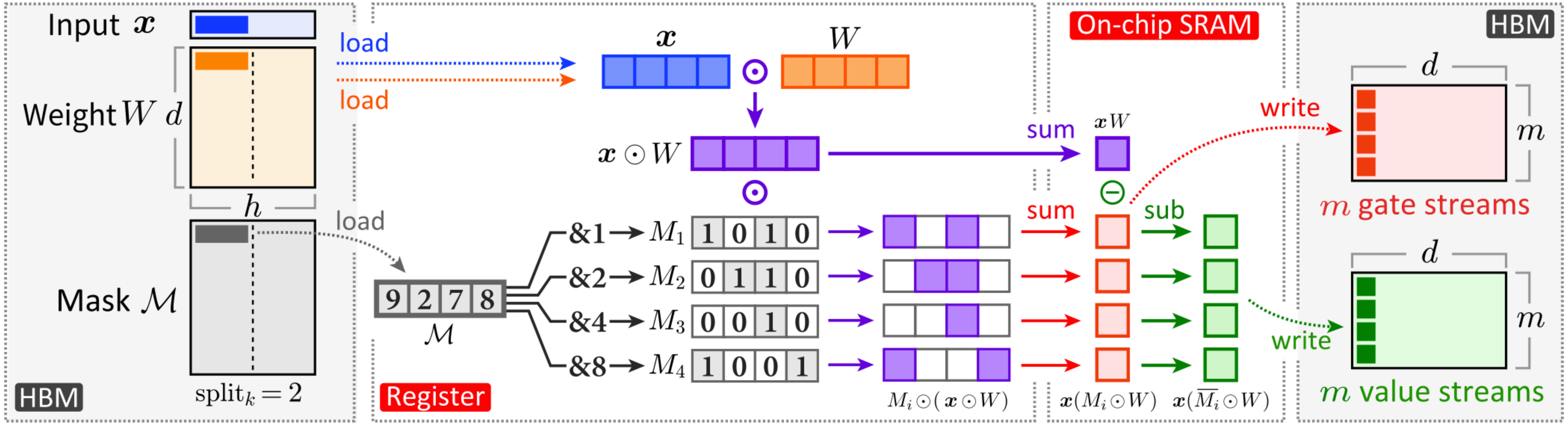


Key Intuition

- Share weights, separate roles: gate/value come from disjoint masked regions.
- Achieves GLU behavior without doubling memory access.
- Compatible with any Transformer FFN (drop-in replacement).

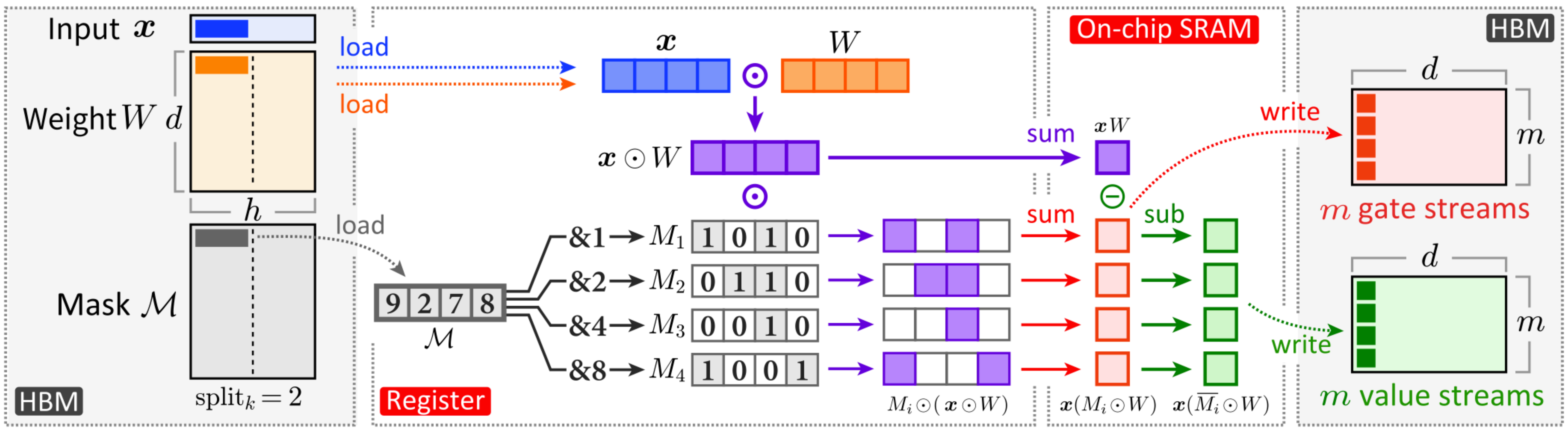
FlashMGLU: Kernel Design Goal

- Fuse weight loading + mask decoding into one kernel.
- Compute gate & value directly on-chip (register/shared memory).



FlashMGLU: Packed Mask Encoding

- Each weight element stores multiple binary masks (e.g., 8) packed into 1 byte.
- Reduces memory load by **up to 47%** vs SwiGLU.

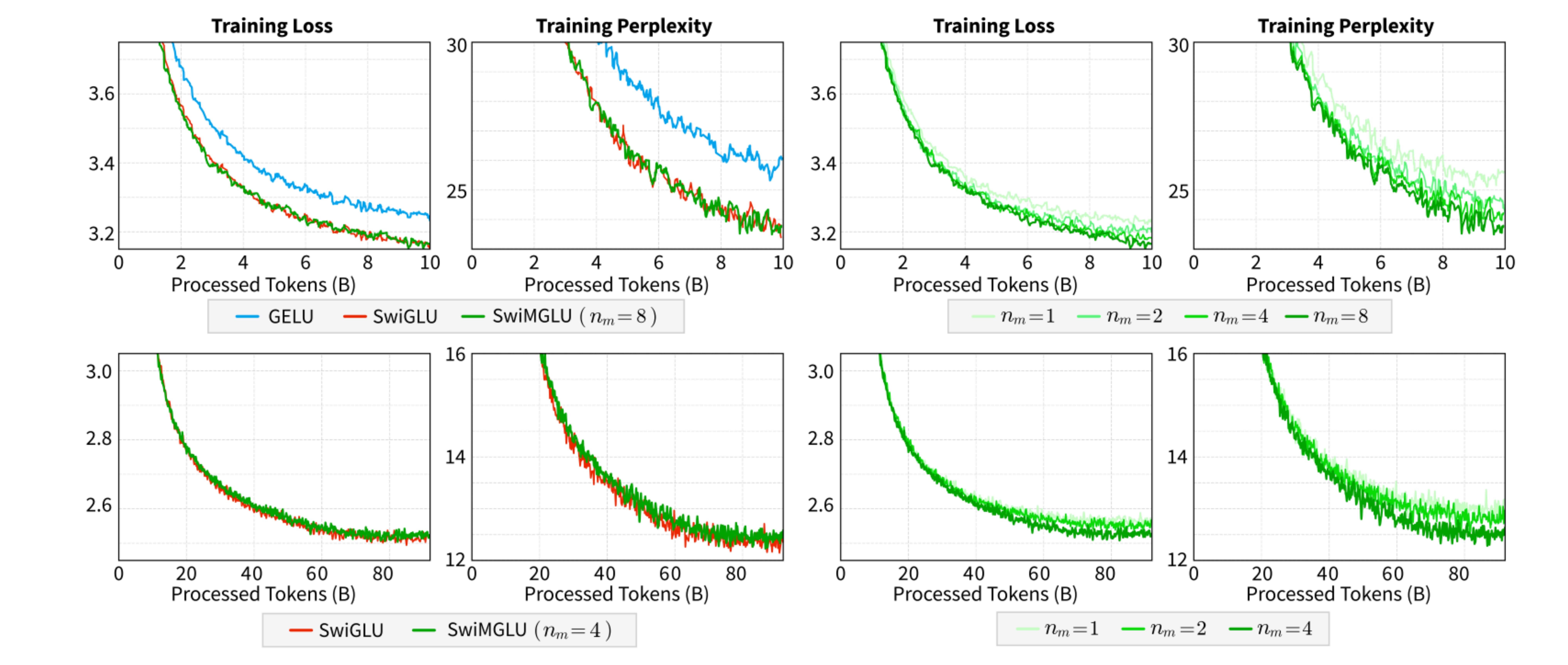


Results

- Accuracy improves with more masks; saturates near $n_m = 4$.
- SwiMGLU ($n_m = 4$) ≈ SwiGLU performance with fewer weights.

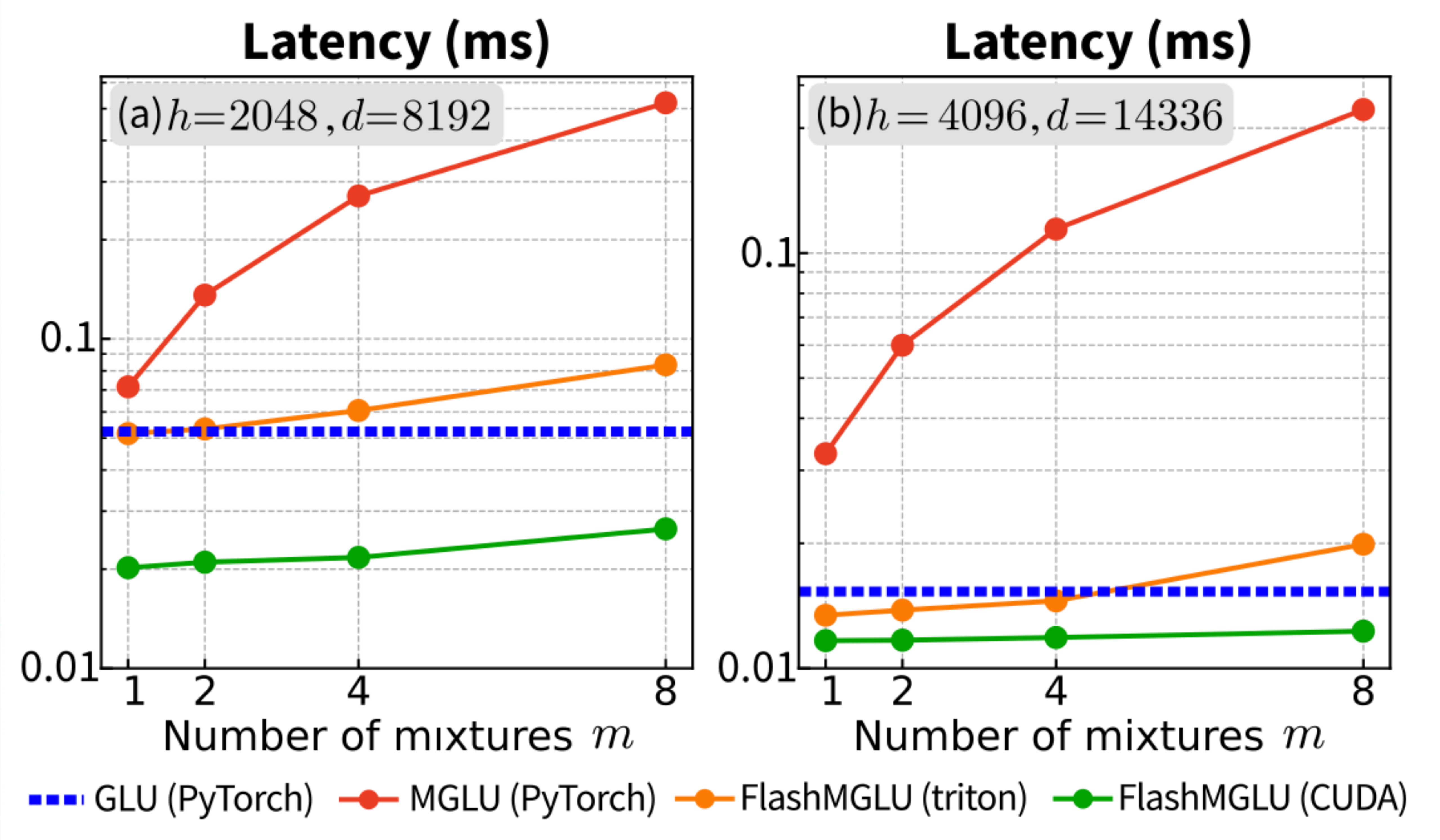
	n_m	#weights	PPL↓	ArcE↑	ArcC↑	HS↑	PiQA↑	SciQ↑	WG↑	Avg ↑
GELU	–	113M	25.8	47.47	19.45	28.09	60.61	64.80	52.41	45.47
SwiGLU	–	141M	23.7	48.15	20.05	28.53	61.43	67.90	51.14	46.20
SwiMGLU	1	113M	25.0	48.91	19.28	28.25	60.72	69.00	50.20	46.06
SwiMGLU	2	113M	24.5	**49.12**	19.97	28.49	60.01	**70.60**	51.53	**46.62**
SwiMGLU	4	113M	23.9	48.99	**20.56**	28.49	**61.70**	69.10	50.04	46.48
SwiMGLU	8	113M	**23.5**	48.65	**20.56**	**28.63**	61.53	68.00	**51.54**	46.49
SwiGLU	–	1.08B	**12.3**	64.94	**28.92**	37.20	69.15	**84.50**	51.30	56.00
SwiMGLU	1	808M	13.0	63.72	27.73	36.20	68.61	83.00	54.30	55.59
SwiMGLU	2	808M	12.7	62.08	26.71	36.52	68.44	84.20	51.85	54.97
SwiMGLU	4	808M	12.4	**65.78**	**28.92**	**37.69**	**69.26**	84.20	**55.25**	**56.85**

Results



Training Loss
Training Perplexity
Training Loss
Training Perplexity
Processed Tokens (B)
GELU
SwiGLU
SwiMGLU ($n_m=8$)
$n_m=1$
$n_m=2$
$n_m=4$
$n_m=8$
SwiGLU
SwiMGLU ($n_m=4$)
$n_m=1$
$n_m=2$
$n_m=4$

Is FlashMGLU fast?



Summary

- MGLU: Shares one weight matrix for gate and value → ~50 % fewer memory reads.
- FlashMGLU: Custom fused kernel achieving > 19× speed-up with no accuracy loss.

Thanks

- DENSO IT LAB Recognition, Control, and Learning Algorithm Collaborative Research Chair (Science Tokyo)
- TSUBAME4.0 supercomputer provided by Institute of Science Tokyo through the HPCI System Research Project (Project ID: hp240170)