Causal Attention with Lookahead Keys

Zhuoqing Song, Peng Sun, Huizhuo Yuan, Quanquan Gu

October XX, 2025

Motivation

- Scaling law runs up against a practical bottleneck — high quality data
- How to improve models' performance under limited token budget?

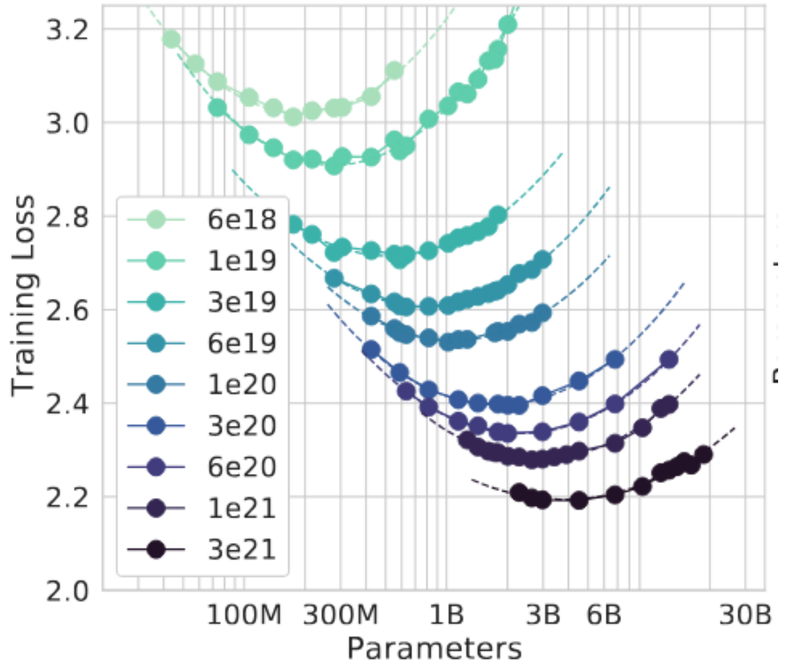


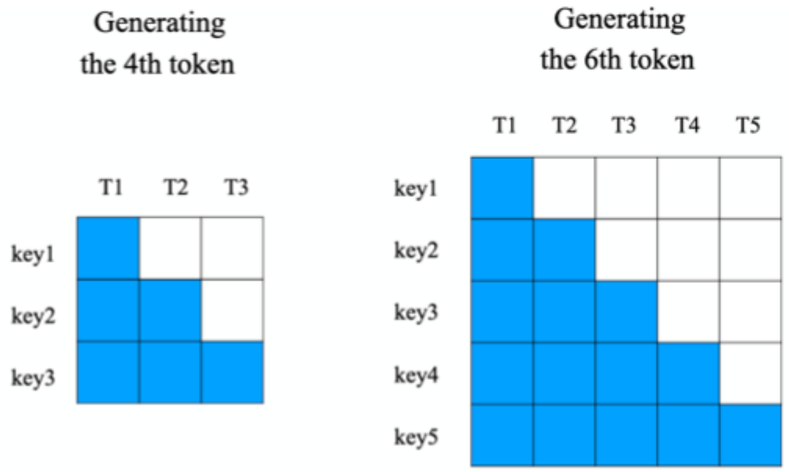
Figure source: https://arxiv.org/pdf/2203.15556

Shortcomings of Standard Causal Attention

- Standard causal attention in recurrent form

$$\text{causal-attention}(\boldsymbol{X}^t) = \text{softmax}\left(\frac{\boldsymbol{q}_t \boldsymbol{K}_t^\top}{\sqrt{d}}\right) \boldsymbol{V}_t = \frac{\sum_{s=1}^{t} \exp\left(\boldsymbol{q}_t \boldsymbol{k}_s^\top / \sqrt{d}\right) \boldsymbol{v}_s}{\sum_{s=1}^{t} \exp\left(\boldsymbol{q}_t \boldsymbol{k}_s^\top / \sqrt{d}\right)} \in \mathbb{R}^{1\times d}.$$

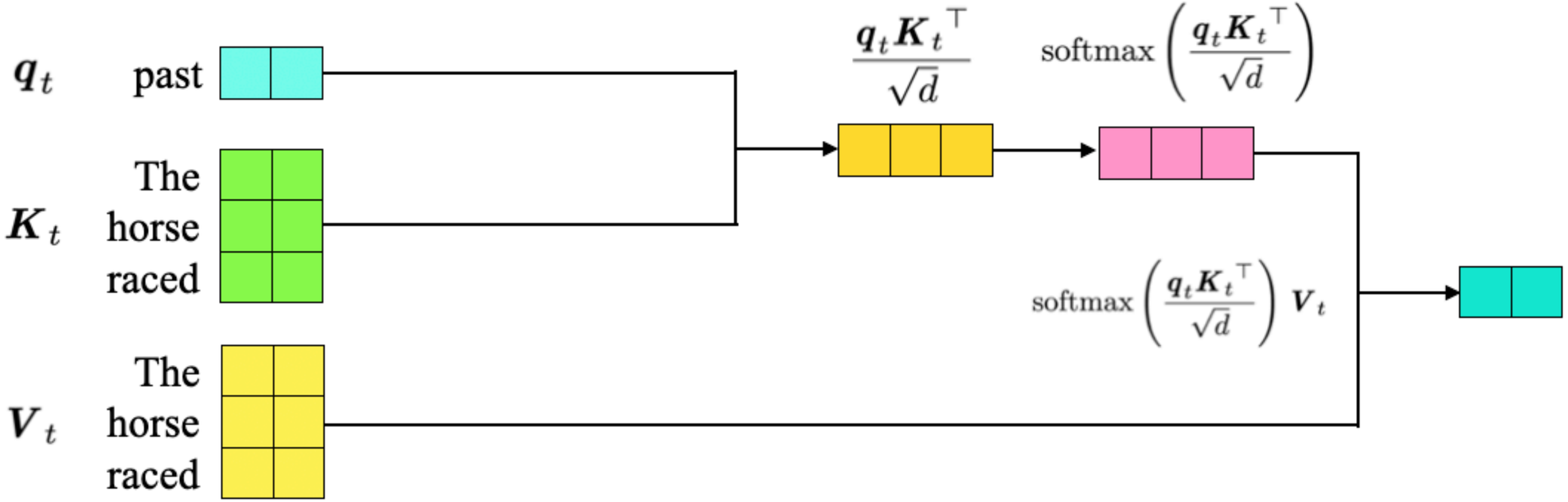
- Receptive fields of keys in standard causal attention



Standard Causal Attention in Recurrent Form

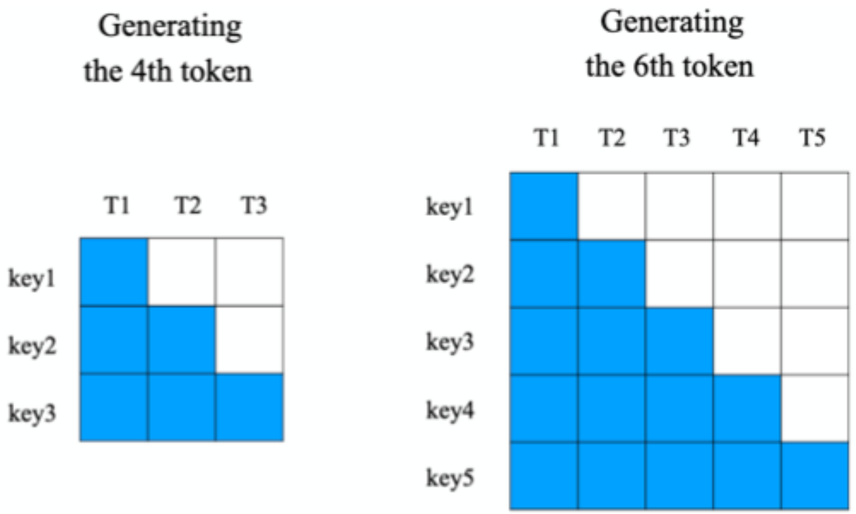
- Example: “The horse raced past the barn fell.”

Input Output



Shortcomings of Standard Causal Attention

- Receptive fields of keys in standard causal attention



- Causal mask blocks each token’s access to its future information
  - Hurt natural language understanding (BERT vs. GPT)
  - Noise from precedent context
  - (Indirect evidence) diffusion LLMs are using bi-directional attention

Examples: Shortcomings of Causal Masking

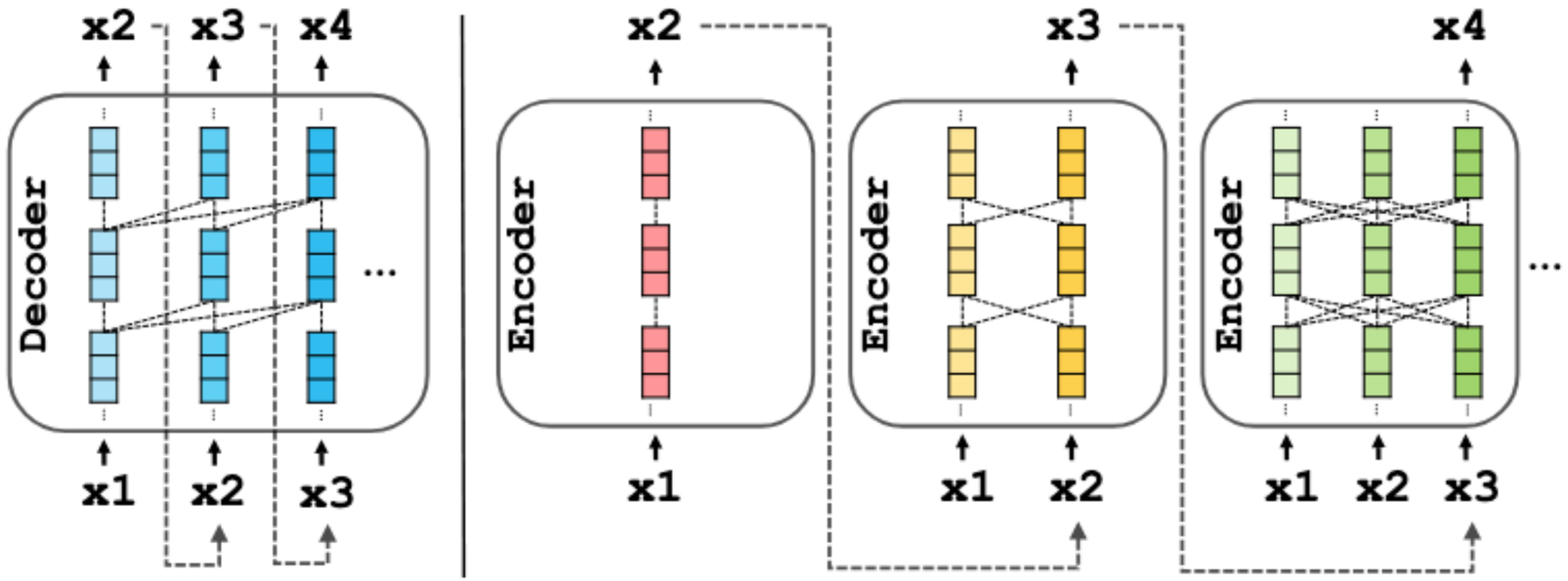
- Garden-path sentence
    - The old man the boat.
    - The complex houses married and single soldiers and their families.
- Question/focus at the end of inputs
- Variable assignment problem
    - x = 1; y = 2; x = 3; y = 5; x = ?

Related work: Sentence Embedding

- Backward Dependency Enhanced LLM
- Echo Embeddings
- Re-Reading

Related work: Encoder-only Next Token Prediction (ENTP)

- NTP by encoder-only Transformers (on KV cache, cubic training complexity)



Decoder-Only Next Token Prediction

Encoder-Only Next Token Prediction (ENTP)

Related work: Selective Attention

- Modify attention matrix, save KV cache

$$\text{SelectiveAttention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}} - F)V$$

$$F = \text{Accumulate}(\text{Constrain}(S))$$

Related work: PaTH Attention

- Data-dependent positional encodings

$$\mathbf{A}_{ij} \propto \exp\Big(\mathbf{k}_j^\top \Big(\prod_{s=j+1}^{i} \mathbf{H}_s\Big)\mathbf{q}_i\Big),$$

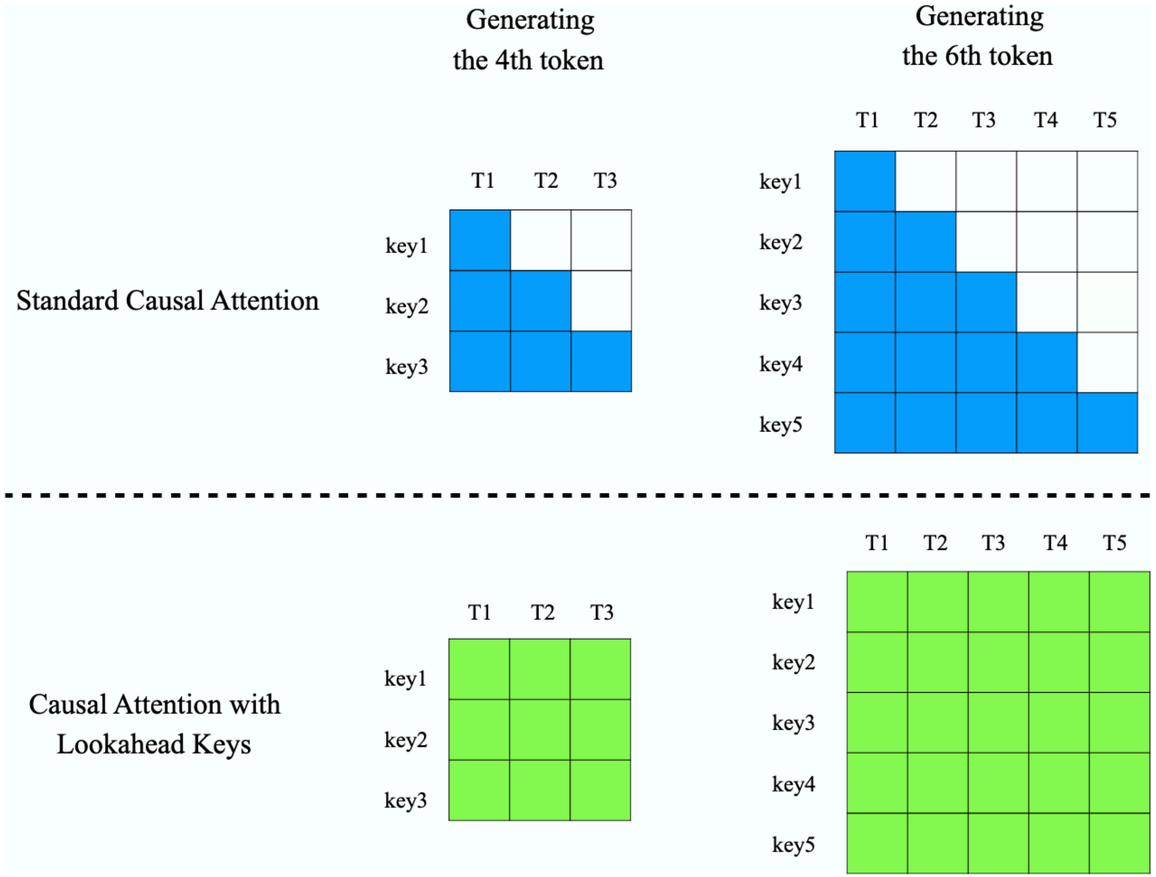
$$\mathbf{H}_t = \mathbf{I} - \beta_t \mathbf{w}_t \mathbf{w}_t^T,$$

- Inference of PaTH

$$\mathbf{k}_i^{(t)} \leftarrow (\mathbf{I} - \beta_t \mathbf{w}_t \mathbf{w}_t^\top)\mathbf{k}_i^{(t-1)} \quad \text{for all } i < t,$$

Our Approach: Expanding Receptive Fields

- Should we expand receptive fields of Q, K or V?



Our Approach: Expanding Receptive Fields

- Should we expand receptive fields of Q, K or V?

  The same reason with using KV cache instead of Q cache!

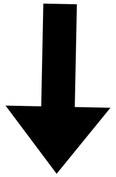
- Why are we updating keys rather than values?

  Updating keys has an equivalent form which enables efficient parallel training.

  If attentions scores are accurate ``enough", information mix in values is unnecessary.

Our Approach: Expanding Receptive Fields

- Expanding receptive fields of keys
- Maintaining efficient training and inference



- Update keys during inference
- Without materializing keys at each position during training

CASTLE in Recurrent Form (Overview)

CASTLE

- Causal-key attention score

$$\boldsymbol{s}_t^C = \frac{\boldsymbol{q}_t^C \boldsymbol{K}_t^{C\top}}{\sqrt{d}} \in \mathbb{R}^{1\times t}$$

- Lookahead-key attention score

$$\boldsymbol{s}_t^U = \frac{\boldsymbol{q}_t^C \boldsymbol{U}^{t\top}}{\sqrt{d}} \in \mathbb{R}^{1\times t}$$

- Attention weights

$$\boldsymbol{p}_t = \mathrm{softmax}\left(\boldsymbol{s}_t^C - \mathrm{SiLU}\left(\boldsymbol{s}_t^U\right)\right) \in \mathbb{R}^{1\times t}$$

- Outputs

$$\mathrm{attention}\left(\boldsymbol{X}^t\right) = \boldsymbol{p}_t \boldsymbol{V}_t^C \in \mathbb{R}^{1\times d}$$

Standard Causal Attention

- Causal-key attention score

$$\boldsymbol{s}_t = \frac{\boldsymbol{q}_t \boldsymbol{K}_t^{\top}}{\sqrt{d}} \in \mathbb{R}^{1\times t}$$

- Attention weights

$$\boldsymbol{p}_t = \mathrm{softmax}\left(\boldsymbol{s}_t\right) \in \mathbb{R}^{1\times t}$$

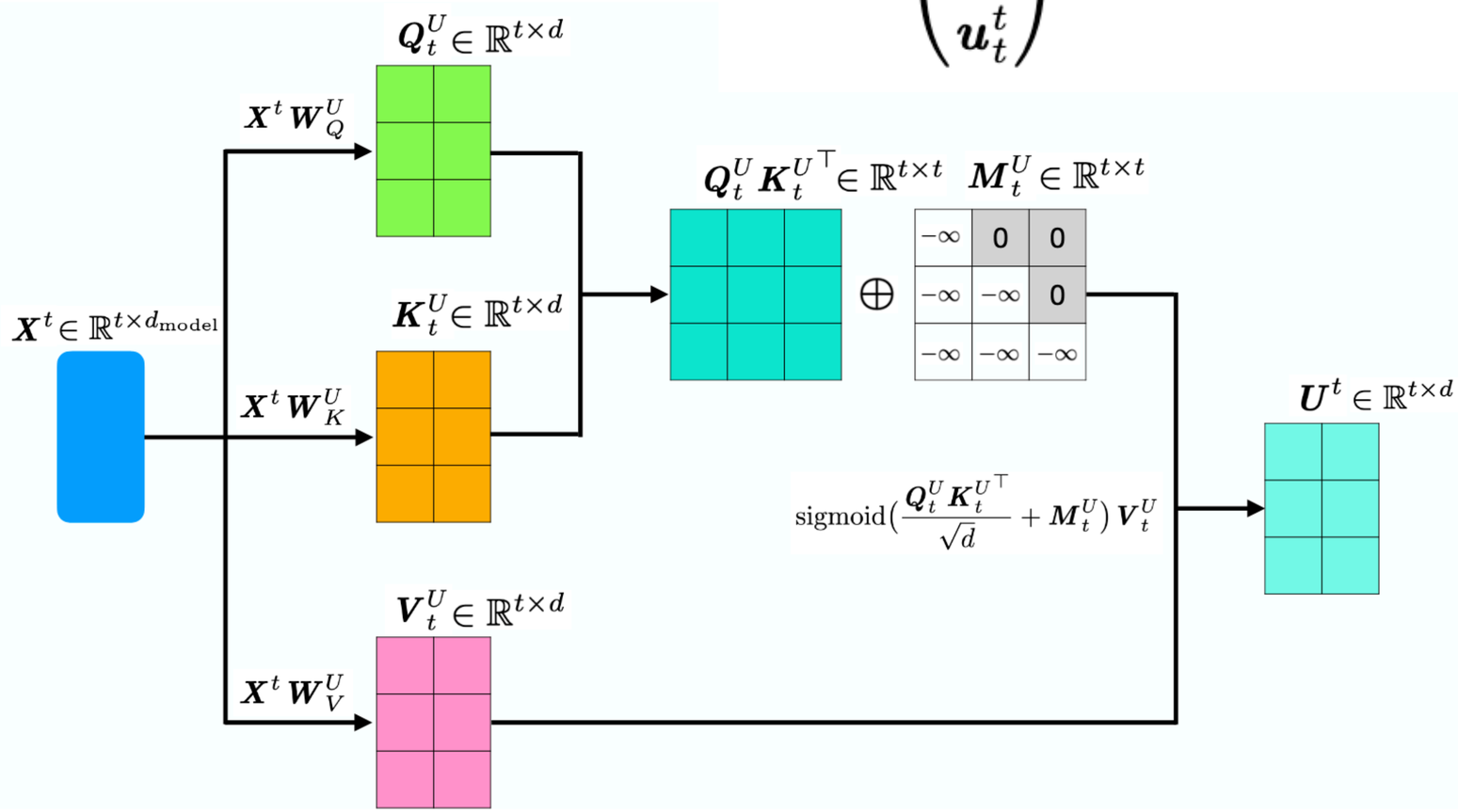
- Outputs

$$\mathrm{attention}\left(\boldsymbol{X}^t\right) = \boldsymbol{p}_t \boldsymbol{V}_t \in \mathbb{R}^{1\times d}$$

Formal Definition of Lookahead Keys

- Similar to attention mechanism

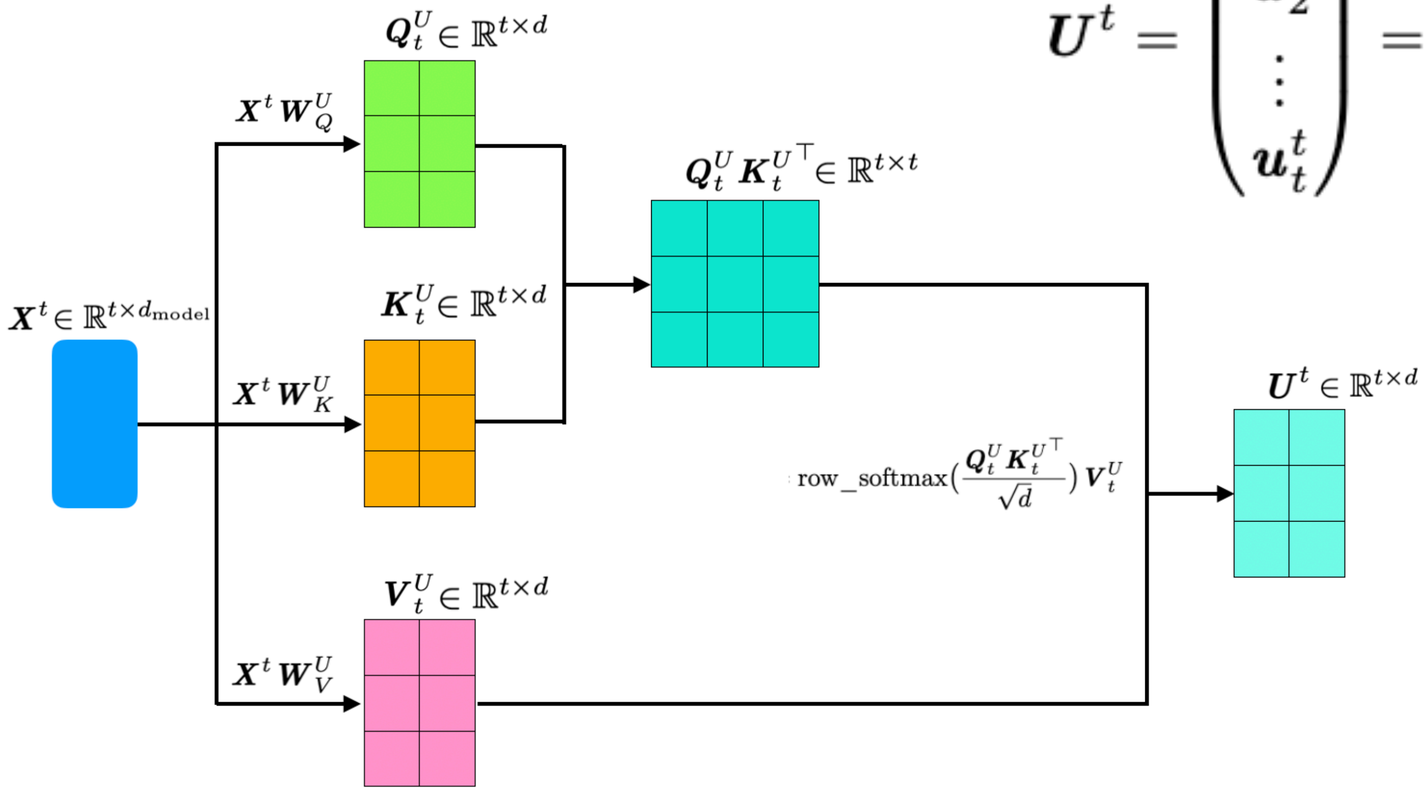
$$\boldsymbol{U}^t = \begin{pmatrix} \boldsymbol{u}_1^t \\ \boldsymbol{u}_2^t \\ \vdots \\ \boldsymbol{u}_t^t \end{pmatrix} = \text{sigmoid}\left(\frac{\boldsymbol{Q}_t^U \boldsymbol{K}_t^{U\top}}{\sqrt{d}} + \boldsymbol{M}_t^U\right) \boldsymbol{V}_t^U \in \mathbb{R}^{t\times d},$$



Failed Attempt of Lookahead keys

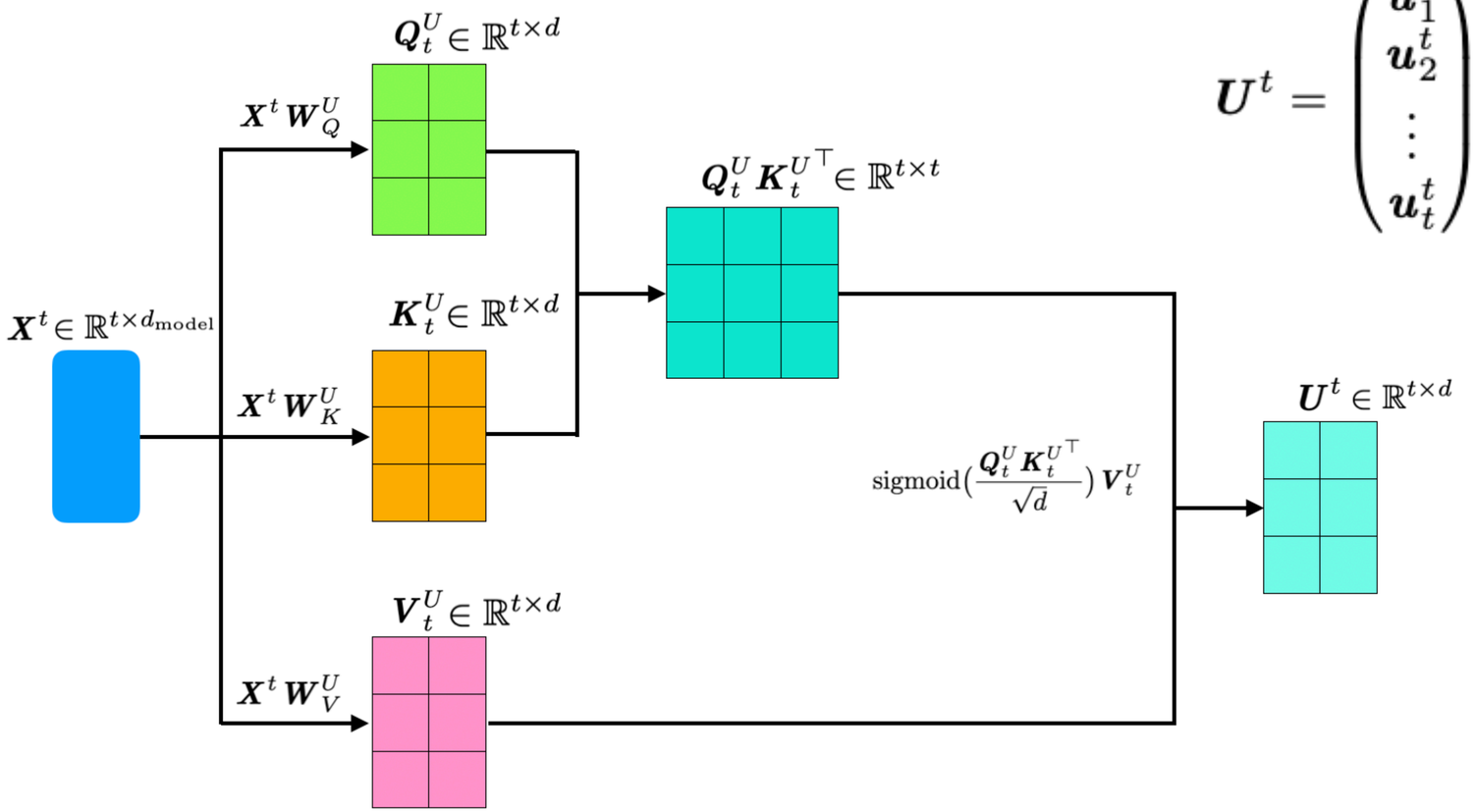
- Encode all information from token 1 to token t: numerically unstable

$$\boldsymbol{U}^t = \begin{pmatrix} \boldsymbol{u}_1^t \\ \boldsymbol{u}_2^t \\ \vdots \\ \boldsymbol{u}_t^t \end{pmatrix} = \mathrm{row\_softmax}\left(\frac{\boldsymbol{Q}_t^U \boldsymbol{K}_t^{U\top}}{\sqrt{d}}\right) \boldsymbol{V}_t^U$$



Failed Attempt of Lookahead keys

- Replace softmax by sigmoid: training instability (blow up easily)

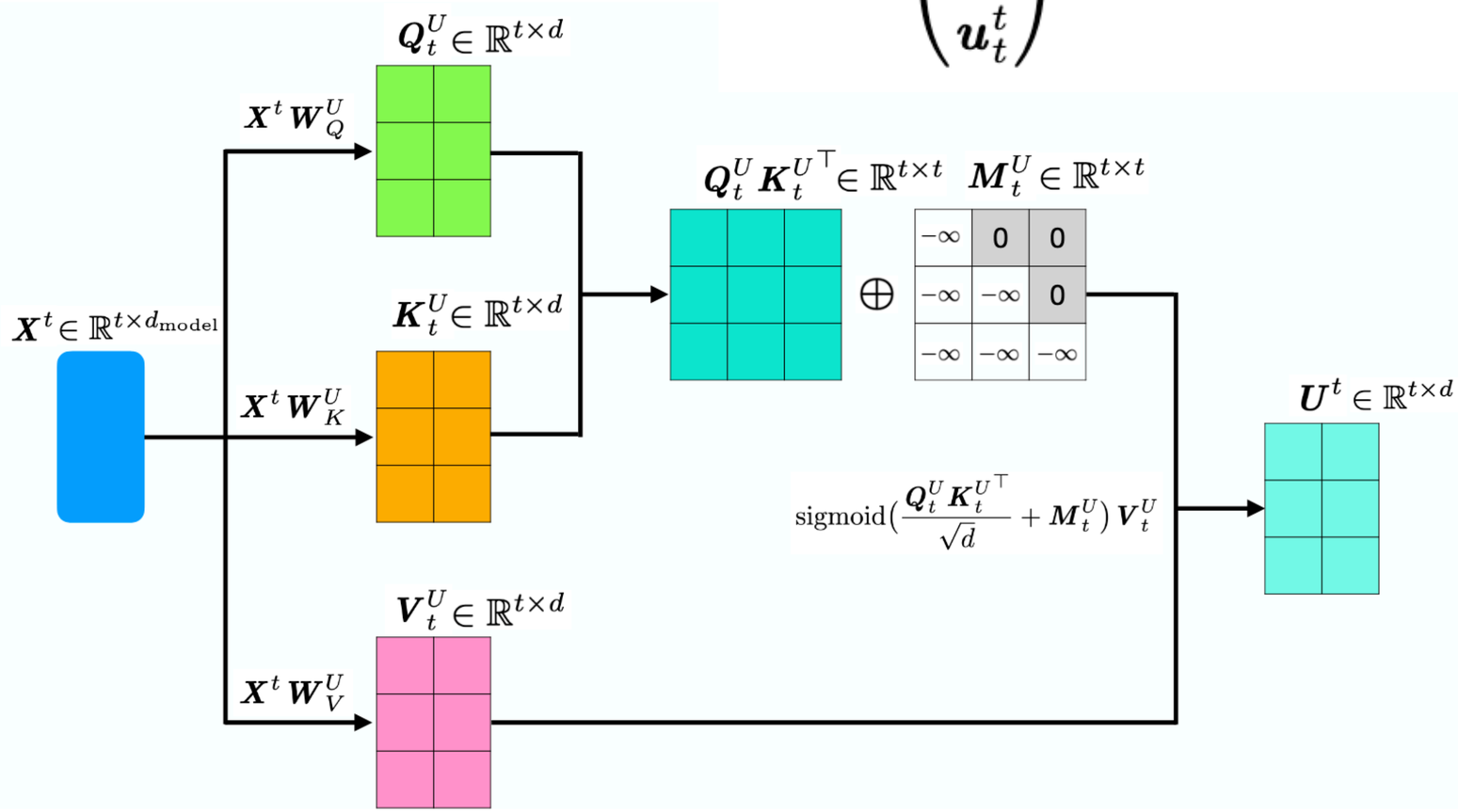


$$U^t = \begin{pmatrix} u_1^t \\ u_2^t \\ \vdots \\ u_t^t \end{pmatrix} = \text{sigmoid}\left(\frac{Q_t^U K_t^{U\top}}{\sqrt{d}}\right) V_t^U \in \mathbb{R}^{t\times d}$$

Formal Definition of Lookahead Keys

- Similar to attention mechanism

$$\boldsymbol{U}^t = \begin{pmatrix} \boldsymbol{u}_1^t \\ \boldsymbol{u}_2^t \\ \vdots \\ \boldsymbol{u}_t^t \end{pmatrix} = \text{sigmoid}\left(\frac{\boldsymbol{Q}_t^U \boldsymbol{K}_t^{U\top}}{\sqrt{d}} + \boldsymbol{M}_t^U\right) \boldsymbol{V}_t^U \in \mathbb{R}^{t\times d},$$



Definition of Causal Keys

- Causal keys are the same as the keys in standard causal attention

$$\boldsymbol{K}_t^C = \begin{pmatrix} \boldsymbol{k}_1^C \\ \boldsymbol{k}_2^C \\ \vdots \\ \boldsymbol{k}_t^C \end{pmatrix} = \boldsymbol{X}^t \boldsymbol{W}_K^C \in \mathbb{R}^{t \times d}$$

Ablation Studies on Causal Keys

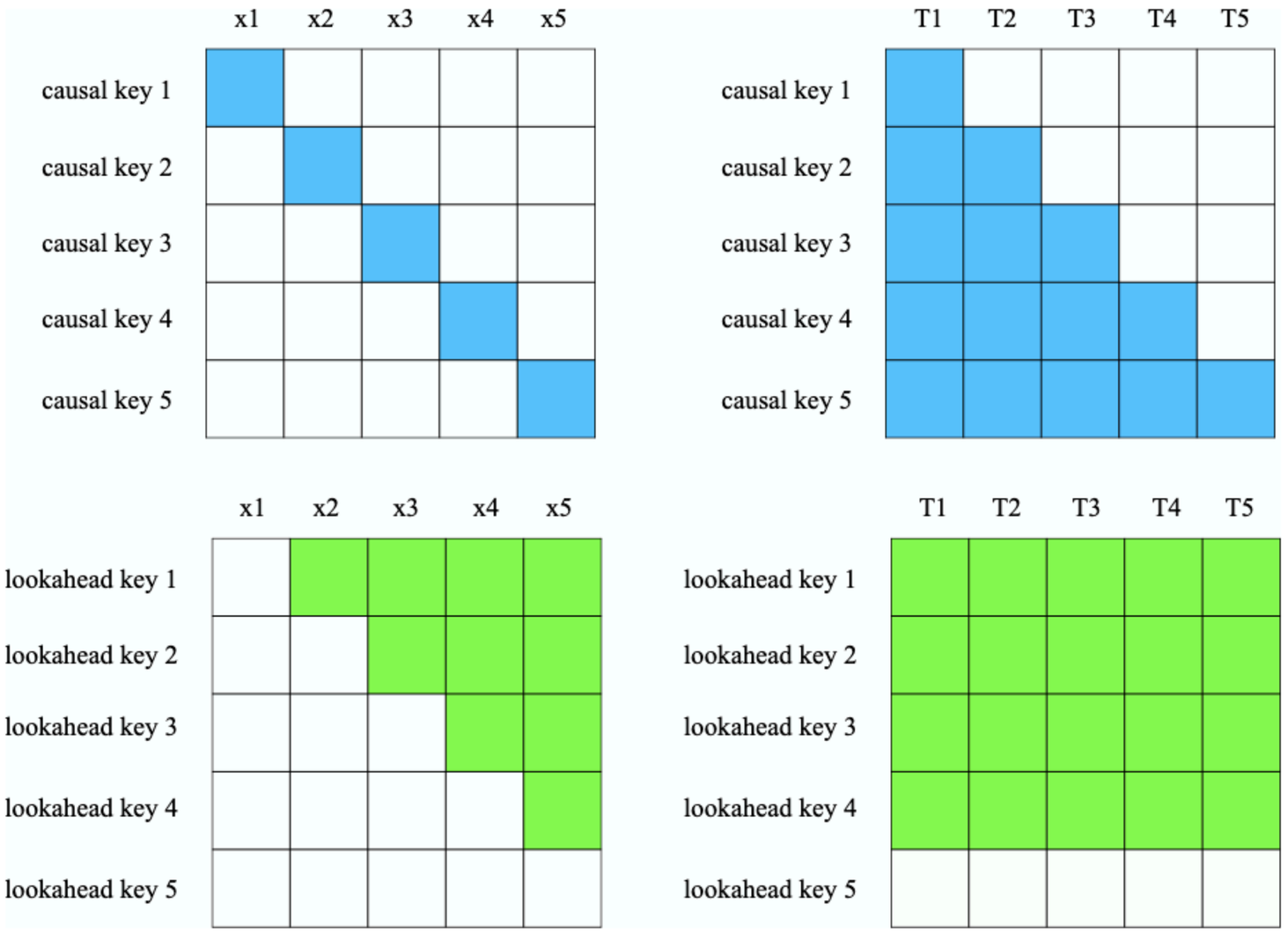
- Whether we need causal keys?
  - Aligning parameters

CASTLE TYPE	n_{params}	n_{layers}	d_{model}	n_{heads}	d
CASTLE	120M	12	768	6	64
CASTLE w/o causal keys	120M	12	768	7	64

  - Training & validation loss

	Train		Eval	
	Loss	PPL	Loss	PPL
CASTLE	**2.913**	**18.417**	**2.920**	**18.541**
CASTLE w/o causal keys	3.006	20.213	3.021	20.505

Comparison between Causal Keys and Lookahead Keys



CASTLE in Recurrent Form

- Causal-key attention score

$$\boldsymbol{s}_t^C = \frac{\boldsymbol{q}_t^C \boldsymbol{K}_t^{C\top}}{\sqrt{d}} \in \mathbb{R}^{1\times t}$$

- Lookahead-key attention score

$$\boldsymbol{s}_t^U = \frac{\boldsymbol{q}_t^C \boldsymbol{U}^{t\top}}{\sqrt{d}} \in \mathbb{R}^{1\times t}$$

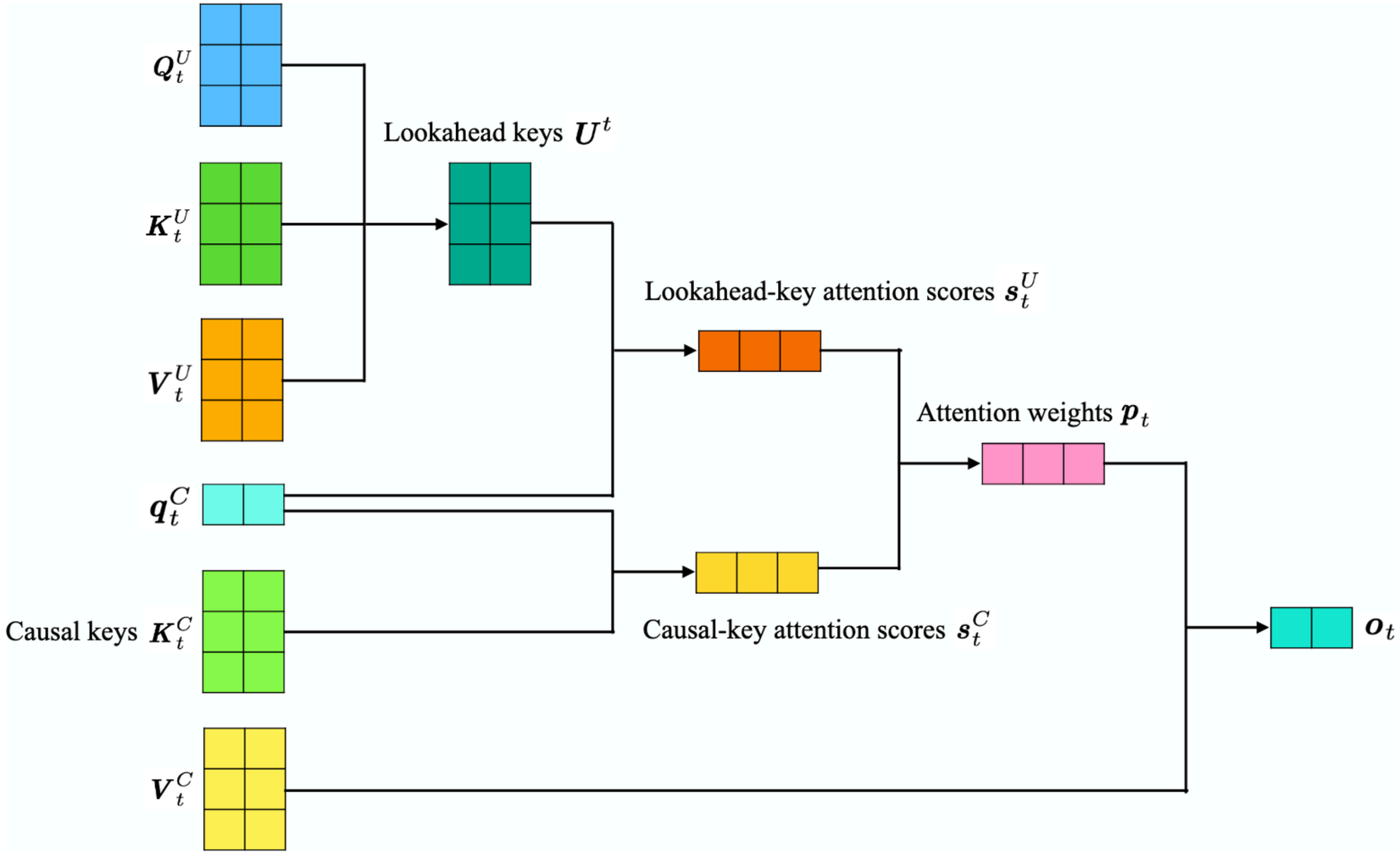
- Attention weights

$$\boldsymbol{p}_t = \mathrm{softmax}\left(\boldsymbol{s}_t^C - \mathrm{SiLU}\left(\boldsymbol{s}_t^U\right)\right) \in \mathbb{R}^{1\times t}$$

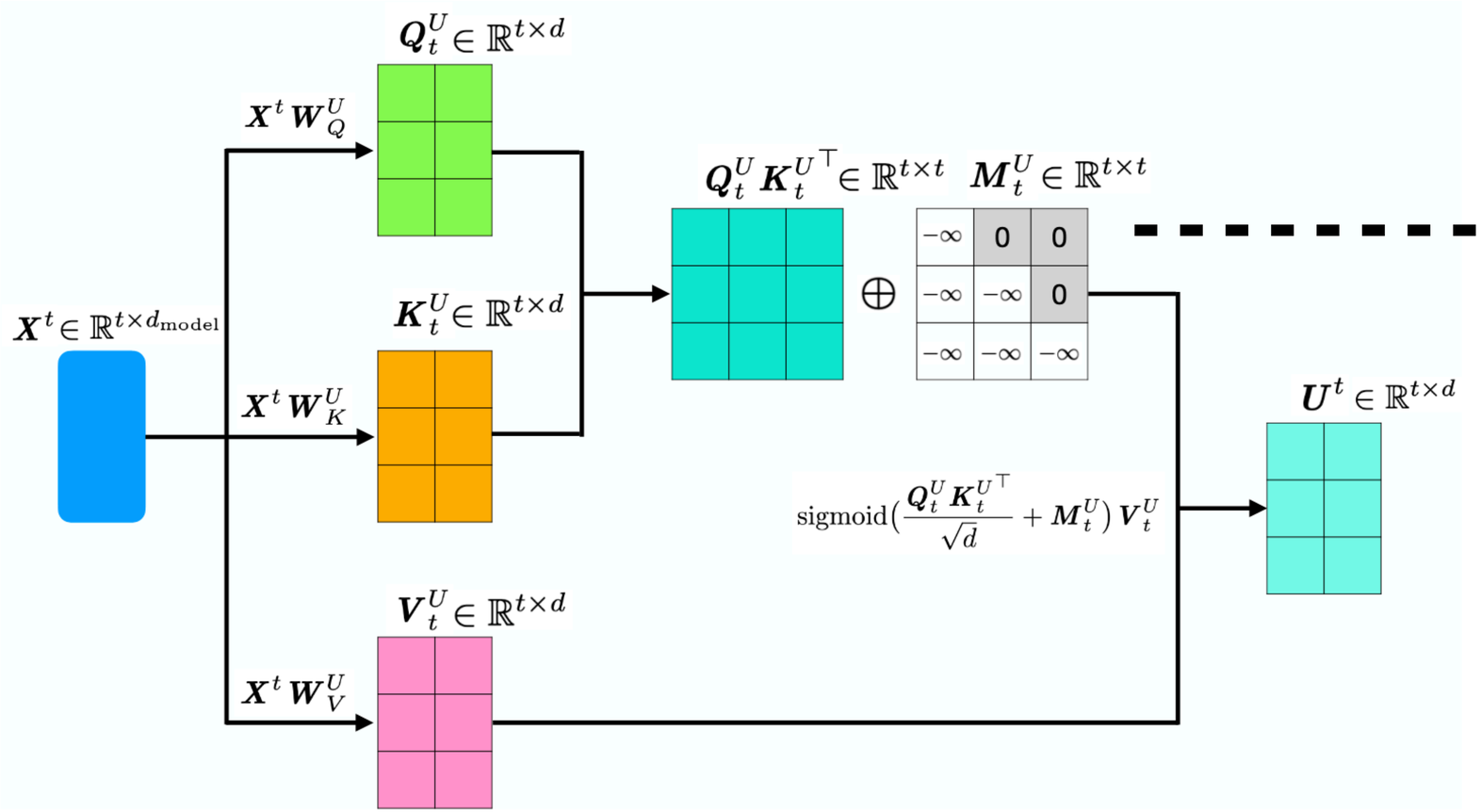
- outputs

$$\mathrm{attention}\left(\boldsymbol{X}^t\right) = \boldsymbol{p}_t \boldsymbol{V}_t^C \in \mathbb{R}^{1\times d}$$

Illustration for CASTLE in Recurrent Form



A variant of CASTLE: CASTLE-SWL



$M_t^U \in \mathbb{R}^{t \times t}$

$-\infty$	0	0	0	$-\infty$	$-\infty$
$-\infty$	$-\infty$	0	0	0	$-\infty$
$-\infty$	$-\infty$	$-\infty$	0	0	0
$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	0
$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0
$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$

Ablation Studies on Sliding Window Sizes in CASTLE-SWL

- Not sensitive in tested range [128, 1024]

		Train		Eval	
	n_{params}	Loss	PPL	Loss	PPL
Baseline-S	160M	2.892	18.037	2.901	18.197
CASTLE-SWL128-S	160M	**2.883**	**17.859**	**2.889**	**17.971**
CASTLE-SWL256-S	160M	2.888	17.952	2.895	18.092
CASTLE-SWL512-S	160M	2.885	17.910	<u>2.892</u>	<u>18.023</u>
CASTLE-SWL1024-S	160M	<u>2.885</u>	<u>17.901</u>	2.892	18.031

		Train		Eval	
	n_{params}	Loss	PPL	Loss	PPL
Baseline-XL	1.310B	2.548	12.779	2.543	12.723
CASTLE-SWL128-XL	1.304B	2.517	12.393	2.513	12.339
CASTLE-SWL256-XL	1.304B	2.514	12.353	2.510	12.300
CASTLE-SWL512-XL	1.304B	**2.506**	**12.255**	**2.503**	**12.217**
CASTLE-SWL1024-XL	1.304B	<u>2.514</u>	<u>12.351</u>	<u>2.509</u>	<u>12.294</u>

$$\boldsymbol{M}_t^U \in \mathbb{R}^{t \times t}$$

$-\infty$	0	0	0	$-\infty$	$-\infty$
$-\infty$	$-\infty$	0	0	0	$-\infty$
$-\infty$	$-\infty$	$-\infty$	0	0	0
$-\infty$	$-\infty$	$-\infty$	$-\infty$	0	0
$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0
$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$

Ablation Studies on SiLU in softmax

- What roles does SiLU play in the combining step? Training stability
- Removing SiLU will cause blowing up when training XL models for both CASTLE and CASTLE-SWL
- Reducing lr can stabilize training but degrade performance severely

		Train		Eval	
	Learning Rate	Loss	PPL	Loss	PPL
CASTLE-SWL-XL	2×10^{-4}	2.506	12.255	2.503	12.217
CASTLE-SWL-XL w/o SiLU	1×10^{-4}	2.523	12.468	2.520	12.424
CASTLE-SWL-XL w/o SiLU	5×10^{-5}	2.571	13.084	2.571	13.075

Efficient Parallel Training

- Revisit the definition of lookahead keys

$$\boldsymbol{U}^t = \begin{pmatrix} \boldsymbol{u}_1^t \\ \boldsymbol{u}_2^t \\ \vdots \\ \boldsymbol{u}_t^t \end{pmatrix} = \text{sigmoid}\left(\frac{\boldsymbol{Q}_t^U \boldsymbol{K}_t^{U\top}}{\sqrt{d}} + \boldsymbol{M}_t^U\right) \boldsymbol{V}_t^U \in \mathbb{R}^{t\times d}, \quad \boldsymbol{s}_t^U = \frac{\boldsymbol{q}_t^C \boldsymbol{U}^{t\top}}{\sqrt{d}} \in \mathbb{R}^{1\times t}, \quad \boldsymbol{p}_t = \text{softmax}\left(\boldsymbol{s}_t^C - \text{SiLU}\left(\boldsymbol{s}_t^U\right)\right) \in \mathbb{R}^{1\times t}$$

$$\text{attention}\left(\boldsymbol{X}^t\right) = \boldsymbol{p}_t \boldsymbol{V}_t^C \in \mathbb{R}^{1\times d}.$$

- Equivalent mathematical formulation
  - Lookahead-keys attention scores

$$\boldsymbol{S}^U = \left(\frac{\boldsymbol{Q}^C \boldsymbol{V}^{U\top}}{\sqrt{d}} \odot \widetilde{\boldsymbol{M}}^C\right) \left(\text{sigmoid}\left(\frac{\boldsymbol{Q}^U \boldsymbol{K}^{U\top}}{\sqrt{d}} + \boldsymbol{M}^U\right)\right)^\top$$

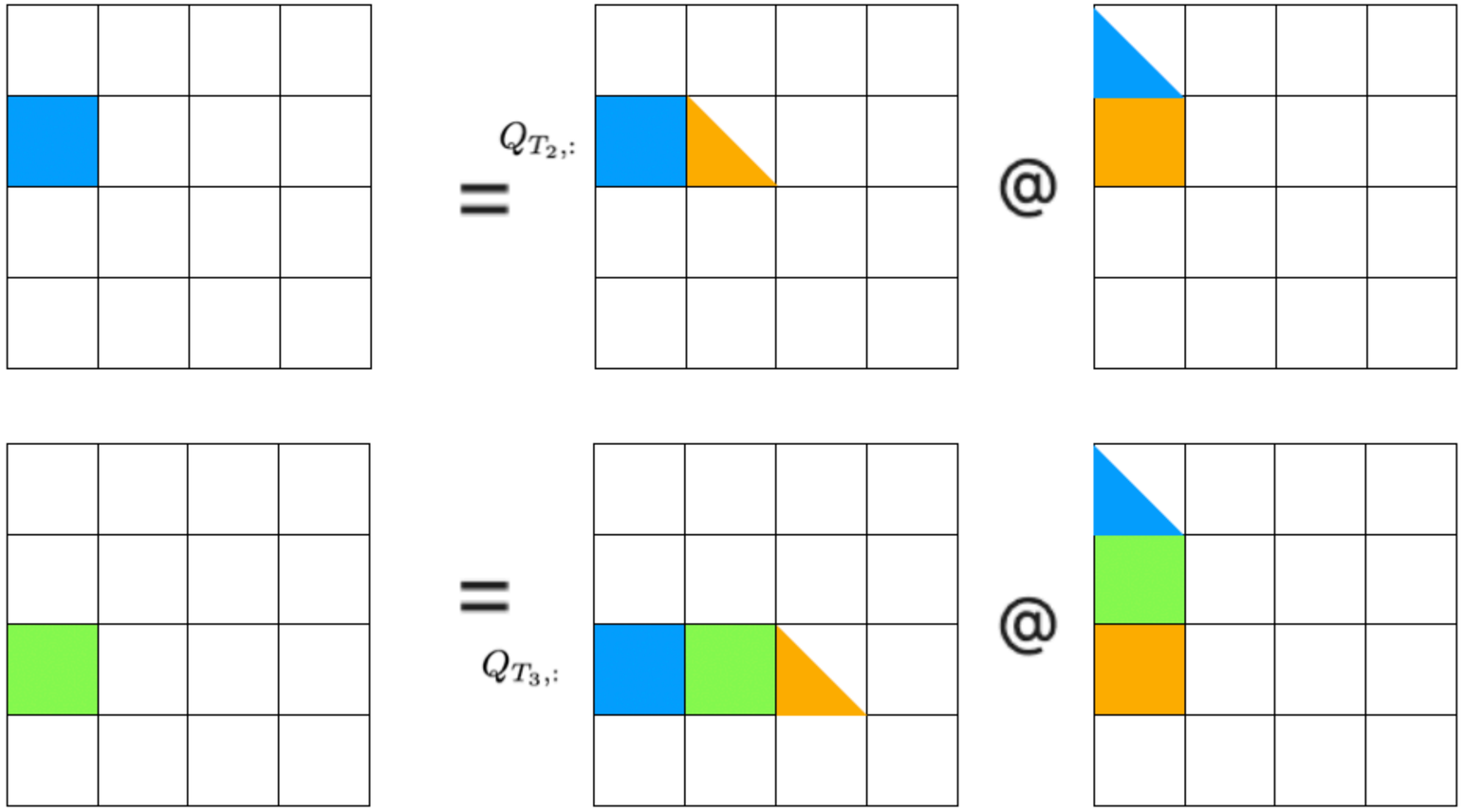
  - Parallel form of attention outputs

$$Attention(\boldsymbol{X}^L) = row\_softmax\left(\frac{\boldsymbol{Q}^C \boldsymbol{K}^{C\top}}{\sqrt{d}} + \boldsymbol{M}^C - \text{SiLU}(\boldsymbol{S}^U)\right) \boldsymbol{V}^C.$$

Efficient Parallel Training

$$\boldsymbol{S}^U \qquad \frac{\boldsymbol{Q}^C \boldsymbol{V}^{U\top}}{\sqrt{d}} \odot \widetilde{\boldsymbol{M}}^C \qquad \left(\text{sigmoid}\left(\frac{\boldsymbol{Q}^U \boldsymbol{K}^{U\top}}{\sqrt{d}} + \boldsymbol{M}^U\right)\right)^\top$$

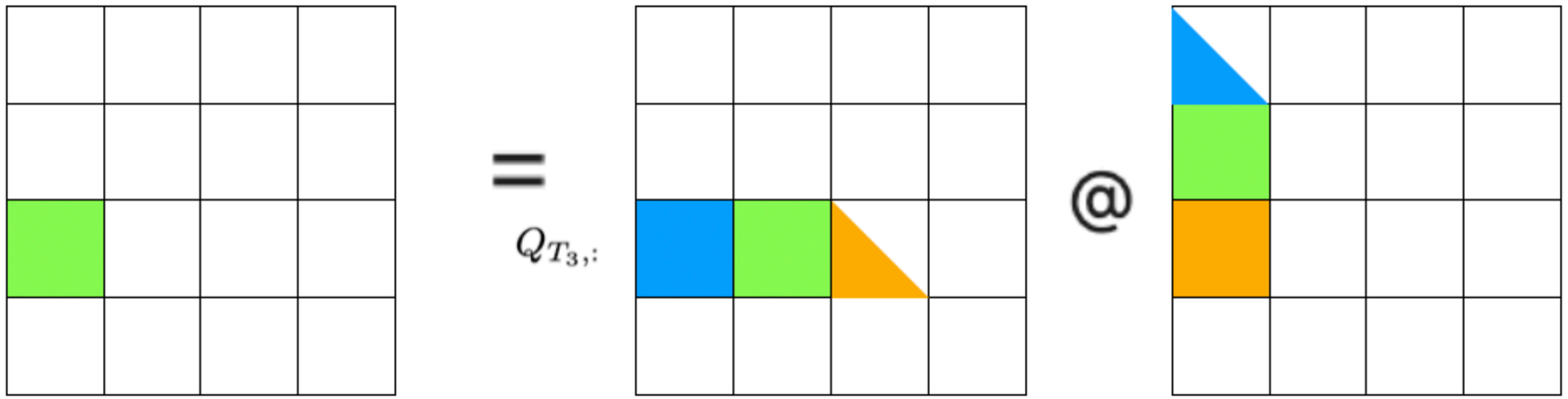
$V_{T_1,:}^\top \; V_{T_2,:}^\top \; V_{T_3,:}^\top \; V_{T_4,:}^\top$



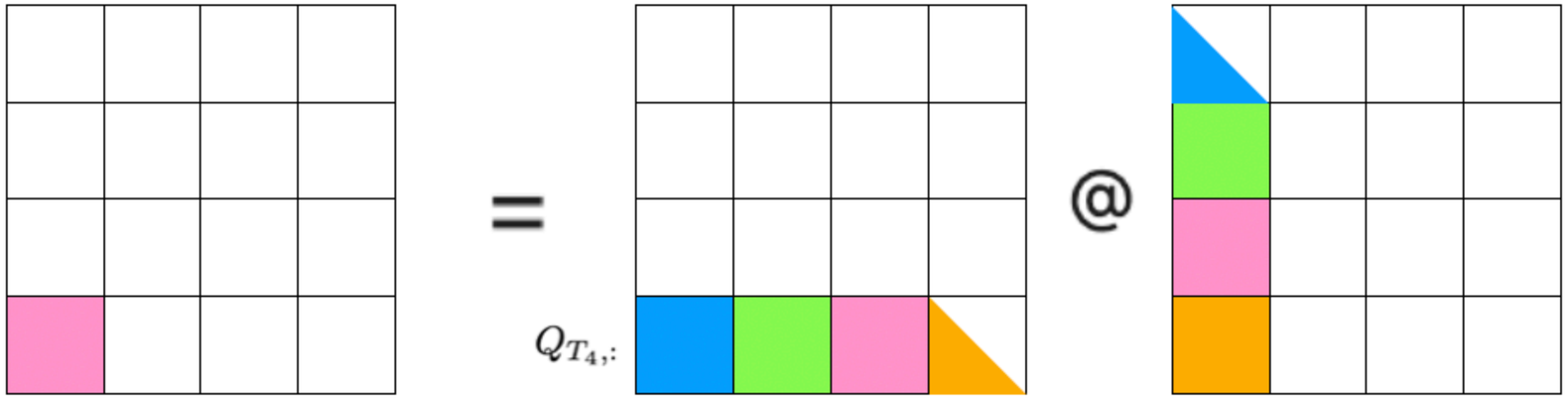
Efficient Parallel Training

$$S^U \qquad \frac{Q^C V^{U\top}}{\sqrt{d}} \odot \widetilde{M}^C \qquad \left(\text{sigmoid}\left(\frac{Q^U K^{U\top}}{\sqrt{d}} + M^U\right)\right)^\top$$

$V_{T_1,:}^\top \; V_{T_2,:}^\top \; V_{T_3,:}^\top \; V_{T_4,:}^\top$



$Q_{T_3,:}$



$Q_{T_4,:}$

Efficient Parallel Training

- Lookahead-key attention scores

$$\boldsymbol{S}^U_{T_{j+k},T_j} = \boldsymbol{Q}^C_{T_{j+k},:} \sum_{i=j}^{j+k-1} \frac{\boldsymbol{V}^U_{T_i,:}}{\sqrt{d}}^\top \left(\text{sigmoid}\left(\frac{\boldsymbol{Q}^U_{T_j,:}\boldsymbol{K}^U_{T_i,:}}{\sqrt{d}}^\top + \boldsymbol{M}^U_{T_j,T_i}\right)\right)^\top$$
$$+ \left(\frac{\boldsymbol{Q}^C_{T_{j+k},:}\boldsymbol{V}^U_{T_{j+k},:}}{\sqrt{d}}^\top \odot \widetilde{\boldsymbol{M}}^C_{T_{j+k},T_{j+k}}\right)\left(\text{sigmoid}\left(\frac{\boldsymbol{Q}^U_{T_j,:}\boldsymbol{K}^U_{T_{j+k},:}}{\sqrt{d}}^\top + \boldsymbol{M}^U_{T_j,T_{j+k}}\right)\right)^\top$$

- Utilizing masked low-rank structure
  - Update auxiliary variable

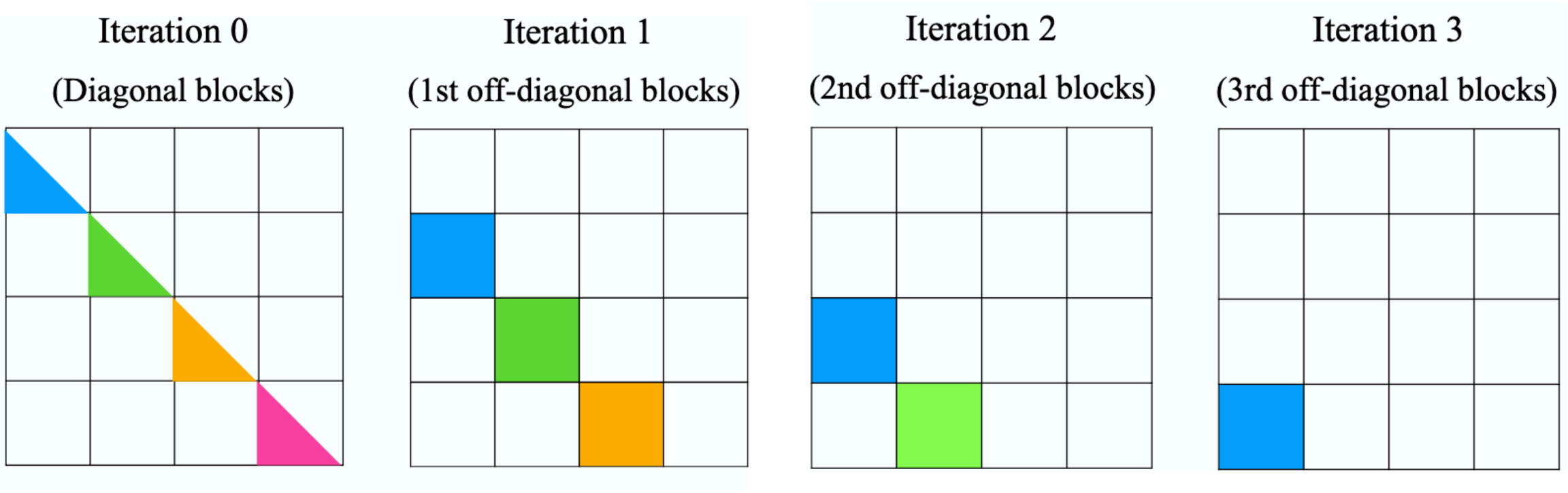
$$\boldsymbol{D}^{(k)}_{:,T_j} = \boldsymbol{D}^{(k-1)}_{:,T_j} + \boldsymbol{V}^U_{T_{j+k-1},:}{}^\top \left(\text{sigmoid}\left(\frac{\boldsymbol{Q}^U_{T_j,:}\boldsymbol{K}^U_{T_{j+k-1},:}}{\sqrt{d}}^\top + \boldsymbol{M}^U_{T_j,T_{j+k-1}}\right)\right)^\top$$

  - Compute lookahead-key attention scores

$$\boldsymbol{S}^U_{T_{j+k},T_j} = \frac{\boldsymbol{Q}^C_{T_{j+k},:}\boldsymbol{D}^{(k)}_{:,T_j}}{\sqrt{d}} + \left(\frac{\boldsymbol{Q}^C_{T_{j+k},:}\boldsymbol{V}^U_{T_{j+k},:}}{\sqrt{d}}^\top \odot \widetilde{\boldsymbol{M}}^C_{T_{j+k},T_{j+k}}\right)\left(\text{sigmoid}\left(\frac{\boldsymbol{Q}^U_{T_j,:}\boldsymbol{K}^U_{T_{j+k},:}}{\sqrt{d}}^\top + \boldsymbol{M}^U_{T_j,T_{j+k}}\right)\right)^\top$$

Efficient Parallel Training

- Parallel scheme of forward pass



Efficient Inference with UQ-KV Cache

- Updating step: store U, Q

$$\boldsymbol{U}^t = \begin{pmatrix} \boldsymbol{U}^{t-1} + \text{sigmoid}\left(\frac{\boldsymbol{Q}_{t-1}^U \boldsymbol{k}_t^{U\top}}{\sqrt{d}}\right)\boldsymbol{v}_t^U \\ \mathbf{0}^{1\times d} \end{pmatrix}$$

- Combining step: store K, V

$$\boldsymbol{s}_t^C = \frac{\boldsymbol{q}_t^C \boldsymbol{K}_t^{C\top}}{\sqrt{d}} \in \mathbb{R}^{1\times t}$$

$$\boldsymbol{s}_t^U = \frac{\boldsymbol{q}_t^C \boldsymbol{U}^{t\top}}{\sqrt{d}} \in \mathbb{R}^{1\times t}$$

$$\boldsymbol{p}_t = \text{softmax}\left(\boldsymbol{s}_t^C - \text{SiLU}\left(\boldsymbol{s}_t^U\right)\right) \in \mathbb{R}^{1\times t}$$

$$\text{attention}\left(\boldsymbol{X}^t\right) = \boldsymbol{p}_t \boldsymbol{V}_t^C \in \mathbb{R}^{1\times d}$$

Experiment Setup

- Model Architecture

$$\boldsymbol{Y}^{(l)} = \text{MultiHead-Attention}(\text{RMSNorm}(\boldsymbol{X}^{(l)}))$$
$$\boldsymbol{X}^{(l+1)} = \text{SwiGLU}(\text{RMSNorm}(\boldsymbol{Y}^{(l)})),$$

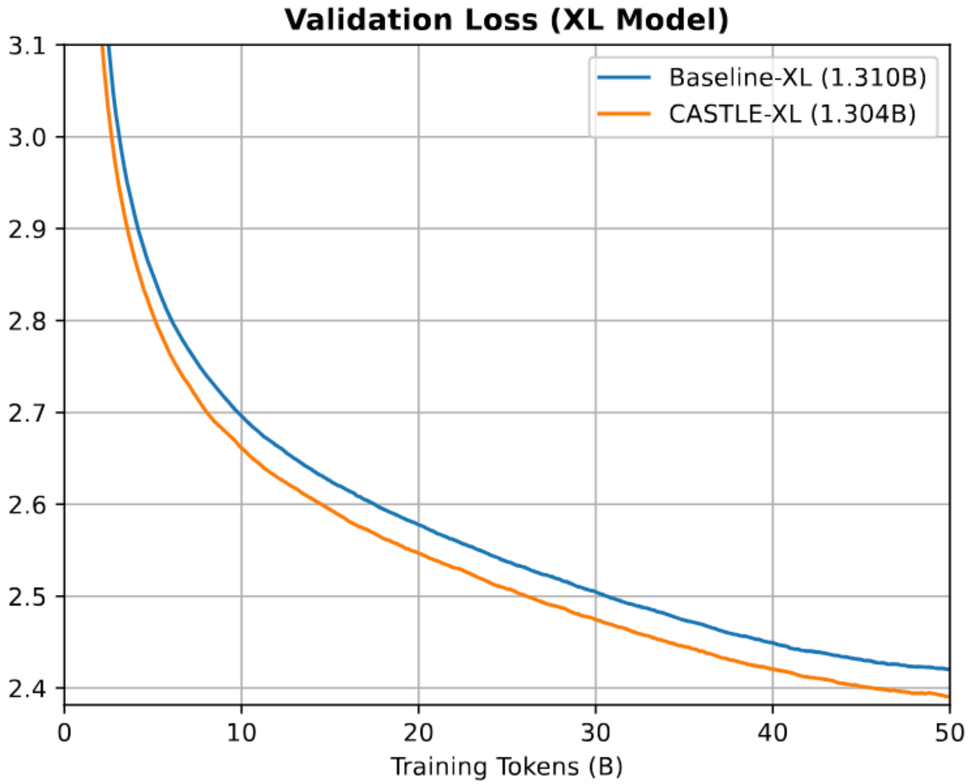
- Model configuration and training recipe

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d	Batch Size	Learning Rate
Baseline-S	160M	12	896 (=14 * 64)	14	64	0.5M	6.0×10^{-4}
CASTLE-S	160M	12	896	8	64		
Baseline-M	353M	24	1024 (=16 * 64)	16	64	0.5M	3.0×10^{-4}
CASTLE-M	351M	24	1024	9	64		
Baseline-L	756M	24	1536 (=16 * 96)	16	96	0.5M	2.5×10^{-4}
CASTLE-L	753M	24	1536	9	96		
Baseline-XL	1.310B	24	2048 (=16 * 128)	16	128	0.5M	2.0×10^{-4}
CASTLE-XL	1.304B	24	2048	9	128		

Experimental Results

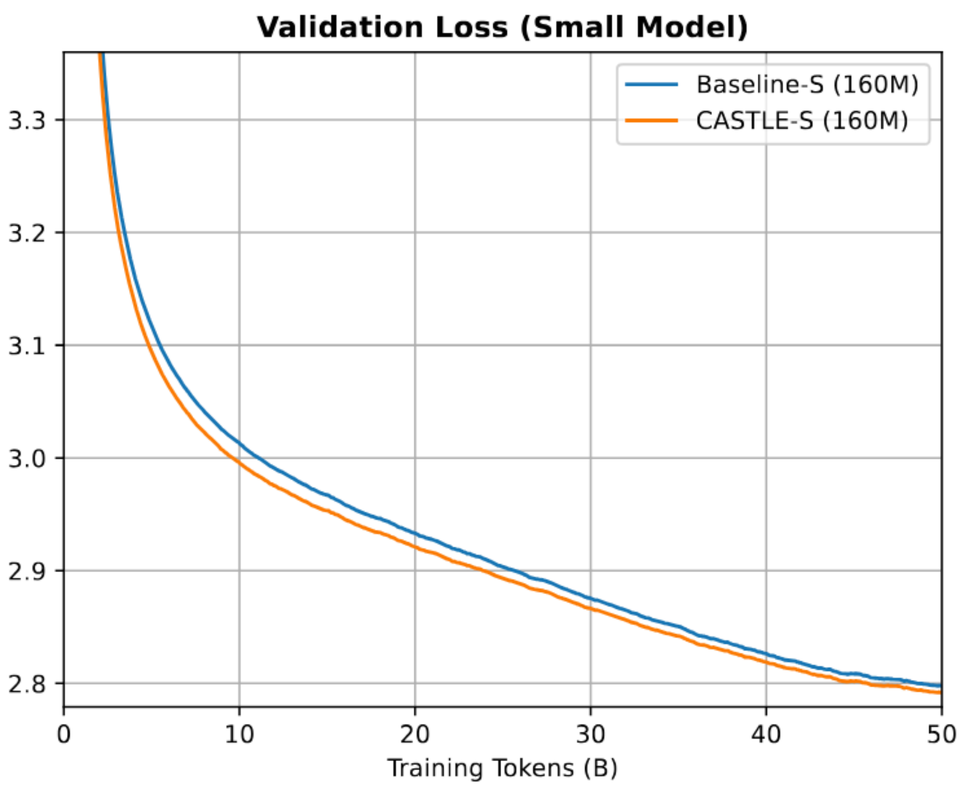
- Training & Validation Loss

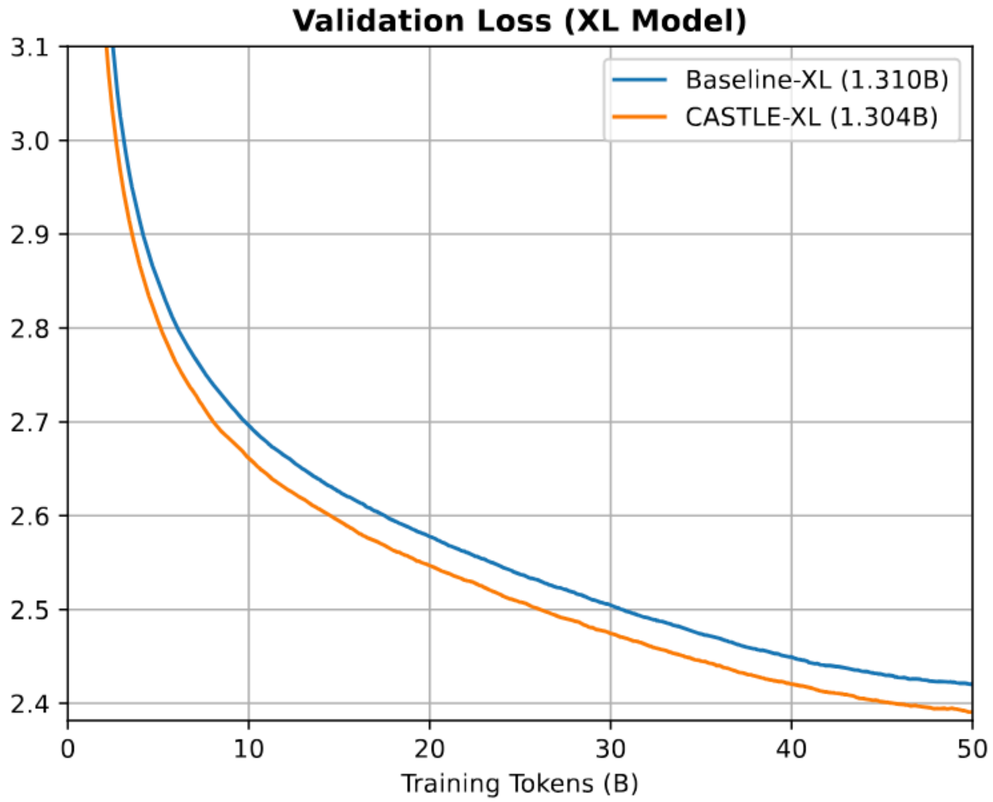
	Train		Eval	
	Loss	PPL	Loss	PPL
Baseline-S	2.795	16.364	2.798	16.411
CASTLE-S	2.789	16.259	2.792	16.315
Baseline-M	2.641	14.030	2.639	14.004
CASTLE-M	**2.616**	**13.684**	**2.615**	**13.665**
Baseline-L	2.513	12.346	2.507	12.269
CASTLE-L	**2.476**	**11.897**	**2.472**	**11.840**
Baseline-XL	2.430	11.360	2.426	11.309
CASTLE-XL	**2.401**	**11.031**	**2.391**	**10.922**



Experimental Results

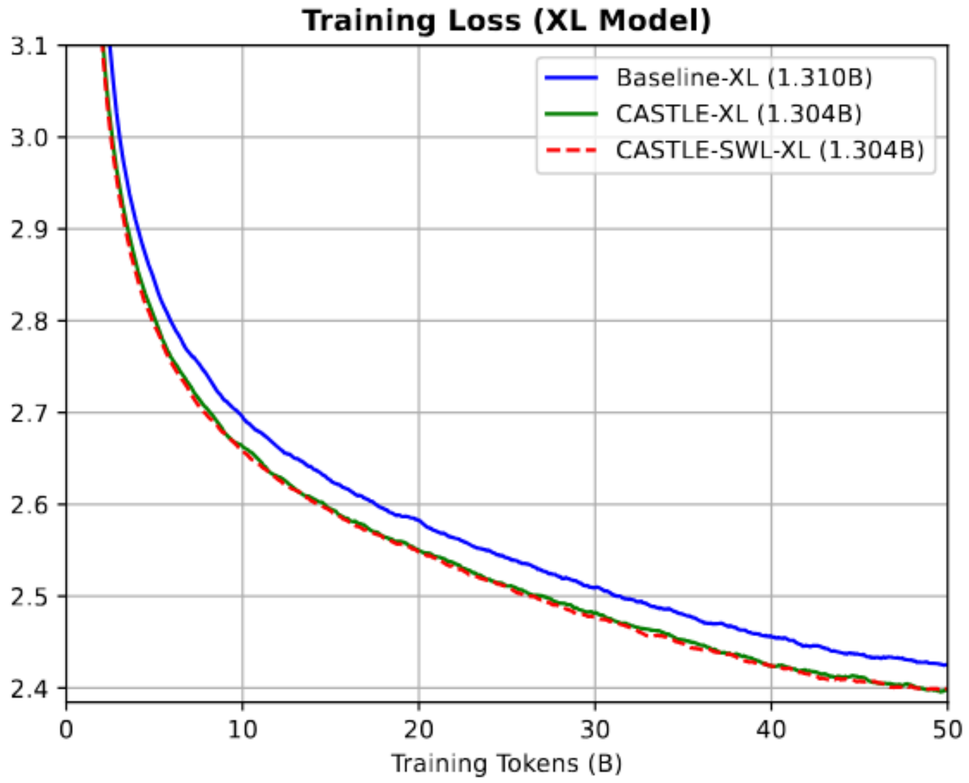
- Comparison between loss curves of small models and XL models

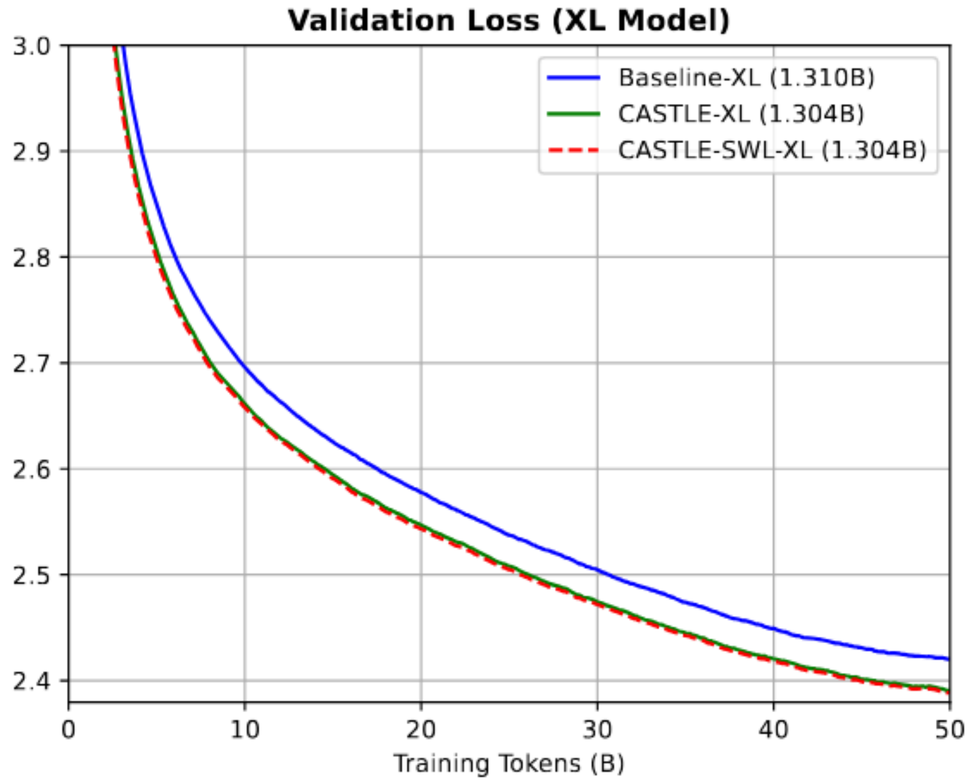




Experimental Results: CASTLE vs. CASTLE-SWL

- CASTLE-SWL matches CASLTE on model scales (XL model as example here)





Experimental Results

- Downstream Tasks (0-shot)

Model Name	ARC-C	ARC-E	BoolQ	Hella.	MMLU	OBQA	PIQA	Wino.	Avg.
Baseline-S	**26.71**	54.76	52.51	35.78	22.89	30.40	63.98	**52.57**	42.45
CASTLE-S	26.19	**56.69**	**59.85**	**36.28**	**23.00**	31.60	64.25	52.25	**43.76**
CASTLE-SWL-S	26.62	54.12	59.60	35.97	22.87	**32.60**	**64.31**	50.51	43.33
Baseline-M	28.58	60.90	53.61	43.01	23.21	33.40	67.95	50.91	45.20
CASTLE-M	**30.20**	61.36	**58.01**	43.24	**25.34**	**34.60**	67.95	52.64	**46.67**
CASTLE-SWL-M	29.52	**61.41**	57.03	**43.76**	23.29	33.00	**68.12**	**53.83**	46.24
Baseline-L	32.59	65.07	57.49	47.45	23.57	32.60	70.51	50.75	47.50
CASTLE-L	32.34	**65.15**	57.65	47.87	**24.51**	35.60	**70.78**	53.51	48.43
CASTLE-SWL-L	**32.76**	63.51	**60.95**	**48.50**	23.72	**36.00**	70.02	**54.85**	**48.79**
Baseline-XL	33.79	66.62	61.04	51.40	**26.72**	36.20	**72.58**	54.06	50.30
CASTLE-XL	35.32	67.51	**62.81**	**52.15**	23.74	37.00	70.67	**56.59**	50.72
CASTLE-SWL-XL	**36.43**	**69.07**	60.24	51.99	24.47	**37.40**	71.27	55.09	**50.74**

Experimental Results

- Downstream tasks (5-shot)

Model Name	ARC-C	ARC-E	BoolQ	Hella.	MMLU	OBQA	PIQA	Wino.	Avg.
Baseline-S	25.68	54.97	56.09	33.81	**25.54**	28.20	63.98	**52.57**	42.60
CASTLE-S	26.02	54.25	**57.13**	35.24	25.22	29.80	64.53	50.99	42.90
CASTLE-SWL-S	**27.39**	**56.19**	56.09	**35.46**	25.34	**30.20**	**64.85**	51.22	**43.34**
Baseline-M	31.06	62.46	48.47	42.83	25.22	33.00	68.39	51.78	45.40
CASTLE-M	32.17	**64.06**	**54.62**	43.47	25.22	**33.80**	**69.48**	52.49	46.91
CASTLE-SWL-M	**32.85**	63.85	54.19	**44.18**	**26.43**	**33.80**	69.04	**53.43**	**47.22**
Baseline-L	33.36	63.64	59.24	46.16	**26.82**	33.40	69.53	**54.06**	48.28
CASTLE-L	**37.37**	**67.89**	50.95	47.71	26.11	34.20	**70.18**	**54.06**	48.56
CASTLE-SWL-L	36.26	65.53	**60.58**	**48.55**	24.70	**34.80**	69.10	53.67	**49.15**
Baseline-XL	35.58	65.78	61.07	50.84	**26.71**	36.20	71.27	52.72	50.02
CASTLE-XL	**39.08**	**70.24**	**62.60**	51.63	24.16	**37.40**	71.00	**58.33**	51.80
CASTLE-SWL-XL	38.99	70.08	61.74	**52.35**	25.85	37.20	**72.52**	56.75	**51.93**

Ablation Studies on Number of Keys

- Is the improvement from more keys?
  - Model configurations

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	$n_{\text{LookaheadKeys}} + n_{\text{CausalKeys}}$	d
Baseline-M	353M	24	1024 (=16 * 64)	16	16	64
CASTLE-M	351M	24	1024	9	18	64
CASTLE-M-16	340M	24	1024	8	16	64
Baseline-XL	1.310B	24	2048 (=16 * 128)	16	16	128
CASTLE-XL	1.304B	24	2048	9	18	128
CASTLE-XL-16	1.260B	24	2048	8	16	128

  - Training & Validation loss

		Train		Eval	
	n_{params}	Loss	PPL	Loss	PPL
Baseline-M	353M	2.740	15.483	2.742	15.523
CASTLE-M	351M	**2.709**	**15.018**	**2.711**	**15.039**
CASTLE-M-16	340M	2.714	15.093	2.716	15.126
Baseline-XL	1.310B	2.548	12.779	2.543	12.723
CASTLE-XL	1.304B	**2.507**	**12.267**	**2.503**	**12.219**
CASTLE-XL-16	1.260B	2.514	12.349	2.511	12.316

		Train		Eval	
	n_{params}	Loss	PPL	Loss	PPL
Baseline-M	353M	2.740	15.483	2.742	15.523
CASTLE-SWL-M	351M	**2.710**	**15.036**	**2.713**	**15.068**
CASTLE-SWL-M-16	340M	2.716	15.117	2.718	15.150
Baseline-XL	1.310B	2.548	12.779	2.543	12.723
CASTLE-SWL-XL	1.304B	**2.506**	**12.255**	**2.503**	**12.217**
CASTLE-SWL-XL-16	1.260B	2.513	12.339	2.508	12.276

Thank you!