Table of Contents

A. GPU Hardware and Kernel Execution for 2D Linear Propagation

Modern GPUs, such as NVIDIA's A100, enable high parallelism through a hierarchical execution model involving grids, thread blocks, and warps. A kernel—a compiled function for GPU execution—is launched as a grid of thread blocks, where each block contains up to 1024 threads organized into 32-thread warps, the basic scheduling unit on streaming multiprocessors (SMs; 108 on A100). Warps execute in a single-instruction, multiple-thread (SIMT) manner, maximizing throughput when occupancy—the proportion of active warps per SM—is high, balanced against constraints like register usage (up to 65,536 per SM) and shared memory (up to 164 KB per SM).

In sequence modeling architectures like 2D linear propagation [23, 34], input tensors of shape $B\times C\times H\times W$ (batch size B, channels C, height H, width W) are processed via a line-scan propagation scheme. This involves sequential row or column updates with parallel computations within each step. The CUDA implementation maps spatial dimensions ($H \times W$) to threads, while B and C define independent slices for concurrent processing. In the kernel, a 1D block configuration might allocate blockDim.x to a fixed number of threads (e.g., 512), with the grid size scaled by $B\times C\times H$ (or $B\times C\times W$) to distribute the workload across SMs. Each thread handles a pixel along the parallel spatial axis, launching a separate kernel per propagation step (e.g., per row or column), which results in thousands of micro-launches. This design, however, faces scalability challenges with large $B\times C$. GPUs have finite concurrency limits, constrained by the number of SMs and per-SM block capacity (32 blocks). When $B\times C$ exceeds these limits, excess slices are processed sequentially, causing runtime spikes despite the theoretical parallelism.

B. Implementation Details

B.1. Pretraining

Before initiating end-to-end distillation, we conduct a lightweight pretraining stage designed to stabilize optimization and provide a strong initialization. Specifically, we train on 5M image–text pairs sampled from the DataComp benchmark [14], which balances diversity and scale. We adopt the AdamW optimizer [26] with a learning rate of 4×10^{-5}, a global batch size of 1024, and 300 warmup steps. The schedule follows linear decay, gradually annealing the learning rate to zero. This setup encourages early convergence without overfitting, and the pretrained weights serve as a robust initialization for subsequent supervised distillation. Our empirical analysis shows that omitting this step leads to unstable training in the early epochs and consistently lower downstream performance.

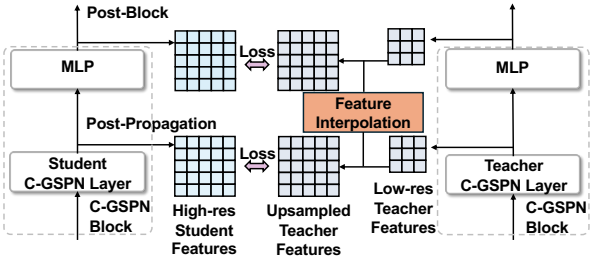

Figure 6. High-resolution encoder distillation: a frozen low-resolution teacher supervises a higher-resolution student via upsampled features at two taps (post-propagation and post-block), with feature interpolation bridging resolutions and applied progressively in a resolution curriculum. See Sec. 4.3 in the main paper for details.

B.2. End-to-End Distillation Training

For full-scale training, we distill C-GSPN on 600M curated image–text pairs from DataComp. The student model is optimized to align with its teacher (OpenCLIP SO/14) through staged supervision, as outlined in Section 4.2. We adopt a sparse distillation strategy, where we only distill every ninth block of the teacher model. We again use AdamW with a higher learning rate of 4×10^{-4}, a global batch size of 8192, and a cosine decay learning-rate schedule with 10 000 warmup steps. This configuration provides both the stability required for large-batch training and the flexibility to adapt across the different supervision stages.

B.3. Loss Composition and Balancing

The total distillation loss combines the two supervision taps per block—*post-propagation (PP)* and *post-block (PB)*:

$$\mathcal{L} = \alpha\,\mathcal{L}_{\text{PP}} + \beta\,\mathcal{L}_{\text{PB}}, \tag{14}$$

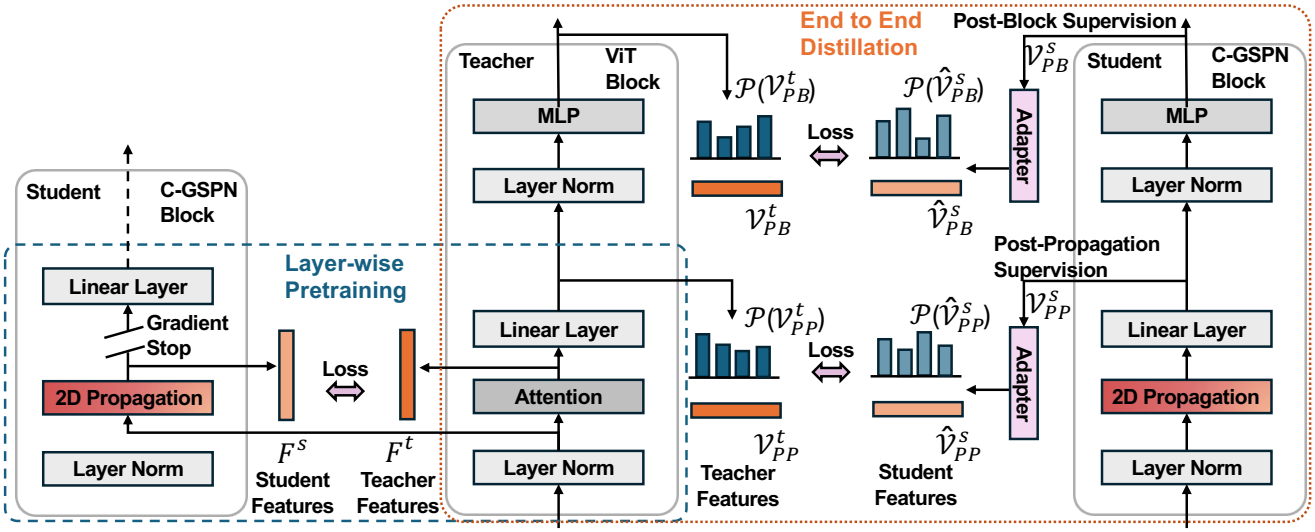

Figure 7. **Two-stage distillation for scaling C-GSPN**. Stage 1: Sublayer-wise pretraining aligns each C-GSPN propagation sublayer to the teacher's attention sublayer. Stage 2: End-to-end distillation applies dual taps—post-propagation (PP) and post-block (PB)—with lightweight feature adaptors to reduce feature-space mismatch.

with

$$\mathcal{L}_{\text{PP}} = \text{MSE}\big(V^s_{\text{PP}}, V^t_{\text{PP}}\big) + \lambda_1 \,\text{KL}\big(P(V^s_{\text{PP}}) \,\|\, P(V^t_{\text{PP}})\big), \quad (15)$$
$$\mathcal{L}_{\text{PB}} = \text{MSE}\big(V^s_{\text{PB}}, V^t_{\text{PB}}\big) + \lambda_2 \,\text{KL}\big(P(V^s_{\text{PB}}) \,\|\, P(V^t_{\text{PB}})\big).$$

Here, $V^{s/t}_{\text{PP}}$ and $V^{s/t}_{\text{PB}}$ denote student/teacher features at the PP and PB taps, and $P(\cdot)$ is the token-wise softmax distribution. We set $\alpha = \beta = 0.5$ to balance PP and PB supervision, ensuring that the propagation sublayer is directly constrained without being overshadowed by block-level matching. The divergence weights $\lambda_1 = \lambda_2 = 7/3$ provide a balance between feature-level alignment (MSE) and distributional matching (KL). To reduce feature-space mismatch, a lightweight 2-layer MLP adaptor is inserted before each tap (Sec. 4.2).

Dataset	378-teacher	378-multires	448-multires	518-multires
ADE20K	46.0	45.8	45.8	45.9

Table 4. Multi-resolution distillation on ADE20K (mIoU). A single student trained to support multiple input resolutions matches the single-resolution baseline.

B.4. Stability Practices

Layer-wise pretraining (Stage 1) provides consistent signals to each sublayer before end-to-end optimization (Stage 2). In ablations, removing either the adaptors or Stage 1 degrades stability and final accuracy.

B.5. Replacement of 1/9 GSPN Blocks by Attention Blocks

For the hybrid configuration studied in Sec. 5.4, we start from the 27-block GSPN backbone and *evenly interleave* attention blocks through depth. Concretely, we replace every ninth GSPN block with a standard multi-head self-attention block that uses the same embedding dimension and MLP as the surrounding GSPN blocks, yielding a $3/27 \approx 1/9$ attention ratio. This keeps the overall depth and parameterization comparable while injecting sparse pairwise mixing at regular intervals, and is the variant reported as the "1/9-attention hybrid" in our overhead and cost–quality trade-off analysis.

C. More Experimental Results

We evaluate multi-resolution distillation by training a single C-GSPN model that operates across multiple input resolutions without special positional embeddings. A low-resolution teacher supervises a multi-resolution student during distillation. As shown in Table 4, the student maintains comparable performance across 378, 448, and 518 resolutions, indicating that our approach transfers effectively across scales.

C.1. More Latency and Throughput Analysis

To complement the system-efficiency results in Sec. 5.1 of the main paper, we provide a detailed breakdown of how C-GSPN compares to attention, FlashAttention, and the original GSPN across more resolutions. Table 5 reports end-to-end throughput (images/second) at batch size 32 for three input resolutions, together with multiplicative gaps relative

	Model Throughput (img/s)					
	518		1554		2590	
Method	**img/s**	×	**img/s**	×	**img/s**	×
Attention	49.42	2.28×	OOM	OOM	OOM	OOM
FlashAttention	104.06	1.08×	4.78	2.58×	0.78	5.32×
GSPN	25.53	4.42×	OOM	OOM	OOM	OOM
C-GSPN (ours)	**112.91**	**1×**	**12.35**	**1×**	**4.15**	**1×**

Table 5. **Model throughput comparison.** Throughput (img/s) at batch size 32 across three resolutions. Shaded columns report multiplicative gap vs ours (×; higher is worse). C-GSPN maintains practical throughput at all resolutions while competing methods either run out of memory (OOM) or suffer severe throughput degradation.

	Sublayer						Layer						Block					
	518		1554		2590		518		1554		2590		518		1554		2590	
Method	**ms**	×	**ms**	×	**ms**	×	**ms**	×	**ms**	×	**ms**	×	**ms**	×	**ms**	×	**ms**	×
image-resolution	518		1554		2590		518		1554		2590		518		1554		2590	
Attention	14.905	169.38×	OOM	OOM	OOM	OOM	17.919	4.06×	OOM	OOM	OOM	OOM	22.900	2.44×	OOM	OOM	OOM	OOM
FlashAttention	2.344	26.64×	164.696	550.82×	1259.135	1949.28×	5.160	1.17×	190.124	5.07×	1336.021	11.70×	10.141	1.08×	235.372	2.85×	1466.011	6.00×
GSPN	3.415	38.81×	OOM	OOM	OOM	OOM	30.391	6.89×	OOM	OOM	OOM	OOM	35.372	3.77×	OOM	OOM	OOM	OOM
C-GSPN (ours)	**0.088**	**1×**	**0.299**	**1×**	**0.646**	**1×**	**4.413**	**1×**	**37.472**	**1×**	**114.170**	**1×**	**9.394**	**1×**	**82.720**	**1×**	**244.160**	**1×**

Table 6. **Latency breakdown across architectural levels.** Sublayer: core 2D propagation unit vs. attention sublayer latency (ms). Layer: full layer including all components. Block: complete transformer block latency (ms). Measurements at batch size 32. Shaded columns report multiplicative gap vs ours (×; higher is worse). C-GSPN's propagation sublayer achieves up to 1949× speedup over FlashAttention at resolution 2590 and 6× speedup on end-to-end block latency, enabling practical high-resolution inference.

VQA tasks	**Seed-Img**	**VizWiz**	**MMMU**
OpenCLIP+Qwen2.5-3B	72.9	55.5	24.4
C-GSPN+Qwen2.5-3B	72.1	55.8	24.2

to C-GSPN. Table 6 further decomposes latency into sublayer, layer, and full-block timings, revealing that the propagation sublayer remains stable while attention-based baselines quickly become memory- or latency-bound at high resolutions. These results highlight that C-GSPN delivers competitive or superior throughput even when attention baselines are able to run, and maintains practical latency up to 2.6K resolution where many attention configurations are out-of-memory.

References

[1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. 6

[2] Aarti Basant, Abhijit Khairnar, Abhijit Paithankar, Abhinav Khattar, Adi Renduchintala, Adithya Renduchintala, Aditya Malte, Akhiad Bercovich, Akshay Hazare, Alejandra Rico, et al. Nvidia nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model. *arXiv preprint arXiv:2508.14444*, 2025. 8

[3] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020. 2

[4] Aviv Bick, Kevin Li, Eric Xing, J Zico Kolter, and Albert Gu. Transformers to ssms: Distilling quadratic knowledge to subquadratic models. *Advances in Neural Information Processing Systems*, 37:31788–31812, 2024. 3, 5

[5] Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17302–17313, 2023. 8

[6] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023. 6

[7] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023. 6

[8] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers, 2019. 1

[9] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 1, 2

[10] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, 2019. Association for Computational Linguistics. 1

[11] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022. 2

[12] Xin Dong, Yonggan Fu, Shizhe Diao, Wonmin Byeon, ZIJIA CHEN, Ameya Sunil Mahabaleshwarkar, Shih-Yang Liu, Matthijs Van keirsbilck, Min-Hung Chen, Yoshi Suhara, Yingyan Celine Lin, Jan Kautz, and Pavlo Molchanov. Hymba: A hybrid-head architecture for small language models. In *The Thirteenth International Conference on Learning Representations*, 2025. 8

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 7

[14] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023. A1

[15] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 1, 2

[16] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. 2

[17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, 2021. 1

[18] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020. 1, 2

[19] Yingyue Li, Bencheng Liao, Wenyu Liu, and Xinggang Wang. Matvlm: Hybrid mamba-transformer for efficient vision-language modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3, 5

[20] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26763–26773, 2024. 6

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[22] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024. 5

[23] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. *Advances in Neural Information Processing Systems*, 30, 2017. 2, 3, A1

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[25] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4122–4134, 2025. 5

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. A1

[27] Kevis-Kokitsi Maninis, Kaifeng Chen, Soham Ghosh, Arjun Karpur, Koert Chen, Ye Xia, Bingyi Cao, Daniel Salz, Guangxing Han, Jan Dlabal, Dan Gnanapragasam, Mojtaba Seyedhosseini, Howard Zhou, and André Araujo. TIPS: Text-Image Pretraining with Spatial Awareness. In *ICLR*, 2025. 8

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, 2015. 7

[30] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021. 2, 5

[31] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 2

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[33] Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, et al. An empirical study of mamba-based language models. *arXiv preprint arXiv:2406.07887*, 2024. 8

[34] Hongjun Wang, Wonmin Byeon, Jiarui Xu, Jinwei Gu, Ka Chun Cheung, Xiaolong Wang, Kai Han, Jan Kautz, and Sifei Liu. Parallel sequence modeling via generalized spatial propagation network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2, 3, 4, 8, A1

[35] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 2

[36] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, pages 14138–14148, 2021. 2

[37] Zhendong Yang, Zhe Li, Ailing Zeng, Zexian Li, Chun Yuan, and Yu Li. Vitkd: Practical guidelines for vit feature knowledge distillation. *arXiv preprint arXiv:2209.02432*, 2022. 2, 5

[38] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017. 2

[39] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020. 2

[40] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 1, 4

[41] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 7