TABLE OF CONTENTS

A ADDITIONAL RELATED WORK

A.1 SUBQUADRATIC ATTENTION AND ALTERNATIVES

Subquadratic alternatives to full softmax attention aim to reduce the $\mathcal{O}(N^2)$ dependence on token count while preserving global interactions. *Sparsity- and window-based* designs such as Longformer and BigBird in NLP and Swin for vision constrain attention to local windows with a few global tokens, yielding near-linear scaling but making long-range mixing sensitive to the chosen sparsity pattern and hyperparameters (Beltagy et al., 2020; Zaheer et al., 2020; Liu et al., 2021). *Kernelized/low-rank* approaches linearize attention—e.g., Linear Transformers, Performer, Nyströmformer, Linformer—trading exactness for approximation; their accuracy often depends on the feature map, rank, or landmark scheme and may require careful tuning (Katharopoulos et al., 2020; Choromanski et al., 2020; Xiong et al., 2021; Wang et al., 2020). IO-aware exact attention like FlashAttention reduces constant factors via optimized memory access, yet latency still scales quadratically with tokens at high resolution (Dao et al., 2022). Beyond attention, *state-space models* (S4; Mamba) offer linear-time sequence operators, but adapting 1D formulations to high-resolution vision typically requires extra 2D inductive bias or hierarchical designs (Gu et al., 2021; Gu & Dao, 2023). In contrast, *spatial propagation networks* operate natively on 2D grids and remove positional embeddings; recent GSPN extends this idea to four-direction propagation with linear complexity (Liu et al., 2017; Wang et al., 2025). Our work scales GSPN to foundation-model pretraining through a compact, CUDA-optimized instantiation distilled from ViT teachers, achieving substantially lower latency at 1K–2K while maintaining competitive transfer.

A.2 FOUNDATION MODEL DISTILLATION

Knowledge distillation (KD) has long been employed to compress large models into more efficient students. Early works explored attention transfer in CNNs (Zagoruyko & Komodakis, 2017), while DeiT (Touvron et al., 2021) demonstrated that ViT-to-ViT distillation can achieve strong performance at scale, highlighting the potential of KD for transformers. Subsequent studies refined these ideas: ViTKD guidelines (Yang et al., 2022) emphasized the importance of intermediate supervision and careful layer alignment for stable training. More recently, KD research has extended beyond isomorphic student–teacher pairs to span across operator families. For instance, quadratic-to-subquadratic transfer has been explored to compress attention-heavy architectures into efficient

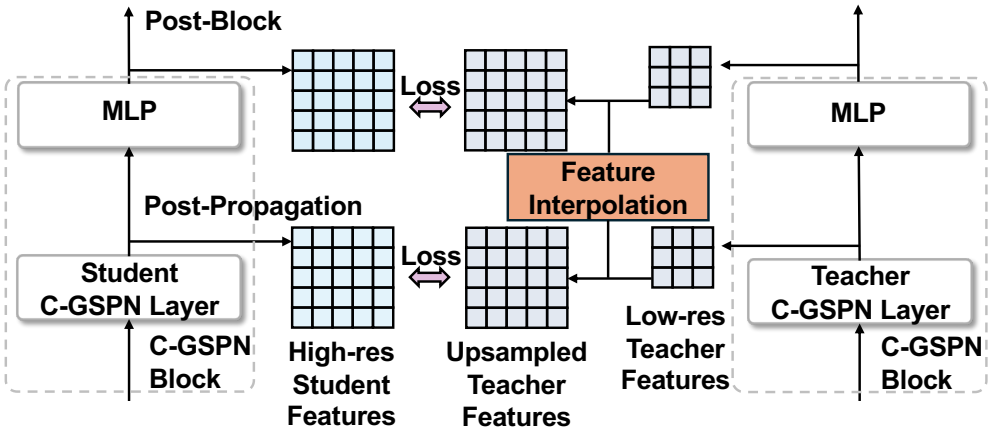

Figure 7: High-resolution encoder distillation: a frozen low-resolution teacher supervises a higher-resolution student via upsampled features at two taps (post-propagation and post-block), with feature interpolation bridging resolutions and applied progressively in a resolution curriculum. See Sec. 4.3 in the main paper for details.

approximations (Bick et al., 2024), and hybrid schemes distilling Mamba into Transformer backbones have emerged (Li et al., 2025). These works suggest that KD can serve as a bridge across heterogeneous operator classes, though most prior efforts remain at modest scales or focus on small task-specific settings. In contrast, our work investigates *foundation-scale distillation across operator families*, targeting the challenging scenario of attention-to-propagation transfer. We demonstrate that staged supervision (layer-wise and block-level) combined with CUDA-optimized latent propagation enables both competitive accuracy and substantial efficiency gains, scaling effectively to high-resolution inputs (1K–2K) while preserving transfer performance.

B 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 (Wang et al., 2025; Liu et al., 2017), 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.

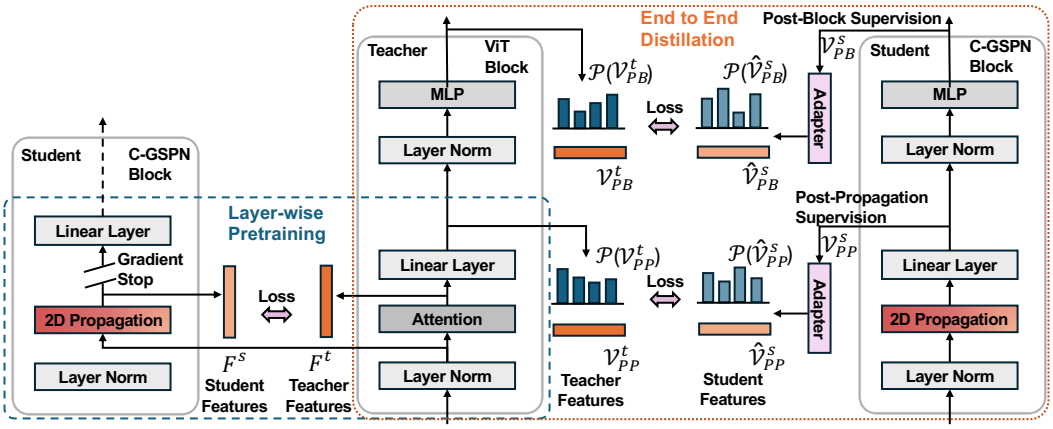

Figure 8: **Two-stage distillation for scaling C-GSPN**. Stage 1: Sublayer-wise pretraining aligns each C-GSPN propagation sublayer to the teacher's attention sublayer. Stage 2: End-to-end distillation applies dual taps—post-propagation (PP) and post-block (PB)—with lightweight feature adaptors to reduce feature-space mismatch.

C Implementation Details

C.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 (Gadre et al., 2023), which balances diversity and scale. We adopt the AdamW optimizer (Loshchilov & Hutter, 2017) 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.

C.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.

C.3 Loss Composition and Balancing

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

$$\mathcal{L} = \alpha\, \mathcal{L}_{\text{PP}} + \beta\, \mathcal{L}_{\text{PB}}, \tag{15}$$

with

$$\begin{aligned} \mathcal{L}_{\text{PP}} &= \text{MSE}\big(V^s_{\text{PP}}, V^t_{\text{PP}}\big) + \lambda_1\, \text{KL}\big(P(V^s_{\text{PP}}) \,\|\, P(V^t_{\text{PP}})\big), \\ \mathcal{L}_{\text{PB}} &= \text{MSE}\big(V^s_{\text{PB}}, V^t_{\text{PB}}\big) + \lambda_2\, \text{KL}\big(P(V^s_{\text{PB}}) \,\|\, P(V^t_{\text{PB}})\big). \end{aligned} \tag{16}$$

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.

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

Table 3: Multi-resolution distillation on ADE20K (mIoU). A single student trained to support multiple input resolutions matches the single-resolution baseline.

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).

C.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.

D 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 3, the student maintains comparable performance across 378, 448, and 518 resolutions, indicating that our approach transfers effectively across scales.

E The Use of Large Language Models (LLMs)

We used large language models (OpenAI GPT) only for wording suggestions. All outputs were reviewed and edited by the authors; no analyses, results, or code central to our contributions were generated by LLMs, and no sensitive data were provided.