GeoSemba: Reconstructing State Space Model for Cross Paradigm Representation in Medical Image Segmentation

Supplementary Material

6. Analysis of Stability and Complexity

This section analyzes two theoretical properties of the proposed Cross-Paradigm Module (CPM) that are not detailed in the main paper: stability under the SSR-induced adaptive parameterization and preservation of linear-time complexity after integrating SSR and CAR. For completeness, we briefly recall that CPM employs the SSR-parameterized discretized transition $\bar{A}_i^{\mathrm{SSR}} = \exp(\Delta_i^{\mathrm{SSR}} A)$ and input projection $\bar{B}_i^{\mathrm{SSR}} = \Delta_i^{\mathrm{SSR}} B_i^{\mathrm{SSR}}$, while CAR refines only the external input pathway. We next analyze the resulting recurrent dynamics and computational complexity.

6.1. Stability Preservation under Adaptive SSR

We analyze the state transition induced by SSR under the adaptive parameterization.

Spectral contractivity of the discretized transition. Assume that the shared continuous-time state matrix A is Hurwitz, namely

$$\Re(\lambda_j(A)) < 0, \qquad \forall j, \tag{19}$$

and that the adaptive discretization factor satisfies

$$0 < \Delta_{\min} \le \Delta_i^{\mathrm{SSR}} \le \Delta_{\max} < +\infty. \tag{20}$$

Then, for each token index i,

$$\bar{A}_i^{\mathrm{SSR}} = \exp(\Delta_i^{\mathrm{SSR}} A). \tag{21}$$

By the spectral mapping theorem,

$$\begin{aligned}
\rho(\bar{A}_i^{\mathrm{SSR}}) &= \rho\big(\exp(\Delta_i^{\mathrm{SSR}} A)\big) && (22)\\
&= \max_j \left|\exp\big(\Delta_i^{\mathrm{SSR}} \lambda_j(A)\big)\right| && (23)\\
&= \max_j \exp\big(\Delta_i^{\mathrm{SSR}} \Re(\lambda_j(A))\big) && (24)\\
&\le \max_j \exp(\Delta_{\min} \Re(\lambda_j(A))) =: \eta < 1. && (25)
\end{aligned}$$

Hence, the SSR-induced discretization preserves a uniform spectral contraction of the canonical Mamba transition.

Bounded adaptive modulation. The dynamic parameters generated by SSR depend on the prototype representations produced by SPG. Since the prototype centers are updated by exponential moving average,

$$p_k^{(t)} = \lambda_{ema} p_k^{(t-1)} + (1-\lambda_{ema})\hat{p}_k^{(t)}, \qquad 0 < \lambda_{ema} < 1, \tag{26}$$

their evolution is a convex combination of historical and current estimates, yielding

$$\|p_k^{(t)}\|_2 \le \max\Big(\|p_k^{(t-1)}\|_2, \|\hat{p}_k^{(t)}\|_2\Big). \tag{27}$$

Therefore, the prototype sequence remains bounded provided the batch estimates are bounded. If the parameter heads that map prototype-conditioned features to Δ_i^{SSR}, B_i^{SSR}, and C_i^{SSR} are Lipschitz and operate on bounded inputs, then these token-wise modulation terms also remain bounded during training. Thus, SSR perturbs the selective state-space update in a bounded manner.

CAR affects only the input pathway. CAR enters CPM only through the external input term and introduces no additional recurrent transition. Consequently, CAR changes the excitation of the state-space system but does not modify the contractive property of $\bar{A}_i^{\mathrm{SSR}}$. The recurrent dynamics are therefore governed by the SSR-modulated transition rather than by CAR itself.

Implication. Taken together, CPM preserves well-posed recurrent dynamics for bounded inputs: the shared matrix A remains Hurwitz, SSR perturbs the discretization and projection terms through bounded token-dependent modulation, and CAR acts only on the external input pathway without altering the recurrent transition.

6.2. Linear-Time Complexity of SSR and CAR

We analyze the computational complexity of SSR and CAR with respect to spatial resolution. Following the notation in the main paper, let $N = HW$ denote the number of flattened spatial tokens. For a given CPM block, the channel width C is treated as a layer-specific constant independent of N. Likewise, the architectural hyperparameters—including the number of prototypes M, the number of channel groups G, the pooled key size $S \times S$, the reduced support size $R \times R$, and the retention ratio K—are treated as constants with respect to spatial resolution.

Complexity of SSR. SSR consists of three main stages. First, semantic prototype assignment matches each of the N tokens against M prototypes in a C-dimensional feature space, resulting in complexity $\mathcal{O}(NMC)$. The subsequent prototype accumulation over all tokens requires a single pass through the feature map and costs $\mathcal{O}(NC)$. Second, prototype-level interaction is performed only among the M prototype nodes rather than all N tokens, yielding complexity $\mathcal{O}(M^2C)$. Third, the generation of token-wise adaptive scan parameters, including Δ_i^{SSR}, B_i^{SSR}, and C_i^{SSR},

Table 7. Asymptotic complexity comparison of representative token mixers.

Method	Complexity
Global Self-Attention	$\mathcal{O}(N^2)$
Local / Window Attention (window size w tokens)	$\mathcal{O}(Nw)$
Multi-Directional Mamba (k scans)	$\mathcal{O}(kN)$
CPM (SSR + CAR)	$\mathcal{O}(\mathbf{N})$

is applied independently at each spatial location and therefore incurs linear cost with respect to N (e.g., $\mathcal{O}(NC)$ for lightweight projection heads). Therefore, the overall complexity of SSR is

$$\mathcal{O}(NMC) + \mathcal{O}(NC) + \mathcal{O}(M^2C) + \mathcal{O}(NC), \quad (28)$$

which is linear in N when M and C are fixed with respect to spatial resolution.

Complexity of CAR. Given an input feature map $x \in \mathbb{R}^{H\times W\times C}$ with $N = H \times W$, CAR follows a coarse-to-fine design. In the macro-perception stage, the depthwise convolutions, normalization, and activation are applied locally on the $H \times W$ grid, resulting in complexity $\mathcal{O}(NC)$. In the micro-focus stage, channel attention and the linear projections used to generate queries and key features are either pointwise or performed on a fixed pooled support and therefore also remain linear with respect to N. Specifically, query features are generated for all N spatial locations, while key features are compressed by adaptive pooling into a fixed set of S^2 regional representatives. The resulting group-wise affinity computation matches N queries against S^2 pooled keys, yielding complexity $\mathcal{O}(NS^2)$ up to fixed channel-group constants. The optional learnable transformation induced by W^d further remaps the affinities from the pooled support of size S^2 to a fixed local support of size R^2, which incurs complexity $\mathcal{O}(NS^2R^2)$. The subsequent Top-K masking and sparse aggregation also remain linear in N, since they operate on fixed-size regional supports under a fixed retention ratio. Consequently, the overall complexity of CAR can be bounded by

$$\mathcal{O}(NC) + \mathcal{O}(NS^2) + \mathcal{O}(NS^2R^2) + \mathcal{O}(NR^2), \quad (29)$$

which reduces to linear complexity with respect to N when C, S, R, and K are fixed with respect to spatial resolution.

6.3. Comparison with Attention and Multi-Directional Scans

Table 7 compares the asymptotic complexity of representative token-mixing paradigms. Global self-attention incurs quadratic complexity in the token length, while local/window attention remains linear only under a fixed window size. Multi-directional Mamba also preserves linear scaling, but its cost grows proportionally with the number of scan paths. In contrast, CPM maintains linear complexity with respect to N while enabling adaptive geometric-semantic and spatial-channel modeling through fixed prototype and regional support sizes.

Overall, compared with quadratic token mixing and constant-factor scan expansion, CPM preserves the linear-time advantage of Mamba with respect to spatial token length.

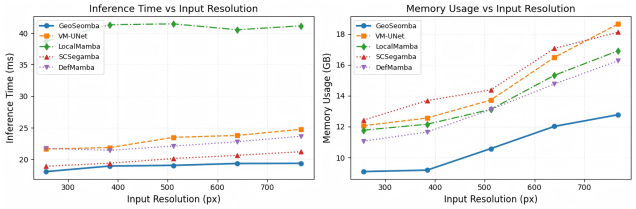

Figure 9. Computational scalability analysis of GeoSemba and representative Mamba variants under different input resolutions.

7. Computational Scalability Analysis

To assess the runtime scalability of GeoSemba across varying input resolutions, we measure inference latency and GPU memory consumption against representative Mamba-based baselines, including VM-UNet [8], LocalMamba [3], SCSegamba [6], and DefMamba [7]. The results are reported in Fig. 9.

GeoSemba maintains a favorable efficiency profile across the evaluated resolution range. Its GPU memory consumption increases moderately from approximately 10.8 GB at 512×512 to 12.8 GB at 768×768, while the inference latency remains around 18 ms per image at both resolutions. This suggests that GeoSemba sustains stable runtime efficiency as the spatial resolution increases. In contrast, competing Mamba-based variants, particularly SCSegamba [6] and LocalMamba [3], exhibit consistently higher memory usage and longer inference latency, indicating less favorable scaling at higher resolutions.

This empirical trend is consistent with the complexity analysis in Sec. 6. Although GeoSemba enhances the state-space update with SSR and CAR, it preserves the linear-time efficiency of single-scan Mamba-style modeling through compact prototype- and region-level interactions rather than dense token-wise computation. Consequently, GeoSemba improves representational capacity without incurring steeper runtime and memory growth associated with more elaborate scan or interaction schemes.

8. Objective Function and Loss Analysis

The training objective is defined as a weighted combination of Dice loss [5] and binary cross-entropy (BCE) loss:

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{Dice}} + \beta\mathcal{L}_{\text{BCE}}, \quad (30)$$

Table 8. Ablation study of BCE/Dice loss weighting ratios on representative medical segmentation benchmarks.

Loss Ratio (α:β)	ISIC2018[2]		COVID19-1[4]		BUSI[1]	
	DSC (%)	IoU (%)	DSC (%)	IoU (%)	DSC (%)	IoU (%)
Dice	88.9	81.6	81.2	74.0	80.7	72.1
BCE	89.3	81.9	81.5	74.2	81.0	72.4
1:1	**91.1**	**84.1**	**83.8**	**77.0**	**82.1**	**73.7**
1:2	90.6	83.2	83.0	76.3	81.8	73.3
2:1	90.4	83.1	82.9	75.8	81.5	73.0

where α and β denote the relative weights of the two terms. The BCE loss is defined as:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right], \tag{31}$$

and the Dice loss is given by

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{i=1}^{N} y_i \hat{y}_i + \epsilon}{\sum_{i=1}^{N} y_i^2 + \sum_{i=1}^{N} \hat{y}_i^2 + \epsilon}, \tag{32}$$

where ϵ is a small constant for numerical stability. BCE and Dice serve complementary roles during optimization: BCE provides dense pixel-level supervision, while Dice emphasizes region-level overlap and is less sensitive to foreground-background imbalance due to its normalized formulation. Unless otherwise stated, we set $\alpha = \beta = 1$ in all experiments.

To examine the effect of loss weighting, we conduct an ablation study over different Dice/BCE ratios, as reported in Table 8. Using either loss alone consistently degrades performance. Specifically, BCE-only training is more affected by class imbalance, whereas Dice-only training provides weaker pixel-level guidance for fine-grained localization. The balanced setting of $1:1$ achieves the best overall results across the evaluated datasets, including 91.1% DSC on ISIC2018 [2], confirming that the two objectives are most effective when jointly optimized.

Moreover, moderate ratio changes, such as $1:2$ or $2:1$, cause only marginal performance drops. Such degradation arises because deviating from the balanced $1:1$ setting weakens the coordination between dense pixel-level supervision and region-level overlap optimization, resulting in a less balanced training objective. Nevertheless, the drop remains limited because both loss terms are still jointly optimized under moderate reweighting, so their complementary roles are largely preserved.

9. Gradient Flow under Top-K Sparsification

In CAR, the row-wise operator $\mathcal{T}_K(\cdot)$ in Eq. (16) performs ratio-controlled sparsification on the affinity matrix Φ_g. Specifically, for each row of Φ_g, it retains the highest-scoring K proportion of affinity responses and suppresses the remaining ones to zero, producing the sparse affinity matrix $\bar{\Phi}_g$.

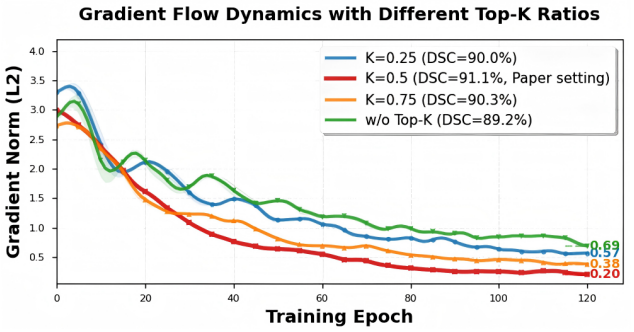

Figure 10. Non-zero gradients under different Top-K settings

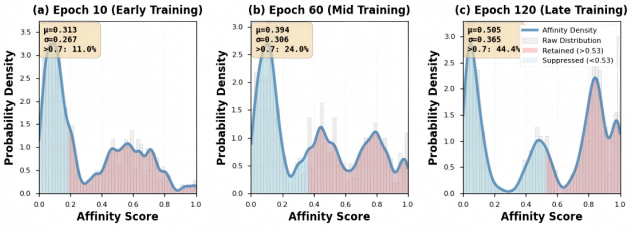

Figure 11. Visualization of affinity patterns at different training stages under Top-K sparsification.

For gradient analysis, conditioned on a fixed Top-K support in the current backward pass (i.e., away from support-switching boundaries and tie cases), the masking operation can be written equivalently in element-wise form as:

$$\bar{\Phi}_g = \Phi_g \odot \mathcal{I}_g, \tag{33}$$

where $\mathcal{I}_g$ is a binary indicator matrix constructed from the Top-K support of each row in Φ_g, with entries equal to 1 at the selected positions and 0 elsewhere.

Under the standard fixed-support view within a backward pass, the masking operation is piecewise differentiable, and gradients are propagated only through the retained entries:

$$\frac{\partial \mathcal{L}}{\partial \Phi_g} = \mathcal{I}_g \odot \frac{\partial \mathcal{L}}{\partial \bar{\Phi}_g}. \tag{34}$$

Based on this formulation, we further analyze the resulting gradient behavior. Specifically, we track the non-zero gradient norms under different Top-K settings and visualize the corresponding affinity patterns across training stages. As shown in Fig. 10 and Fig. 11, gradients remain concentrated on the selected entries throughout training, while the learned affinity maps progressively evolve from diffuse responses to more structured and discriminative patterns. These observations suggest that the Top-K masking strategy in CAR enables sparse affinity selection without compromising optimization on the retained neighbors.

References

[1] Walid Al-Dhabyani, Mohammed Gomaa, Hussien Khaled, and Aly Fahmy. Dataset of breast ultrasound images. *Data in brief*, 28:104863, 2020.

[2] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 168–172. IEEE, 2018.

[3] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. In *European Conference on Computer Vision*, pages 12–22. Springer, 2024.

[4] Ma Jun, Ge Cheng, Wang Yixin, An Xingle, Gao Jiantao, Yu Ziqi, Zhang Minqing, Liu Xin, Deng Xueyuan, Cao Shucheng, et al. Covid-19 ct lung and infection segmentation dataset. *(No Title)*, 2020.

[5] Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. Dice loss for data-imbalanced nlp tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 465–476, 2020.

[6] Hui Liu, Chen Jia, Fan Shi, Xu Cheng, and Shengyong Chen. Scsegamba: lightweight structure-aware vision mamba for crack segmentation in structures. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29406–29416, 2025.

[7] Leiye Liu, Miao Zhang, Jihao Yin, Tingwei Liu, Wei Ji, Yongri Piao, and Huchuan Lu. Defmamba: Deformable visual state space model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8838–8847, 2025.

[8] Jiacheng Ruan, Jincheng Li, and Suncheng Xiang. Vm-unet: Vision mamba unet for medical image segmentation. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2024.