

Soft Routing-Inspired Specialization for Efficient Satellite Image Time Series Segmentation

Paulo Victor de Magalhães Rozatto, Saulo Moraes Villela, Luiz Maurílio da Silva Maciel, and Marcelo Bernardes Vieira(✉)

Department of Computer Science, Federal University of Juiz de Fora, Juiz de Fora, Brazil
paulo.rozatto@estudante.ufjf.br,
{saulo.moraes,luiz.maciel,marcelo.bernardes}@ufjf.br

Abstract. Satellite image time series segmentation requires models that capture complex temporal patterns while remaining computationally efficient for large-scale remote sensing applications. Motivated by principles from Mixture-of-Experts (MoE) models, we introduce routing-inspired specialization into the Exchanger architecture to increase representational flexibility without substantially increasing inference cost. Rather than using sparse expert routing, the proposed approach leverages soft routing mechanisms, incorporating expert-specific transformations within the temporal processing pipeline to utilize model capacity more efficiently while maintaining linear computational complexity.

We evaluate the approach on the PASTIS and MTLCC benchmarks and compare it to a consistently trained Exchanger baseline. While absolute improvements in mean Intersection-over-Union (mIoU) are modest and within the expected cross-validation variability, certain configurations achieve baseline-level accuracy while reducing FLOPs by up to 30%, lowering latency and increasing throughput. These results provide evidence that MoE-inspired specialization can improve the efficiency and scalability of dense temporal models for satellite image segmentation.

Keywords: Satellite Image Time Series · Semantic Segmentation · Mixture of Experts · Remote Sensing

1 Introduction

Satellite Image Time Series (SITS) are sequences of satellite observations of the same geographic area acquired over time. They provide rich information for monitoring agricultural activity, vegetation cycles, and land-cover change at large scales [13]. Effectively exploiting SITS requires modeling long-term temporal contexts to capture seasonal patterns, crop phenology, and gradual land-cover dynamics.

Vision Transformers (ViTs) have recently emerged as a powerful paradigm for visual representation learning and have been successfully adapted to SITS analysis due to their ability to model long-range dependencies in time and space [16].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16768, pp. 38–51, 2027.
https://doi.org/10.1007/978-3-032-30488-9_3

However, their quadratic self-attention cost makes direct application to long temporal sequences computationally expensive.

Exchanger [1] addresses this limitation by replacing full self-attention with a linear-complexity token exchange mechanism based on a Collect–Update–Distribute pipeline. While this design enables efficient global information flow, the grouped representations are processed using shared transformations across all groups, leaving limited room for group-specific specialization.

In this work, we apply Mixture-of-Experts (MoE) [4] principles to the Exchanger framework to increase representational flexibility while maintaining linear computational complexity. Inspired by Soft MoE [9], we reinterpret Exchanger's collect stage as a form of soft routing, analogous to the dispatch operation, and incorporate expert-specialized transformations within the update stage. We further adapt the distribute stage to more closely resemble the combine operation in Soft MoE.

Our experiments suggest that interpreting Exchanger through the lens of soft routing specialization can lead to a more efficient utilization of model capacity on the PASTIS benchmark [13], where Exchanger already serves as a strong baseline. While the observed improvements in mean Intersection over Union (mIoU) are modest and fall within typical cross-validation variability, the proposed modifications provide a more favorable trade-off between predictive performance and computational cost. In particular, the revised combine operation allows certain configurations to achieve baseline-level accuracy while reducing FLOPs (float point operations) and latency, and improving throughput.

Contributions. The main contributions of this work are as follows:

- We reinterpret Exchanger's Collect–Update–Distribute pipeline using the perspective of soft expert routing, highlighting structural and functional similarities between token exchange and mixture-of-experts computation.
- We introduce a soft routing-inspired specialization mechanism within the Exchanger temporal encoder, where latent groups act as soft slots processed by expert-specific modules, enabling group-wise specialization while maintaining linear complexity.
- We propose a more efficient implementation of the distribute (combine) stage by reusing attention logits from the collect stage, reducing computational and memory costs without noticeably affecting predictive performance.
- We empirically show that the proposed modifications maintain comparable segmentation accuracy to the Exchanger baseline on PASTIS [13] and MTLCC [12], while providing improved efficiency and more favorable scaling behavior with sequence length.

2 Related Work

Transformer architectures have become widely used for SITS segmentation. U-TAE [13] introduced temporal attention on top of a convolutional backbone, while TSViT [16] demonstrated the effectiveness of fully transformer-based spatiotemporal modeling.

To improve scalability for long temporal sequences, Exchanger proposed a non-hierarchical Vision Transformer that replaces quadratic self-attention with a linear-complexity token exchange mechanism based on a three-stage pipeline and, to the best of our knowledge, represents the state of the art on the PASTIS dataset [13]. VistaFormer [8] later explored scalable transformer architectures for large-scale SITS segmentation through factorized spatiotemporal modeling. Our work builds directly on Exchanger and investigates how introducing routing-based specialization affects the balance between computational cost and segmentation accuracy.

Mixture-of-Experts (MoE) models increase model capacity through expert specialization, combined with routing mechanisms that assign inputs to different subnetworks [15]. MoE layers were first widely adopted to scale Transformer models in natural language processing, where sparse routing enables large parameter counts with limited per-token computation [3,5]. Similar sparse MoE strategies have recently been explored in computer vision to scale Vision Transformers while controlling computational costs [10].

More recently, soft routing has emerged as a competitive alternative in computer vision, replacing hard expert selection with continuous routing weights that enable specialization without conditional execution [9]. In Soft MoE, routing is performed by first computing token–slot affinities and then deriving normalized dispatch and combination weights that distribute information between tokens and expert-specific latent representations. We adopt this viewpoint to reinterpret the collect stage of Exchanger as an affinity-based routing mechanism and introduce expert-specialized transformations within its update stage, increasing model capacity while preserving the linear-complexity inference behavior of the original architecture.

3 Proposed Method

The Exchanger has a factorized design, which means it separates temporal and spatial modeling for satellite image time series. Given an input sequence $X \in \mathbb{R}^{T\times C\times H\times W}$, the temporal encoder processes each spatial location as an independent temporal sequence. After this processing, the features are reshaped back to the spatial grid and fed to a spatial encoder for segmentation.

In this work, we modify only the temporal encoder, keeping both the spatial encoders unchanged. We also retain Exchanger's original strategy for collapsing spatial dimensions prior to temporal modeling: each position (h, w) is treated independently across time, reshaping the input from $T\times C\times H\times W$ into $(HW)\times T\times C$. This ensures that temporal attention operates at each spatial location without altering the spatial modeling stage.

Figure 1 provides an overview of the full architecture and highlights the routing-based specialization introduced within the temporal encoder.

Building on this formulation, the remainder of this section details the proposed modifications to the Collect–Update–Distribute pipeline. We first describe the collect operation as an attention-based soft routing mechanism, then intro-

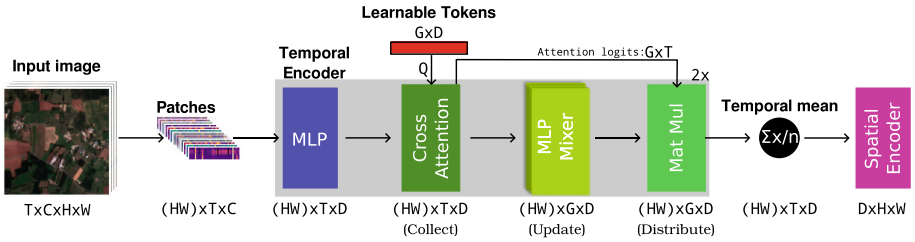


Fig. 1. Overview of the proposed architecture. The input is a satellite image time series of shape $T \times C \times H \times W$, where T denotes the number of time steps, C the number of channels, and $H \times W$ the spatial resolution. Spatial locations are processed independently over time by reshaping the input to $(HW) \times T \times C$ and projecting features to a D-dimensional embedding. Cross-attention with learnable tokens softly routes temporal features into G temporal groups. Grouped features are then updated by E MLP mixer experts. Finally, the Distribute step softly combines grouped features back into the temporal dimension. Temporal mean aggregation is applied before reshaping features to $D \times H \times W$ and passing them to a spatial encoder for pixel-wise segmentation.

duce expert-specialized transformations in the update stage, and finally present the modified distribute stage that recombines group representations using routing-derived weights.

3.1 Collect: Exchanger Group Attention as Soft Routing

Let $X \in \mathbb{R}^{T \times D}$ denote a temporal sequence of feature embeddings, where T is the number of time steps and D is the feature dimension after an MLP projection. We denote $X_t \in \mathbb{R}^D$ the embedding at time step $t \in \{1, \ldots, T\}$. The collect stage aggregates these temporal tokens into a fixed set of G latent group tokens using the group attention mechanism introduced in Exchanger. Each group is associated with a learnable query vector $q_g \in \mathbb{R}^D$, where $g \in \{1, \ldots, G\}$. Temporal tokens provide keys $k_t \in \mathbb{R}^D$ and values $v_t \in \mathbb{R}^D$, obtained through linear projections of X_t.

Formally, the attention score between group g and time step t is computed as:

$$A_{g,t} = \frac{q_g \cdot k_t}{\sqrt{d}} + B_{g,t}, \tag{1}$$

where $B_{g,t}$ denotes the positional bias defined in [1]. The attention weights are obtained via softmax over the temporal dimension:

$$a_{g,t} = \mathrm{softmax}_t(A_{g,t}). \tag{2}$$

The resulting group token representations are given by:

$$Z_g = \sum_{t=1}^{T} a_{g,t} X_t. \tag{3}$$

From a routing perspective, the attention weights $a_{g,t}$ can be interpreted as a learned assignment matrix that distributes each temporal token across a fixed set of latent group representations. This matches the general routing formulation of [6], where routing consists of computing affinity scores followed by normalized assignment weights that determine how inputs contribute to intermediate processing units. Here, the softmax normalization over t ensures that each group token aggregates information from all temporal positions with data-dependent weights, making the collect stage a continuous, differentiable routing mechanism over the temporal dimension.

3.2 Update: Expert Specialization Within the Exchanger Block

In the original Exchanger block, the update stage processes all group tokens using the same set of shared transformations. This stage enables information exchange among groups, playing the role of global interaction in place of full self-attention.

We retain this overall structure but introduce specialization within the feature transformation applied to each group. Specifically, the update stage still consists of two steps: (1) a shared transformation that mixes information across groups and (2) a feature transformation applied within each group. In the original Exchanger, both steps use shared MLP-Mixer components. In our model, the across-group transformation remains shared, while the within-group feature transformation is replaced with expert-specific modules.

First, a shared across-group mixing module, denoted as $\mathrm{Mix}_{\mathrm{group}}(\cdot)$, is applied to all group tokens:

$$Z'_g = Z_g + \mathrm{Mix}_{\mathrm{group}}(\mathrm{Norm}(Z_g)), \tag{4}$$

which enables information exchange across latent groups and preserves the global interaction mechanism of the Exchanger.

Subsequently, the within-group feature transformation is performed using expert-specific modules. Let E denote the number of experts, which may differ from the number of group tokens G. Each group token Z'_g is associated with an expert through a predefined mapping $e(g) \in \{1, \ldots, E\}$ and is updated as:

$$\tilde{Z}_g = Z'_g + f_{e(g)}(\mathrm{Norm}(Z'_g)), \tag{5}$$

where $f_{e(g)}(\cdot)$ denotes the expert-specific feature transformation for the expert $e(g)$. When $E < G$, multiple groups share the same expert, enabling controlled parameter sharing and allowing us to adjust the degree of specialization without increasing the number of groups.

By introducing specialization only in the within-group transformation while keeping the across-group mixing shared, the update stage preserves the global information exchange mechanism of the Exchanger. At the same time, expert-specific transformations allow different latent groups to develop specialized representations for distinct temporal patterns.

3.3 Distribute: Combine Operation

The distribute stage maps the updated group representations back to temporal tokens by reusing the attention logits from the collect stage with a different normalization. Under the routing interpretation, this corresponds to the combine operation in Soft Mixture-of-Experts, where outputs are recombined using the same affinity scores that define dispatch.

This perspective enables a more efficient formulation of the distribute stage. In the original Exchanger, distribution is implemented as a separate cross-attention operation, where temporal tokens act as queries over group representations, requiring additional projection matrices and a full attention computation. In contrast, our method leverages the shared routing structure and directly reuses the logits $A_{g,t}$ computed during collect. The combine weights are obtained by simply changing the normalization dimension, eliminating the need for new query–key interactions or additional attention passes. As a result, routing, expert processing, and recombination are all derived from a single set of attention scores, reducing both computational cost and memory overhead.

Let $A_{g,t}$ denote the attention scores defined in Eq. (1). During the collect stage, routing (dispatch) weights are obtained via a softmax over the temporal dimension,

whereas in the distribute stage, the same scores are normalized over the group dimension:

$$c_{g,t} = \mathrm{softmax}_g(A_{g,t}), \tag{6}$$

producing combined weights that determine how each group representation contributes to a temporal token.

The redistributed temporal features are then computed as follows:

$$\tilde{X}_t = \sum_{g=1}^{G} c_{g,t}\, \tilde{Z}_g, \tag{7}$$

where $\tilde{Z}_g$ denotes the updated group tokens from the Update stage as defined in Eq. (5).

This asymmetric normalization enables a two-way soft routing mechanism: temporal tokens are first aggregated into group representations, and group representations are then softly recombined back into the temporal domain. Unlike classical sparse MoE layers, this process involves no conditional execution; all groups contribute to every temporal token through continuous combine weights.

Following redistribution, the resulting features are integrated into the block output in the same manner as in the Exchanger. The redistributed features are concatenated with the original temporal features and linearly projected:

$$Y_t = \mathrm{Proj}\big([X_t \,\|\, \tilde{X}_t]\big), \tag{8}$$

where $[\cdot\|\cdot]$ denotes channel-wise concatenation, and $\mathrm{Proj}(\cdot)$ is a learned linear projection.

Finally, a feed-forward network (FFN) with a residual connection is applied:

$$X_t^{\text{out}} = Y_t + \text{FFN}(\text{Norm}(Y_t)). \tag{9}$$

This sequence of operations substitutes the standard residual self-attention and feed-forward layers of a Transformer block, following the design principles introduced in GPViT [17] and adopted in Exchanger.

3.4 Spatial Encoder

After the Collect–Update–Distribute pipeline, the resulting token sequence is averaged over the temporal dimension and reshaped to recover the original spatial grid of size $C \times H \times W$. This design choice, retained from Exchanger, allows the use of any established semantic segmentation architecture for spatial modeling and pixel-level prediction.

4 Experiments

4.1 Datasets

We evaluate our method primarily on the Panoptic Agricultural Satellite Time Series (PASTIS) dataset [13], which consists of Sentinel-2 image time series acquired over French rural areas. The dataset contains 2,433 multi-spectral Satellite Image Time Series (SITS), each covering a spatial extent of 1.28 km×1.28 km at a spatial resolution of 10 m, resulting in image patches of size 128×128 pixels and spanning from September 2018 to November 2019.

PASTIS includes 20 semantic classes: 18 crop types, a background class, and a void class. Pixels labeled as void correspond either to parcels belonging to non-annotated crop types or to parcels whose majority lies outside the patch boundaries. Following the dataset protocol, the void class is ignored during evaluation. The dataset is provided with an official 5-fold cross-validation scheme, summarized in Table 1.

Table 1. Official 5-fold cross-validation splits used for PASTIS. Each row defines one split, indicating which folds are assigned to the training, validation, and test sets.

Split	Train	Validation	Test
I	1, 2, 3	4	5
II	2, 3, 4	5	1
III	3, 4, 5	1	2
IV	4, 5, 1	2	3
V	5, 1, 2	3	4

In addition to PASTIS, we report results on the Multitemporal Land Cover Classification (MTLCC) dataset [12] to assess the applicability of the proposed

method across different SITS benchmarks. MTLCC comprises 17 crop classes and consists of Sentinel-2 image time series extracted over agricultural regions in Germany, provided as small spatial tiles of size 24×24 pixels.

4.2 Training Protocol

Our training protocol is largely similar to that of Exchanger, with a small number of modifications described below. We first pre-train the temporal encoder using the pixel-set representation introduced in PASTIS-R [14], which isolates parcel-level temporal signals while discarding background pixels. This pre-training stage focuses on temporal modeling and runs for 50 epochs with a batch size of 128. We then attach a spatial encoder and fine-tune the full model end-to-end on the standard semantic segmentation split of the PASTIS dataset for 20 epochs with a batch size of 8. While Exchanger evaluates two spatial decoders, namely U-Net [11] and Mask2Former [2], we use a U-Net architecture in all experiments based on preliminary trials with our implementation.

When initializing the full model from the temporal pre-training checkpoint, we do not load all weights. Specifically, parameters associated with group token embeddings and update-stage MLP-Mixers are reinitialized, while the remaining weights are transferred from pre-training. We believe that this partial weight loading avoids forcing pretrained representations onto architectural components that differ between the temporal-only and full spatiotemporal models and provides additional regularization during fine-tuning.

During training, we divide each 128×128 image patch into a regular grid of 32×32 spatial crops and treat each crop as an independent sample. This strategy differs from the standard Exchanger training protocol, and it increases both the number of training samples and the spatial coverage of the training data. We further apply temporal dropping as data augmentation, randomly removing time steps from each temporal sequence with a drop rate of 0.3. During validation and testing, we perform inference on full-resolution 128×128 images without cropping and use the complete temporal sequences without temporal dropping.

We train all models using a standard pixel-wise cross-entropy loss for semantic segmentation, ignoring pixels belonging to the void class during loss computation. Optimization is performed with AdamW [7] using a learning rate of 2×10^{-4}, weight decay of 5×10^{-3}, and $(\beta_1, \beta_2) = (0.9, 0.999)$. We apply a polynomial learning rate schedule with linear warm-up. The warm-up phase spans 10% of the total training iterations with a warm-up factor of 10^{-3}. After warm-up, the learning rate follows a polynomial decay with power 0.9, combined with step-wise decay at 70% and 90% of the training schedule.

Unless otherwise specified, we use the same optimization settings in all experiments and follow the remaining hyperparameters of Exchanger.

4.3 Ablation Study

The ablation study examines two main architectural factors that control specialization and model capacity: the number of groups and the number of expert

mixers. Grouping determines how temporal tokens are partitioned and routed, affecting both representation learning and computational cost, as attention relies on learnable group-specific queries. Varying the number of mixers is analogous to adjusting expert capacity in MoE-inspired designs. For a fair comparison, we also train the original Exchanger baseline using an identical training protocol.

The experiments are conducted exclusively on the first split of the PASTIS cross-validation protocol, using folds 1–3 for training, fold 4 for validation, and fold 5 held out. Ablation results reported in Table 2 refer to metrics on the validation set only, and no test data are used for model selection or architectural comparison.

Table 2. Ablation study on PASTIS using the first cross-validation split. Results report validation mIoU and overall accuracy (OA) for different numbers of groups and expert mixers.

Effect	Model	Groups	Mixers	mIoU	OA
Groups	Ours	2	1	68.86%	85.47%
	Ours	4	1	69.26%	85.45%
	Ours	8	1	69.25%	85.69%
	Ours	16	1	69.38%	85.68%
Experts	Ours	4	2	69.56%	85.62%
	Ours	4	4	69.51%	85.76%
	Ours	16	2	69.25%	85.44%
	Ours	16	4	69.47%	85.71%
	Ours	16	8	69.76%	85.66%
	Ours	16	16	69.37%	85.71%

Looking into the groups effect, increasing the number of groups while keeping a single mixer leads to small but consistent gains in mIoU, a trend that is less pronounced for overall accuracy. This suggests that finer-grained token partitioning improves class-wise segmentation quality, even if the impact on global accuracy remains limited. Based on these results, we select the two most promising group configurations (4 and 16 groups) to further investigate the effect of increasing the number of expert mixers.

When varying the number of mixers within these group settings, we again observe modest but generally consistent improvements as model capacity increases, followed by saturation at higher values. This behavior aligns with the intended role of expert specialization, where additional mixers provide greater representational flexibility but with diminishing returns. From this analysis, we retain two representative configurations for the main experiments: 16 groups with 8 mixers, which achieves the highest validation mIoU and represents a higher-capacity setting, and 4 groups with 2 mixers, which offers a favorable balance between performance and theoretical computational efficiency.

4.4 Main Results

Final performance on PASTIS is reported following the official 5-fold cross-validation protocol. For each fold, we retrain the best-performing configuration identified in the ablation study using the corresponding training split and evaluate it on the designated test split. Reported results are averaged across all five folds.

For MTLCC, no additional ablation is conducted. All architectural choices and hyperparameters are fixed to those selected on PASTIS. The temporal encoder is initialized with the same Pixel-Set pretraining weights obtained from the first PASTIS split and used during ablation; the full model is then fine-tuned on the MTLCC training set and evaluated on its test set.

Table 3 compares our models with the original Exchanger architecture. Because our training protocol differs from that used in [1], we report both the results from the literature and the performance of the original Exchanger architecture trained under our setup. This distinction helps separate improvements due to optimization and training strategies from those introduced by routing-based specialization. Our models are identified by the pair (G, E), where G denotes the number of groups and E the number of expert mixing modules.

Table 3. Main results on PASTIS and MTLCC in terms of mIoU, model parameters, and GFLOPs, computed using a dummy input of size $30 \times 10 \times 128 \times 128$.

Model	mIoU		Params	GFLOPs
	PASTIS	MTLCC		
Exchanger+Unet [1]	66.8%	90.7%	8.08 M	300
Exchanger+Mask2Former [1]	67.9%	90.5%	24.59 M	329
Exchanger+Unet (our training)	68.67%	91.07%	8.08 M	300
Ours (16, 8)+Unet	**68.81%**	**91.62%**	9.93 M	276
Ours (4, 2)+Unet	68.78%	91.40%	8.34 M	**210**

In order to ensure a fair comparison with prior work, FLOPs are computed following [1] using a dummy input of size $T \times C \times H \times W = 30 \times 10 \times 128 \times 128$. Under this protocol, our models achieve comparable or improved segmentation performance while reducing computational cost. In particular, the $(4, 2)$ configuration matches the performance of the re-trained Exchanger+UNet baseline on PASTIS (+0.11 mIoU) and improves results on MTLCC (+0.33 mIoU), while reducing FLOPs by approximately 30%. The $(16, 8)$ configuration provides consistent, though modest, gains in accuracy across both datasets (+0.14 mIoU on PASTIS and +0.55 mIoU on MTLCC) together with a moderate reduction in FLOPs.

While FLOPs provide a hardware-agnostic estimate of computational complexity, they do not fully capture the practical efficiency of a model. To complement this analysis, we report additional runtime metrics including inference

latency and throughput (in samples/s) across varying sequence lengths T. All measurements are performed on a single NVIDIA A30 GPU (24 GB VRAM) using float32 precision. Due to memory constraints of the profiling setup, experiments are limited to sequence lengths up to $T = 40$. These results are summarized in Table 4 and illustrated in Fig. 2.

Table 4. Comparison of models across sequence lengths T. We report GFLOPs, latency (ms), and throughput (samples/s).

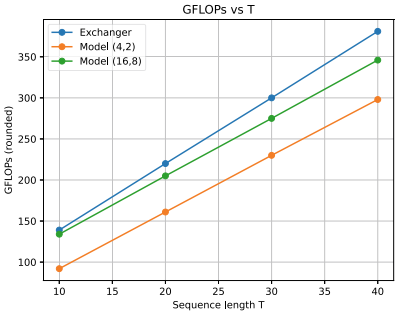
T	Exchanger			Model (4,2)			Model (16,8)		
	GFLOPs	Lat.	Thr.	GFLOPs	Lat.	Thr.	GFLOPs	Lat.	Thr.
10	139	98.81	10.12	92	63.50	15.75	134	99.86	10.01
20	220	155.29	6.44	161	95.56	10.46	205	158.35	6.32
30	300	209.17	4.78	230	129.89	7.70	275	197.05	5.07
40	381	291.41	3.43	298	162.75	6.14	346	232.60	4.30

The extended evaluation provides a more complete view of the computational behavior of the models. In particular, the $(4, 2)$ configuration achieves lower latency and higher throughput than the baseline across all tested sequence lengths, while maintaining comparable segmentation accuracy. In addition, its computational cost increases more gradually with T, as observed in both FLOPs and runtime measurements. The $(16, 8)$ configuration exhibits intermediate behavior, reflecting a trade-off between the baseline and the more efficient $(4, 2)$ variant.

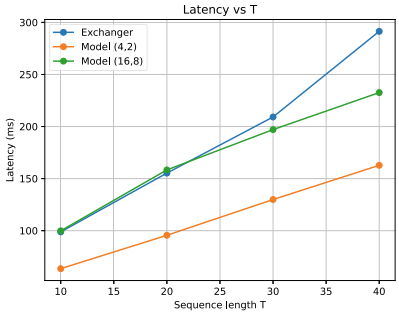
Across all models, throughput decreases as the sequence length increases, which is expected given the growing temporal dimension. However, the rate of degradation differs: the proposed models show a slower decline in throughput, indicating improved scalability with respect to T.

A qualitative example of predicted segmentation maps is shown in Fig. 3. All three models produce spatially coherent predictions that largely respect parcel boundaries, indicating that they can capture the geometric structure of agricultural fields. While the overall predictions are similar, differences can be observed in specific regions, particularly for smaller or less distinctive parcels where class confusion occurs. For instance, the Exchanger model incorrectly classifies a road segment as spring barley, whereas the other models do not exhibit this error. Additionally, areas labeled as void are ignored during both training and evaluation. As a result, apparent misclassifications in these regions do not impact the reported performance and should not be interpreted as model errors.

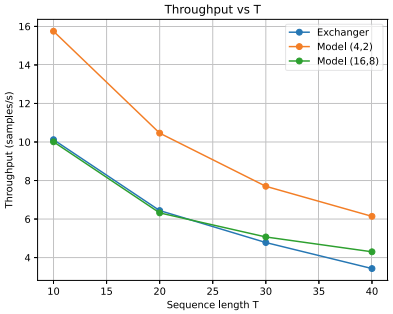
Given the inherent variability of cross-validation, we do not claim statistically significant accuracy gains. Instead, the results demonstrate that the proposed architecture enables a more favorable trade-off between accuracy and efficiency. This is consistent with the design motivation of mixture-of-experts approaches, which aim to increase representational capacity while controlling computational cost through conditional computation.



(a) GFLOPs vs Sequence length T



(b) Latency vs Sequence length T



(c) Throughput vs Sequence length T

Fig. 2. Efficiency comparison across models as a function of sequence length T. We report (a) computational cost in GFLOPs, (b) inference latency (ms), and (c) throughput (samples/s).

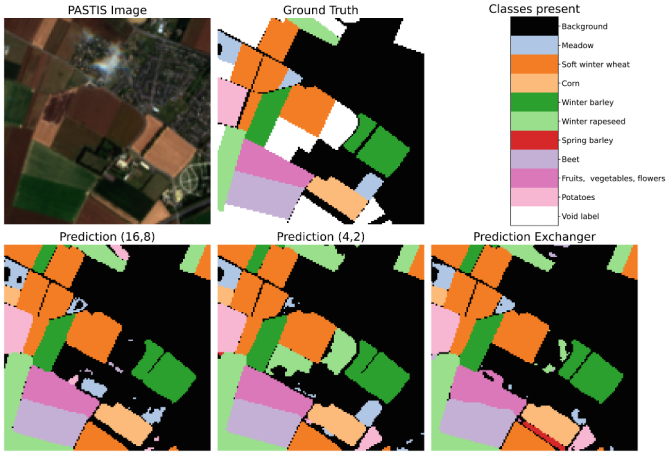


Fig. 3. Qualitative segmentation example on PASTIS. Top row (left to right): RGB composite, ground-truth labels, and the legend of classes present in the patch. Bottom row (left to right): predictions from the proposed (16,8) and (4,2) models, and the baseline Exchanger model.

5 Conclusion

In this work, we introduced routing-inspired specialization into the Exchanger architecture for satellite image time series segmentation. Motivated by principles from Mixture-of-Experts (MoE) models, the proposed design incorporates expert-specialized transformations within the temporal processing pipeline while maintaining linear computational complexity.

Evaluations on the PASTIS and MTLCC benchmarks show that the proposed modifications maintain segmentation accuracy comparable to an original Exchanger baseline while reducing computational cost. Beyond FLOPs, we analyzed practical efficiency by measuring inference latency and throughput across varying sequence lengths. Results indicate that certain configurations, such as $(4, 2)$, achieve lower latency and higher throughput than the baseline, and their efficiency degrades more gradually as sequence length increases. The $(16, 8)$ configuration exhibits intermediate behavior, providing a trade-off between baseline accuracy and efficiency.

Overall, these findings demonstrate that soft MoE-inspired specialization can improve both theoretical and practical efficiency of dense temporal models in remote sensing. Future work could explore adaptive or more explicit routing mechanisms to further enhance scalability and representation learning in large-scale satellite time series analysis.

Acknowledgments. The authors would like to thank the Minas Gerais State Research Foundation - FAPEMIG, Brazil. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. This work has been supported by UFJF's High-Speed Integrated Research Network (RePesq) https://www.repesq.ufjf.br/.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Cai, X., Bi, Y., Nicholl, P., Sterritt, R.: Revisiting the encoding of satellite image time series. In: 34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, 20–24 November 2023. BMVA (2023)
2. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation (2022). https://doi.org/10.1109/CVPR52688.2022.00135
3. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: scaling to trillion parameter models with simple and efficient sparsity (2022)
4. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Comput. **3**(1), 79–87 (1991). https://doi.org/10.1162/neco.1991.3.1.79
5. Lepikhin, D., et al.: {GS}hard: scaling giant models with conditional computation and automatic sharding. In: International Conference on Learning Representations (2021)

6. Liu, T., Blondel, M., Ruiz, C.R., Puigcerver, J.: Routers in vision mixture of experts: an empirical study. Trans. Mach. Learn. Res. (2024). https://doi.org/10.48550/arXiv.2401.15969
7. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019). https://doi.org/10.48550/arXiv.1711.05101
8. MacDonald, E., Jacoby, D., Coady, Y.: Vistaformer: scalable vision transformers for satellite image time series segmentation. arXiv preprint arXiv:2409.08461 (2024)
9. Puigcerver, J., Ruiz, C.R., Mustafa, B., Houlsby, N.: From sparse to soft mixtures of experts. In: The Twelfth International Conference on Learning Representations (2024). https://doi.org/10.48550/arXiv.2308.00951
10. Riquelme, C., et al.: Scaling vision with sparse mixture of experts. In: Proceedings of the 35th International Conference on Neural Information Processing Systems, NIPS '21. Curran Associates Inc., Red Hook (2021)
11. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. CoRR arxiv:1505.04597 (2015). https://doi.org/10.48550/arXiv.1505.04597
12. Rußwurm, M., Körner, M.: Multi-temporal land cover classification with sequential recurrent encoders. ISPRS Int. J. Geo-Inf. **7**(4) (2018). https://doi.org/10.3390/ijgi7040129
13. Sainte Fare Garnot, V., Landrieu, L.: Panoptic segmentation of satellite image time series with convolutional temporal attention networks. In: ICCV (2021). https://doi.org/10.1109/ICCV48922.2021.00483
14. Sainte Fare Garnot, V., Landrieu, L., Chehata, N.: Multi-modal temporal attention models for crop mapping from satellite time series. ISPRS J. Photogram. Remote Sens. (2022). https://doi.org/10.1016/j.isprsjprs.2022.03.012
15. Shazeer, N., et al.: Outrageously large neural networks: the sparsely-gated mixture-of-experts layer. In: International Conference on Learning Representations (2017)
16. Tarasiou, M., Chavez, E., Zafeiriou, S.: Vits for sits: vision transformers for satellite image time series. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10418–10428 (2023)
17. Yang, C., Xu, J., Mello, S.D., Crowley, E.J., Wang, X.: GPViT: a high resolution non-hierarchical vision transformer with group propagation (2023)