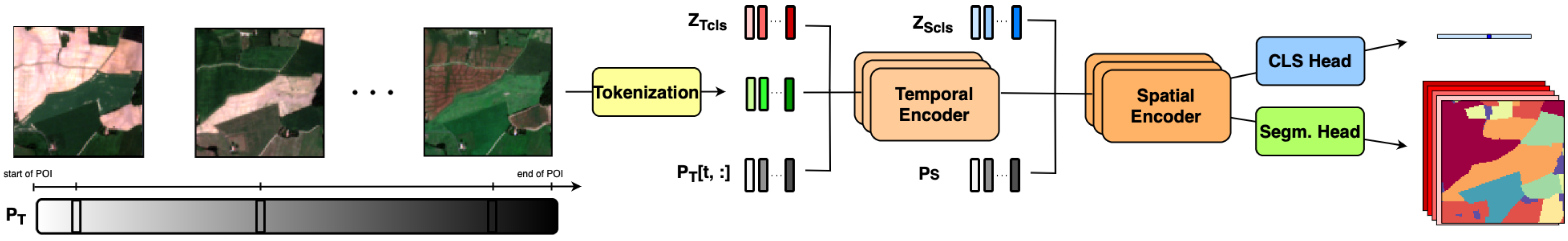
ViTs for SITS: Vision Transformers for Satellite Image Time Series

Michail Tarasiou, Erik Chavez, Stefanos Zafeiriou
Imperial College London

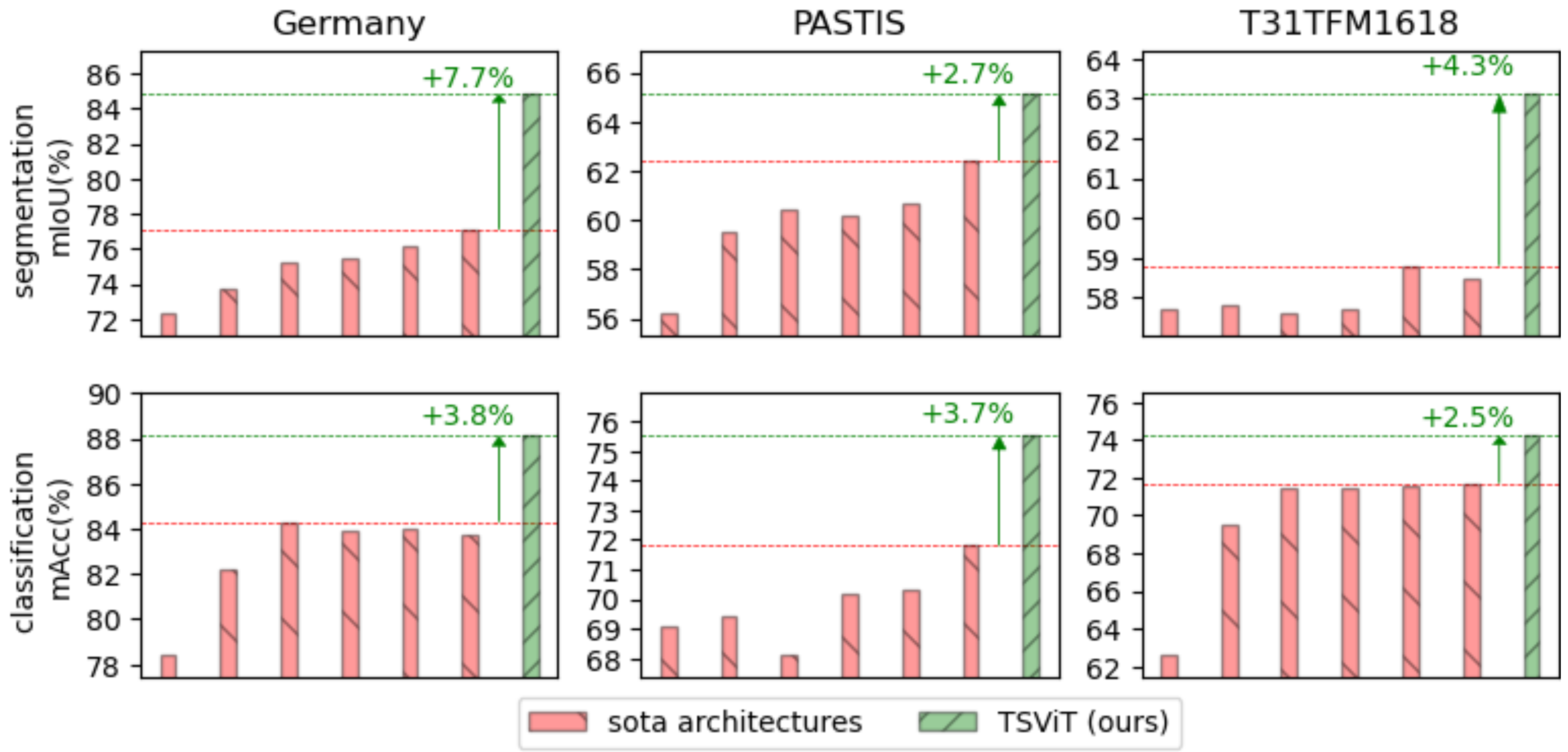
WED-AM-209

Vision Transformers for Satellite Image Time Series

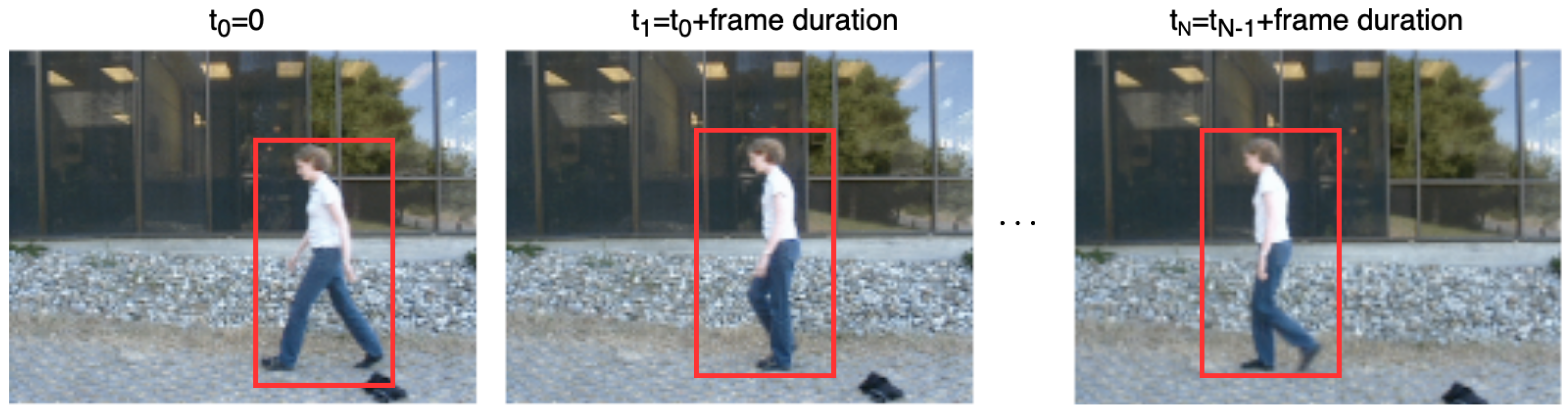


Temporo-Spatial Vision Transformer (TSViT)

- Order of factorization
- Dynamic, date-aware position encodings
- Constrained spatial modelling
- SOTA in crop type recognition



Why Temporal-Spatial factorisation?



Learning Layered Motion Segmentations of Video, Kumar et. al., ICCV2005

Spatial-Temporal factorisation makes sense for video but not for SITS

- Context can be misleading
- A single pixel is informative in SITS
- No moving objects

Dynamic Temporal Encodings

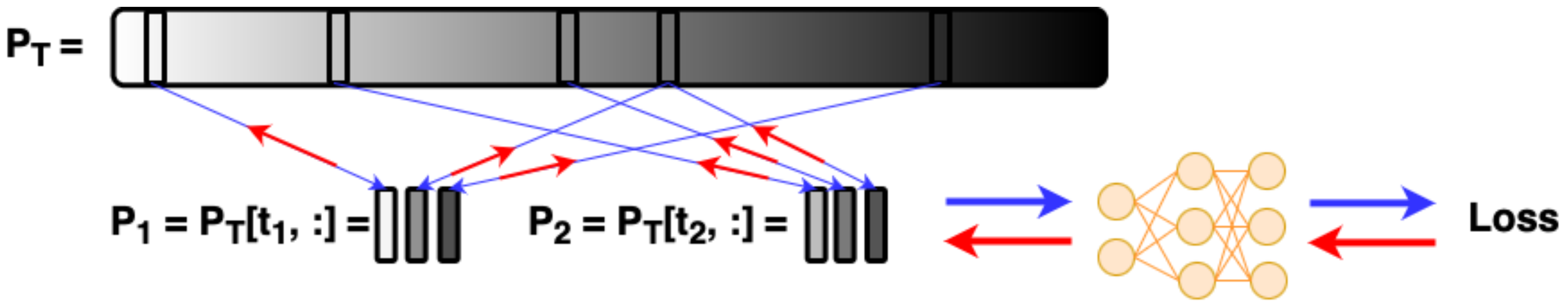
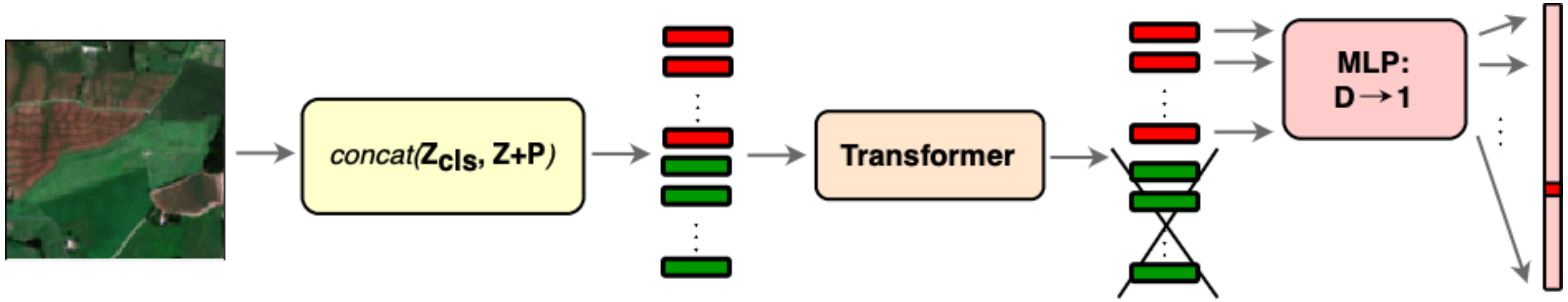


Image distribution is uneven in time

- Duration between acquisitions varies and acquisitions can be corrupted
- Absolute time matters not only order
- Keep $\mathbf{P_T}$ for all dates seen during training
- Index $\mathbf{P_T[t, :]}$ by sample times **t**
- Backpropagate to update used indices of $\mathbf{P_T}$

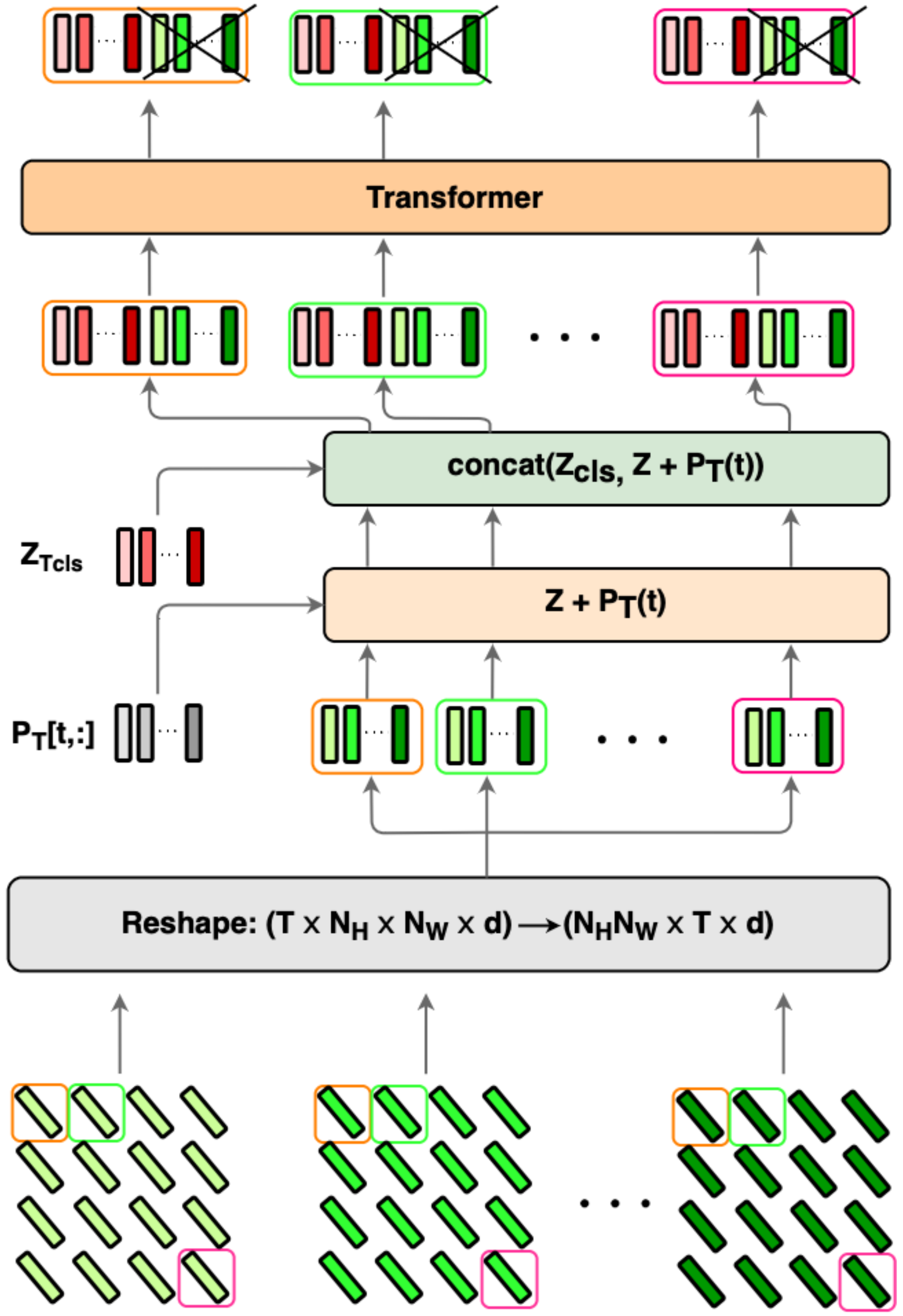
Multiple *cls*-tokens



Use learned *cls*-tokens as in BERT, Devlin et. al., 2018.

- Multiple tokens (#tokens = #classes) vs single token
- Increased capacity
- Each token predicts single class logits

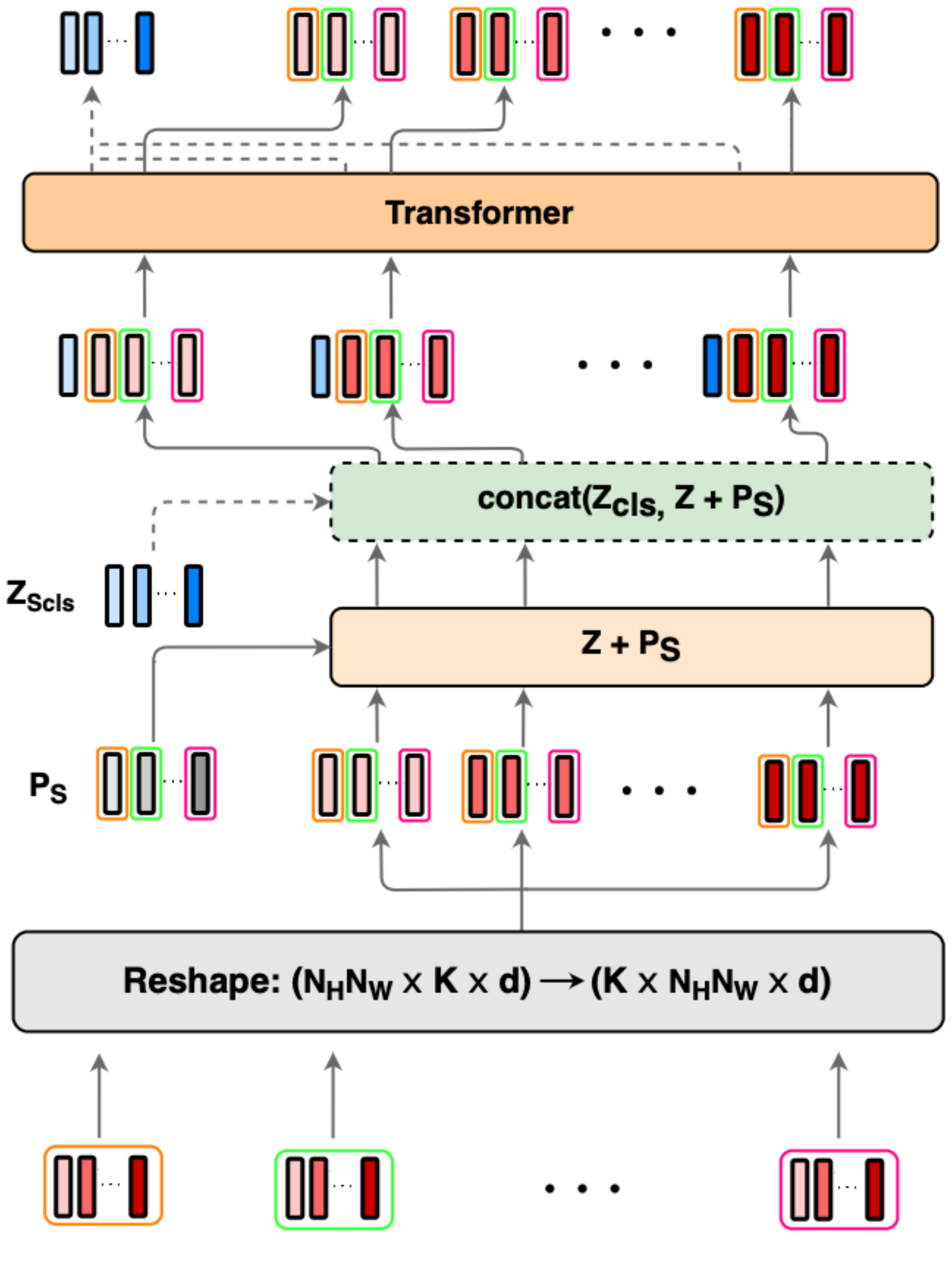
Temporal Encoder architecture


Transformer
concat(Z_{cls}, Z + P_T(t))
Z_{Tcls}
Z + P_T(t)
P_T[t,:]
Reshape: (T x N_H x N_W x d) → ($N_H N_W$ x T x d)

(a) Temporal Encoder

- Reshape tokenised input to timeseries for all $N_H N_W$ token locations
- Dynamic temporal position encodings $\mathbf{P_T}$
- Concatenate $\mathbf{Z_{Tcls}}$
- Process all locations in parallel
- Keep only first K output tokens

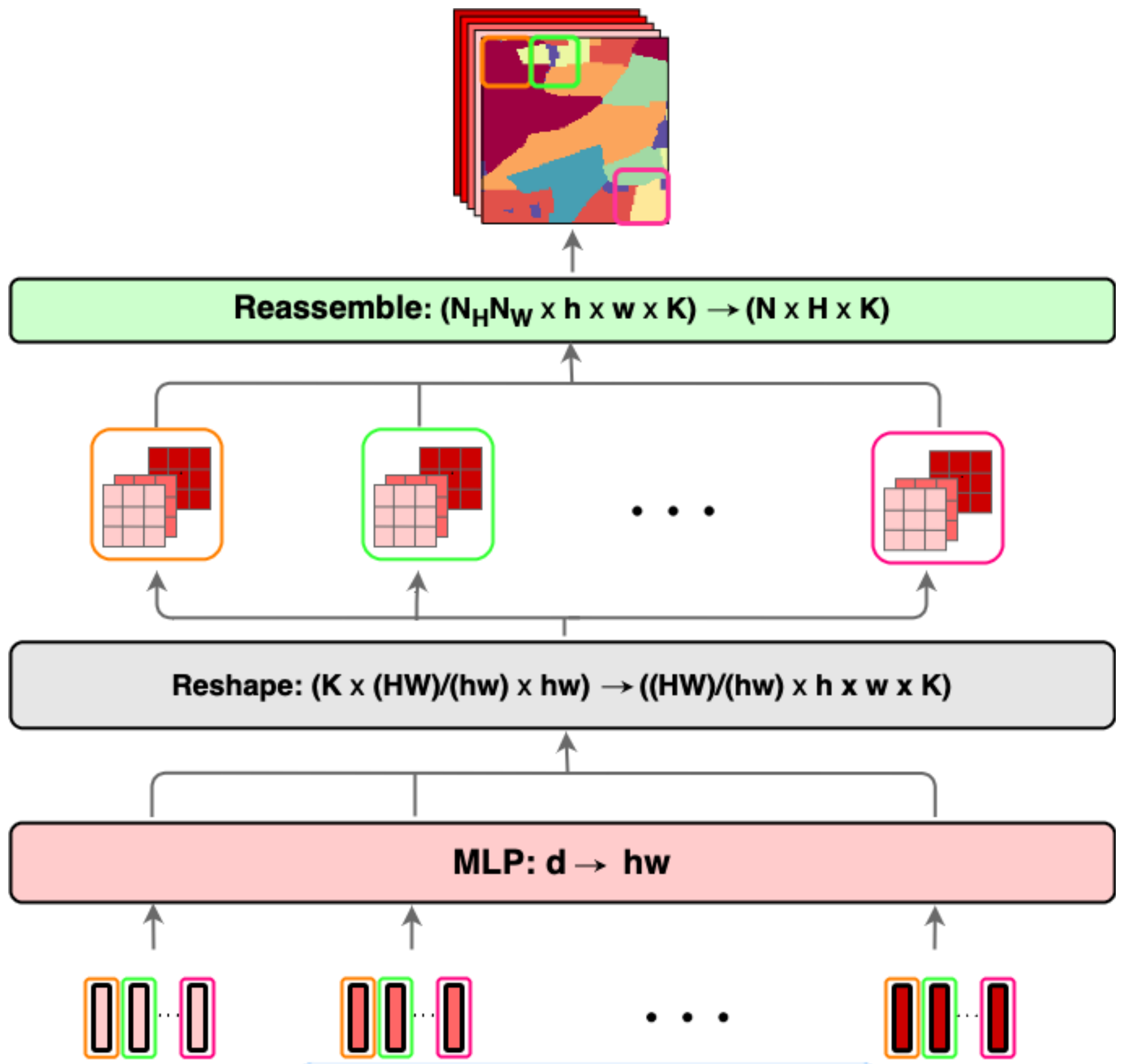
Spatial Encoder architecture


Transformer
concat(Z$_{cls}$, Z + P$_S$)
Z$_{Scls}$
Z + P$_S$
P$_S$
Reshape: (N$_H$N$_W$ x K x d) ⟶ (K x N$_H$N$_W$ x d)

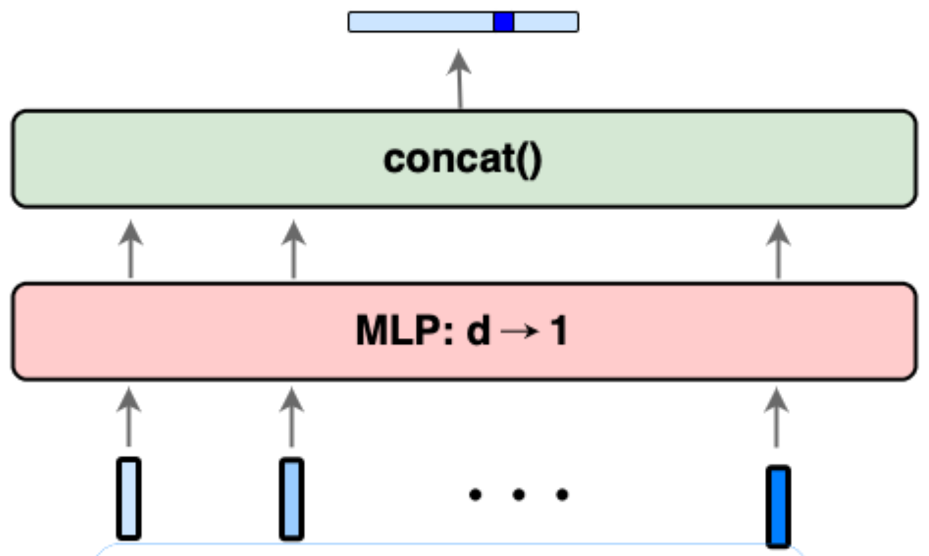
(b) Spatial Encoder

- Reshape input to locations for all K classes
- Static spatial position encodings **P_S**
- Concatenate **Z_{Scls}**
- Process all classes in parallel

Decoder heads architecture


Reassemble: ($N_H N_W$ x h x w x K) → (N x H x K)
. . .
Reshape: (K x (HW)/(hw) x hw) → ((HW)/(hw) x h x w x K)
MLP: d → hw
. . .

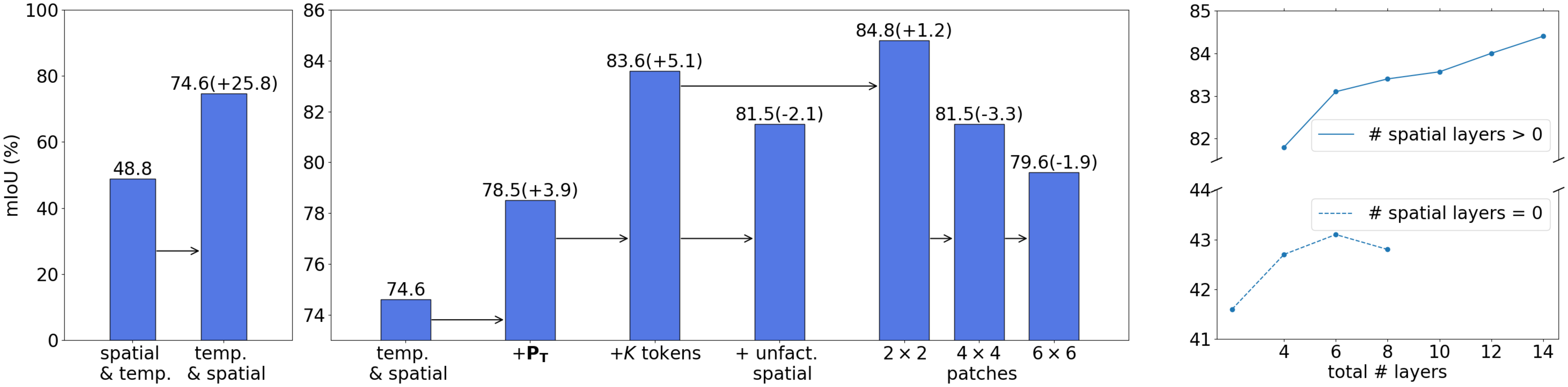
(c) Segmentation Head



(d) Classification Head

- Tokens separated into $[\mathbf{Z}^{L}_{Sglobal} \mid \mathbf{Z}^{L}_{Slocal}]$
- Each token responsible for specific class logits.
- Segmentation head ($\mathbf{Z}^{L}_{Slocal}$)
  - Token to patch (single class)
  - Reassemble patches to size HxWxK logits
- Classification head ($\mathbf{Z}^{L}_{Sglobal}$)
  - Token to scalar
  - Concatenate to size K logits

Ablations



- Order of factorization most important design choice (+25.8% mIoU)

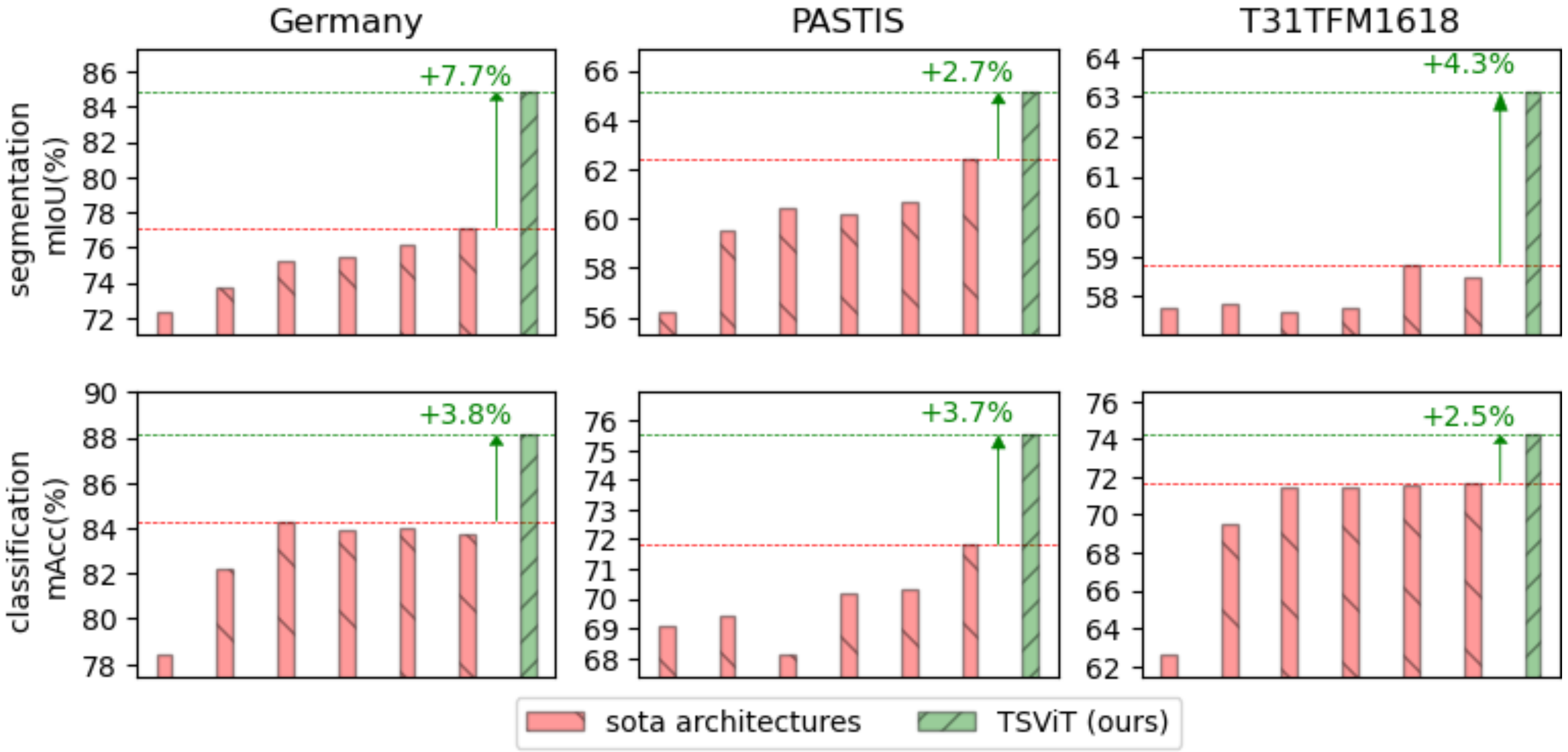
- $\mathbf{P}_T$ and K tokens improve performance

- Inter-class spatial interactions expensive ($O(K^2)$ vs O(K)) and less performant

- Clear performance deterioration with decreasing patch size

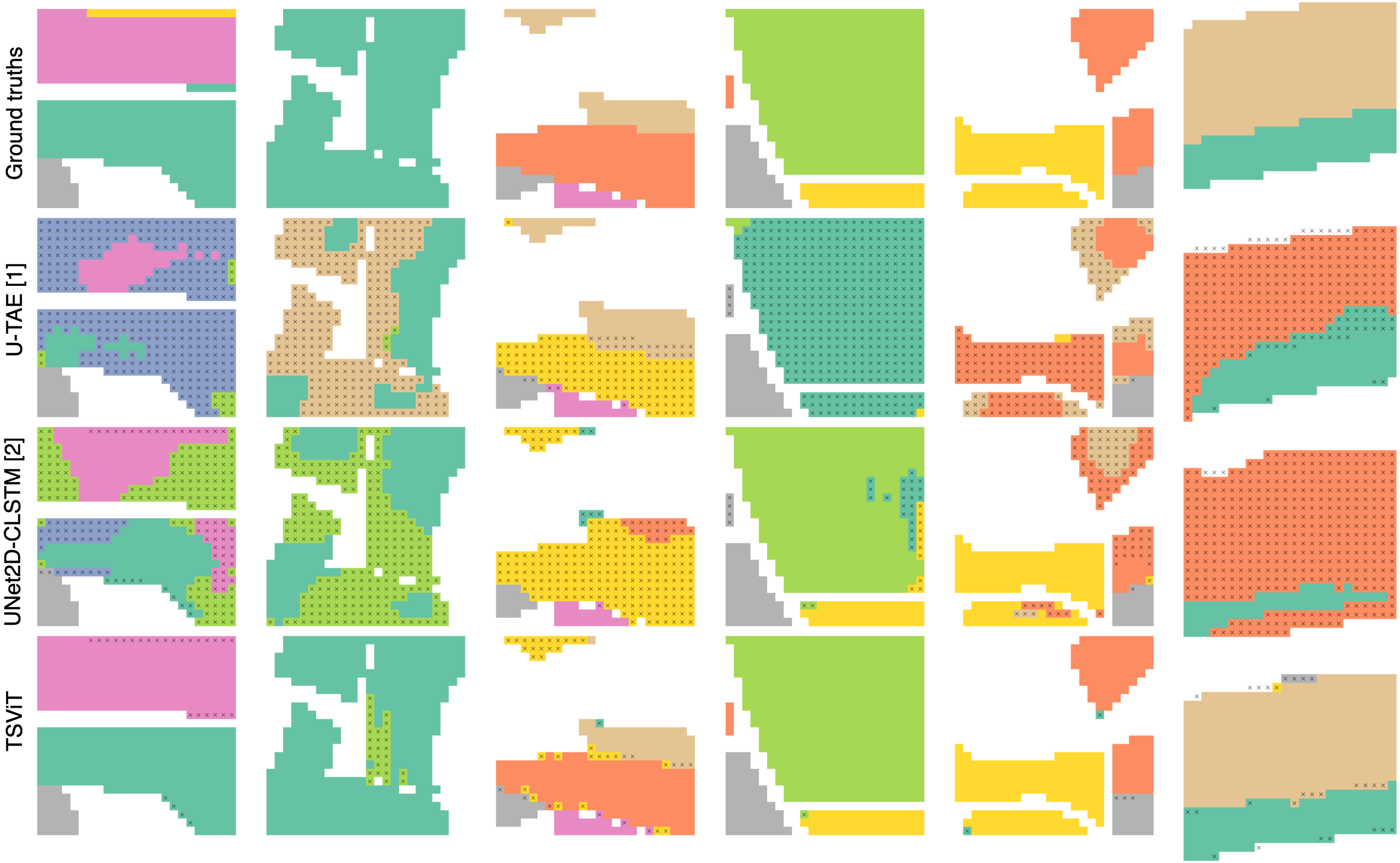
- Spatial encoder is essential for functionality, depth improves performance

Comparison with state-of-the-art



- State-of-the-art performance in SITS classification and segmentation in three publicly available datasets

Comparison with state-of-the-art


Ground truths
U-TAE [1]
UNet2D-CLSTM [2]
TSViT