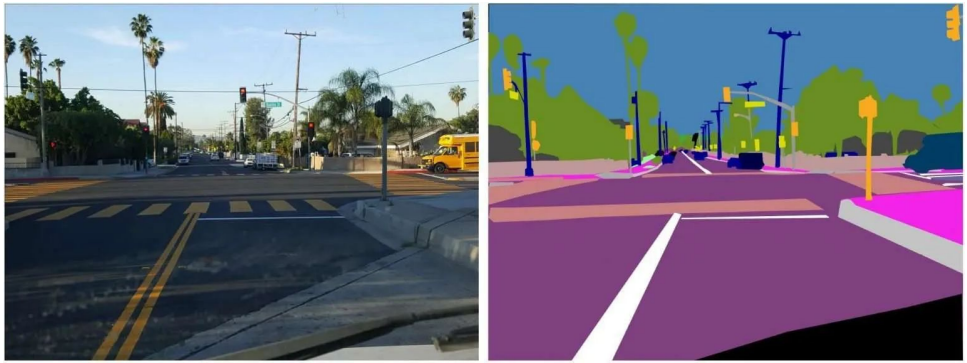
# Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers

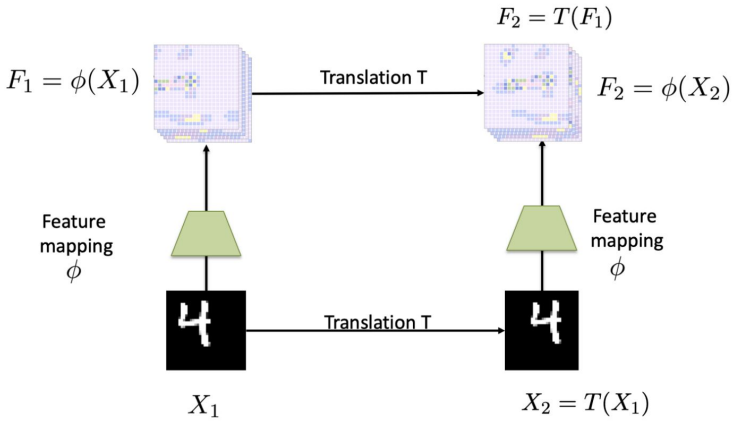
Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, Li Zhang

CVPR 2021

# Background

- Most recent semantic segmentation methods adopt a fully-convolutional network (FCN) with an encoder-decoder architecture.
  - Encoder
    - reduces the resolution, learns more abstract features
  - Decoder
    - upsamples the feature representations into a full-resolution segmentation map
  - Merits
    - Translation Equivariance & Locality





# Motivation

- Problems
  - Stacked encoder layers helps with extracting global abstract information.
  - But, the encoder would diminish rapidly once reaching certain depths.
  - Contradiction between encoder depth and size of receptive fields
- Related Work
  - Enlarge the receptive field
    - Dilated conv
  - Context modeling
    - Decomposed large kernels, PSPNet
  - Attention based models
    - PSANet, DANet
- Motivation
  - By using the encoder transformer, model global context at every layer
  - Keep the same spatial resolution throughout

# Question #1

- Why it is beneficial to maintain resolution (eliminate downsampling)?

# Question #1

- Why it is beneficial to maintain resolution (eliminate downsampling)?
  - Correlations between high-res content is important
  - Because of quadratic complexity w.r.t. the pixel number, conv layers don’t have such high-res correlation.
  - Dependency learning on lower-level feature tensors is lacking, leading to sub-optimal representation learning.

# SETR: SEgmentation TRansformer

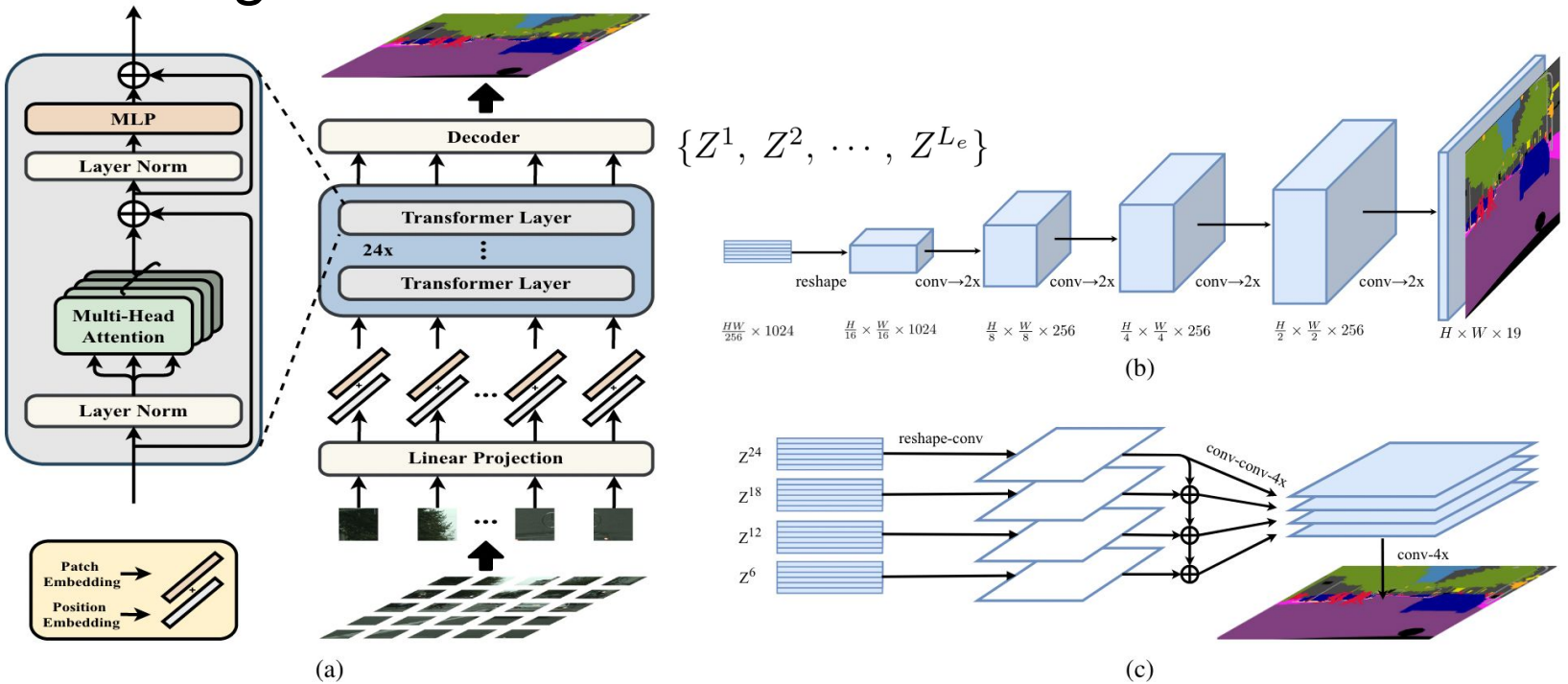
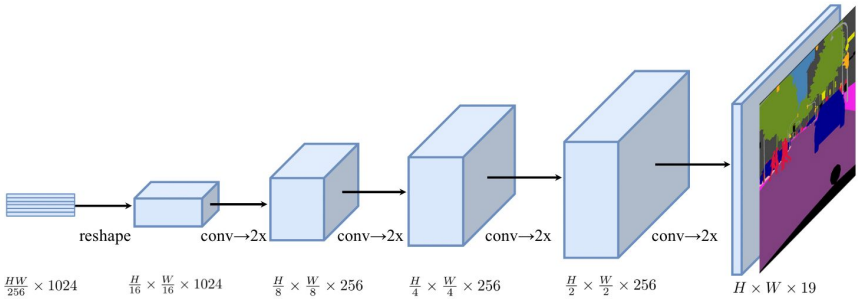



Figure 1. **Schematic illustration of the proposed *SEgmentation TRansformer* (SETR)** (a). We first split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. To perform pixel-wise segmentation, we introduce different decoder designs: (b) progressive upsampling (resulting in a variant called SETR-*PUP*); and (c) multi-level feature aggregation (a variant called SETR-*MLA*).

# Decoder

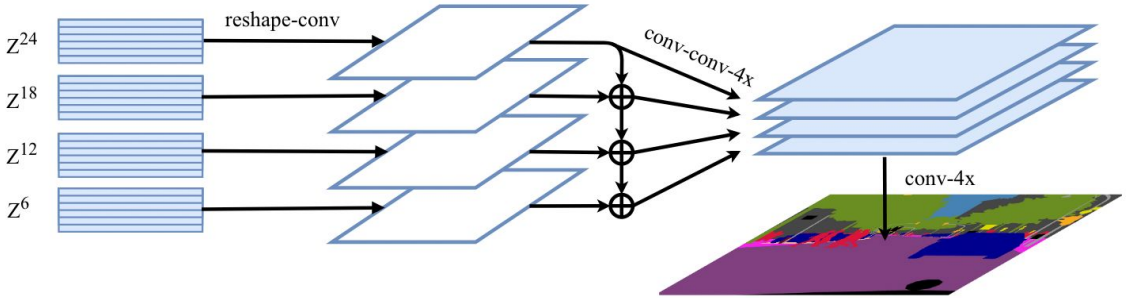
- Naive upsampling (Naive)
  - 1 × 1 conv + sync batch norm (w/ ReLU) + 1 × 1 conv.
  - bilinearly upsample the output to the full image resolution
- Progressive UPsampling (PUP)




(b)

# Decoder

- Multi-Level Feature Aggregation (MLA)
  - Use different transformer layers
  - Element wise addition
  - Channel-wise concatenation
  - Upsampled 4× twice to the full resolution.




(c)

# Experiments-Setting

- Datasets
  - Cityscapes
  - ADE20K
  - PASCAL Context
- Data augmentation and training schedule
  - Follow the default setting (e.g., data augmentation) of public codebase mmsegmentation
  - random resize, random cropping, random horizontal flipping
- Auxiliary loss
  - SETR-Naive (Z_10, Z_15, Z_20)
  - SETR-PUP (Z_10, Z_15, Z_20, Z_24)
  - SETR-MLA (Z_6, Z_12, Z_18, Z_24)
- Multi-scale test
  - Multi-scale scaling and random horizontal flip
- Baselines
  - Dilated FCN
  - Semantic FPN

# Experiment Setting

- SETR variants
  - Decoders: SETR-Naive, SETR-PUP, SETR-MLA
  - Encoders: T-Base (12 layers), T-Large (24 layers)

| Model | T-layers | Hidden size | Att head |
|---|---|---|---|
| T-Base | 12 | 768 | 12 |
| T-Large | 24 | 1024 | 16 |

Table 1. Configuration of Transformer backbone variants.

- Hybrid baseline
  - a ResNet-50 based FCN encoder and feeding its output feature into SETR
- Pre-training
  - pre-trained weights provided by ViT or DeiT
  - All the layers without pre-training are randomly initialized.
- Evaluation metric
  - mIoU, pixel-wise accuracy (for ADE20K)

# Ablation studies-Cityscape

| Method | Pre | Backbone | #Params | 40k | 80k |
|---|---|---|---|---|---|
| FCN [38] | 1K | R-101 | 68.59 | 73.93 | 75.52 |
| Semantic FPN [38] | 1K | R-101 | 47.51 | - | 75.80 |
| *Hybrid-Base* | R | T-Base | 112.59 | 74.48 | 77.36 |
| *Hybrid-Base* | 21K | T-Base | 112.59 | 76.76 | 76.57 |
| *Hybrid-DeiT* | 21K | T-Base | 112.59 | 77.42 | 78.28 |
| SETR-*Naïve* | 21K | T-Large | 305.67 | 77.37 | 77.90 |
| SETR-*MLA* | 21K | T-Large | 310.57 | 76.65 | 77.24 |
| SETR-*PUP* | 21K | T-Large | 318.31 | 78.39 | 79.34 |
| SETR-*PUP* | R | T-Large | 318.31 | 42.27 | - |
| SETR-*Naïve-Base* | 21K | T-Base | 87.69 | 75.54 | 76.25 |
| SETR-*MLA-Base* | 21K | T-Base | 92.59 | 75.60 | 76.87 |
| SETR-*PUP-Base* | 21K | T-Base | 97.64 | 76.71 | 78.02 |
| SETR-*Naïve-DeiT* | 1K | T-Base | 87.69 | 77.85 | 78.66 |
| SETR-*MLA-DeiT* | 1K | T-Base | 92.59 | 78.04 | 78.98 |
| SETR-*PUP-DeiT* | 1K | T-Base | 97.64 | **78.79** | **79.45** |

- SETR-PUP has the best performance among all the variants
- “T-Large” are always superior to their “T-Base” counterparts
- SETR-PUP-Base (76.71) performs worse than Hybrid-Base (76.76), it shines (78.02) when training with more iterations (80k)
  - Question #2 why?
- Pre-training is critical for our model.

# Power of pre-training

| Method | Pre | Backbone | ADE20K | Cityscapes |
|---|---|---|---|---|
| FCN [38] | 1K | R-101 | 39.91 | 73.93 |
| FCN | 21K | R-101 | 42.17 | 76.38 |
| SETR-*MLA* | 21K | T-Large | **48.64** | 76.65 |
| SETR-*PUP* | 21K | T-Large | 48.58 | 78.39 |
| SETR-*MLA-DeiT* | 1K | T-Large | 46.15 | 78.98 |
| SETR-*PUP-DeiT* | 1K | T-Large | 46.24 | **79.45** |

Table 3. **Comparison to FCN with different pre-training** with single-scale inference on the ADE20K val and Cityscapes val set.

- The advantage of the approach largely comes from the proposed sequence-to-sequence modeling strategy rather than bigger pre-training data.

# Results on ADE20K

| Method | Backbone | mIoU | Pixel Acc. |
|---|---|---|---|
| FCN (16, 160k, SS) [38] | ResNet-101 | 39.91 | 79.52 |
| FCN (16, 160k, MS) [38] | ResNet-101 | 41.40 | 80.65 |
| EncNet [53] | ResNet-101 | 44.65 | 81.69 |
| PSPNet [58] | ResNet-269 | 44.94 | 81.69 |
| DMNet [17] | ResNet-101 | 45.50 | - |
| CCNet [24] | ResNet-101 | 45.22 | - |
| Strip pooling [22] | ResNet-101 | 45.60 | 82.09 |
| APCNet [18] | ResNet-101 | 45.38 | - |
| OCNet [52] | ResNet-101 | 45.45 | - |
| SETR-*Naïve* (16, 160k, SS) | T-Large | 48.06 | 82.40 |
| SETR-*Naïve* (16, 160k, MS) | T-Large | 48.80 | 82.92 |
| SETR-*PUP* (16, 160k, SS) | T-Large | 48.58 | 82.90 |
| SETR-*PUP* (16, 160k, MS) | T-Large | 50.09 | **83.58** |
| SETR-*MLA* (16, 160k, SS) | T-Large | 48.64 | 82.64 |
| SETR-*MLA* (16, 160k, MS) | T-Large | **50.28** | 83.46 |

Table 4. **State-of-the-art comparison on the ADE20K dataset.** Performances of different model variants are reported. SS: Single-scale inference. MS: Multi-scale inference.

In ADE20K, SETRs outperforms all CNN based methods

# Results on Pascal Context

| Method | Backbone | mIoU |
|---|---|---|
| FCN (16, 80k, SS) [38] | ResNet-101 | 44.47 |
| FCN (16, 80k, MS) [38] | ResNet-101 | 45.74 |
| PSPNet [58] | ResNet-101 | 47.80 |
| DANet [16] | ResNet-101 | 52.60 |
| EMANet [30] | ResNet-101 | 53.10 |
| SVCNet [14] | ResNet-101 | 53.20 |
| Strip pooling [22] | ResNet-101 | 54.50 |
| GFFNet [29] | ResNet-101 | 54.20 |
| APCNet [18] | ResNet-101 | 54.70 |
| SETR-*Naïve* (16, 80k, SS) | T-Large | 52.89 |
| SETR-*Naïve* (16, 80k, MS) | T-Large | 53.61 |
| SETR-*PUP* (16, 80k, SS) | T-Large | 54.40 |
| SETR-*PUP* (16, 80k, MS) | T-Large | 55.27 |
| SETR-*MLA* (16, 80k, SS) | T-Large | 54.87 |
| SETR-*MLA* (16, 80k, MS) | T-Large | **55.83** |

Table 5. **State-of-the-art comparison on the Pascal Context dataset.** Performances of different model variants are reported. SS: Single-scale inference. MS: Multi-scale inference.

In Pascal Context, SETR variants don't necessarily outperform all CNN based methods

# Results on Cityscapes

| Method | Backbone | mIoU |
|---|---|---|
| FCN (40k, SS) [38] | ResNet-101 | 73.93 |
| FCN (40k, MS) [38] | ResNet-101 | 75.14 |
| FCN (80k, SS) [38] | ResNet-101 | 75.52 |
| FCN (80k, MS) [38] | ResNet-101 | 76.61 |
| PSPNet [58] | ResNet-101 | 78.50 |
| DeepLab-v3 [9] (MS) | ResNet-101 | 79.30 |
| NonLocal [47] | ResNet-101 | 79.10 |
| CCNet [24] | ResNet-101 | 80.20 |
| GCNet [3] | ResNet-101 | 78.10 |
| Axial-DeepLab-XL [46] (MS) | Axial-ResNet-XL | 81.10 |
| Axial-DeepLab-L [46] (MS) | Axial-ResNet-L | 81.50 |
| SETR-*PUP* (40k, SS) | T-Large | 78.39 |
| SETR-*PUP* (40k, MS) | T-Large | 81.57 |
| SETR-*PUP* (80k, SS) | T-Large | 79.34 |
| SETR-*PUP* (80k, MS) | T-Large | **82.15** |

Table 6. **State-of-the-art comparison on the Cityscapes validation set.** Performances of different training schedules (*e.g.*, 40k and 80k) are reported. SS: Single-scale inference. MS: Multi-scale inference.

- In Cityscapes, SETR variants don't necessarily outperform all CNN based methods.Does multi-scale inference bring the boost?
- Better than Axial-DeepLab, which aims to use an attention-alone model but still follows the basic structure of FCN
- Didn't run Axial DeepLab on other datasets

# Qualitative Results

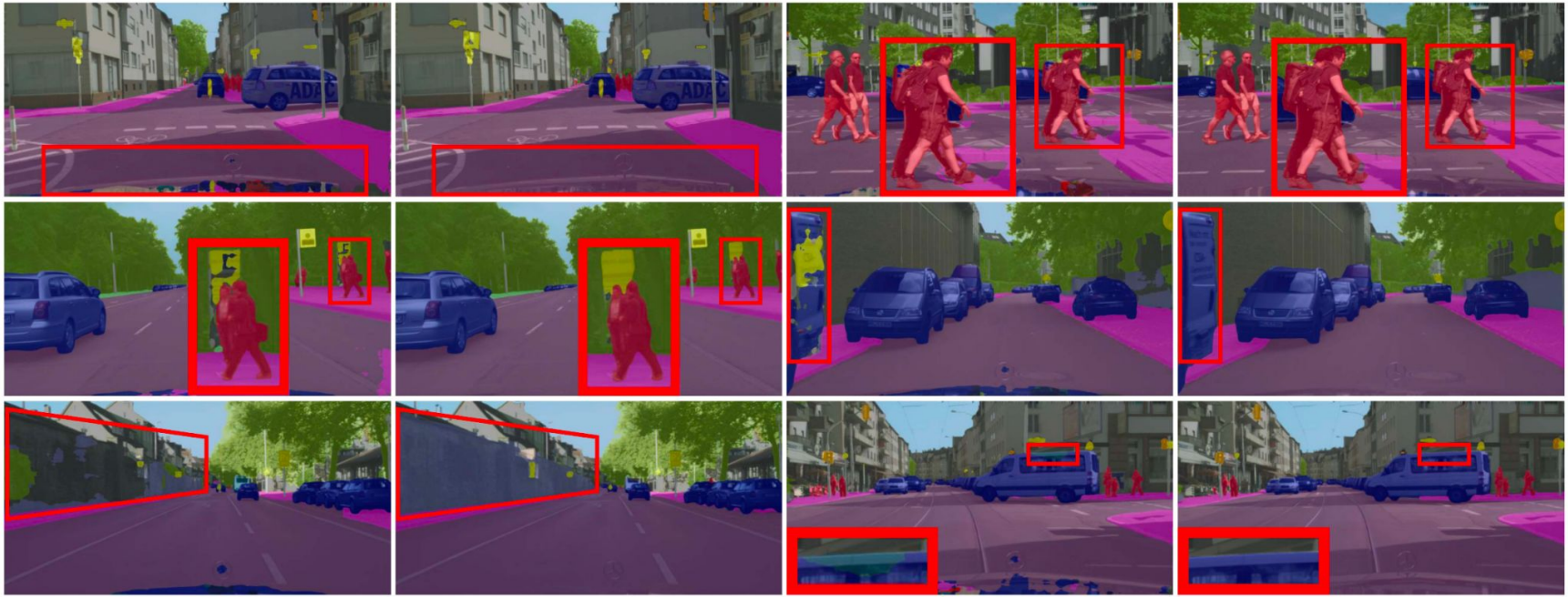


Figure 4. **Qualitative results on Cityscapes:** SETR (right column) vs. dilated FCN baseline (left column) in each pair. Best viewed in color and zoom in.

# Discussion

- Was Question #1 perfectly answered?
  - Why it is beneficial to maintain resolution (eliminate downsampling)?

# Discussion

- Was Question #1 perfectly answered?
  - Why it is beneficial to maintain resolution (eliminate downsampling)?
  - Yes, but not totally. Interesting to see
    - In transformer, what if we reduce the patch resolution/size?
    - In encoder-decoder based network, what if we use more attentions?

# Discussion

- How much benefit can SETR bring for the small object?
  - Can attention/correlation still benefit small objects segmentation in the patch?

# Discussion

- How much benefit can SETR bring for the small object (smaller than patch size)?
  - The linear projection might be more critical for grasping tiny objects than the attention
  - Should include more diverse dataset to see