CAPE: Encoding Relative Positions with Continuous Augmented Positional Embeddings

Tatiana Likhomanenko[1], Qiantong Xu[1], Ronan Collobert[1], Gabriel Synnaeve[1], Alex Rogozhnikov[2]
[1]Facebook AI Research, [2]Herophilus Inc.

NeurIPS 2021

Motivation & Overview

No positions	Transformer's attention		is	permutational invariant		
Absolute positions	1	**2**	3	4	5	simple efficient
Relative positions	$i - j = -1$	$\boldsymbol{i - j = 0}$	$i - j = 1$	$i - j = 2$	$i - j = 3$	generalize better
CAPE	$\Delta + 1$	$\Delta + 2$	$\Delta + 3$	$\Delta + 4$	$\Delta + 5$	simple efficient generalizes
	$\in \mathbb{R}$		$i - j = 2$			

Continuous Augmented Positional Embedding (CAPE)

Sinusoidal Positional Embedding (*sinpos*)

For texts: $E_k(n) = e^{i\omega_k n}$ $\quad \omega_k = 10000^{-k/K}$

For audio: $E_k(t) = e^{i\omega_k t}$ $\quad \omega_k = 30 \cdot 10000^{-k/K}$ $\quad t \in \mathbb{R}$

- tie positions to timestamps *t* in seconds
- select necessary scale to guarantee 30 ms specificity (minimal audible gap)

For images: $E_k(x, y) = e^{i\omega_{k,x}x + i\omega_{k,y}y}$ $\quad \omega_{k,x} = 10^{k/K} \sin k$ $\quad \omega_{k,y} = 10^{k/K} \cos k$

- Coordinates *x* and *y* are scaled to *[-1, 1].*

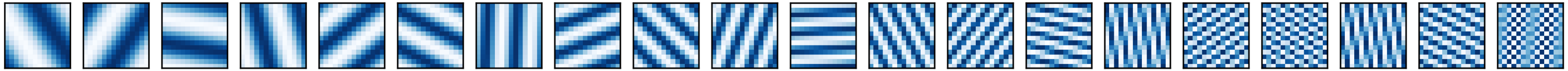
Shared property of sinusoidal embeddings: unitary translation operators *S*

$$\mathbf{E}(n+1) = S\,\mathbf{E}(n) \qquad \mathbf{E}(t+\delta) = S_t^{\delta}\,\mathbf{E}(t) \qquad \mathbf{E}(x+\delta_x, y+\delta_y) = S_x^{\delta_x} S_y^{\delta_y}\,\mathbf{E}(x, y)$$

Sinusoidal Positional Embedding for Images (2D)

Scale positions coordinates to [-1, +1] and

$$E_k(x, y) = e^{i\omega_{k,x}x + i\omega_{k,y}y} \qquad \omega_{k,x} = 10^{k/K} \sin k \quad \omega_{k,y} = 10^{k/K} \cos k$$

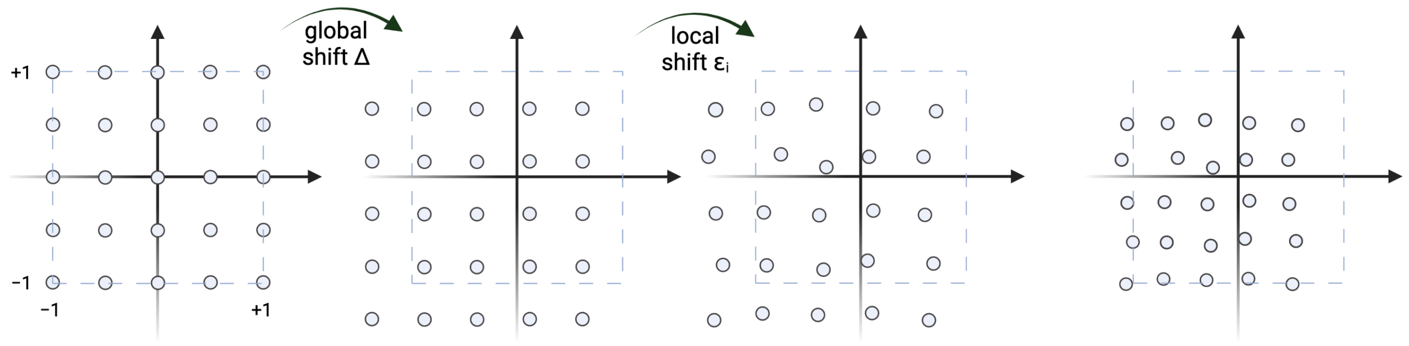


E_0 E_{20} E_{40} …

- No *selected directions* on the plane via different frequencies for axes
- Angle of *hatching* via *inner* cosine and sine
- Different *hatching densities* allow both precise and approximate positioning

Continuous Augmented Positional Embedding (CAPE)

- Apply **positions augmentation** (during training only)
- Use sinusoidal positional (1D/2D) embedding



- Global/local shifts and global scaling are sampled from uniform distribution

Behind CAPE

- No capacity increase + computationally efficient
- We force model to learn querying relative positions, no explicit mechanism
- Large global shift (and scaling) provide positions not seen during training, thus, model is able to generalize on longer inputs
- Global shift breaks spontaneous correlations between content and position embeddings
- Scaling breaks potential memorization of relative positions

Empirical Evaluation

- Image recognition
- Speech recognition
- Machine translation

Study generalization on longer sequences not seen during training

Image Recognition

Setup

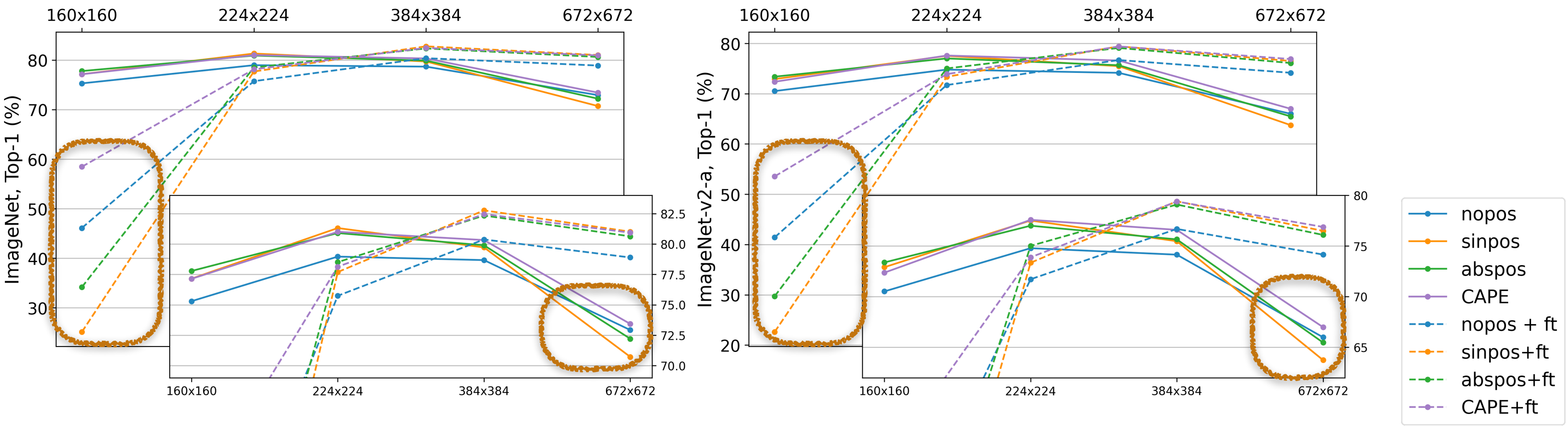
- Classification problem on ImageNet
- Baseline: vanilla ViT model* + DeiT optimization scheme**
  learnable absolute positional embedding (*abspos*);
- Vary **only** positional embedding and training data resolution
- Fine-tune on higher resolution (224x224 → 384x384)
- Test on ImageNet-val and ImageNet-v2{a,b,c}
- Test generalization on {160x160, 228x228, 384x384, 640x640} resolutions

Note: *abspos* is upsampled/downsampled via bicubic interpolation, according to **

*Dosovitskiy A, et.al. An image is worth 16x16 words: Transformers for image recognition at scale. ICLR 2021.

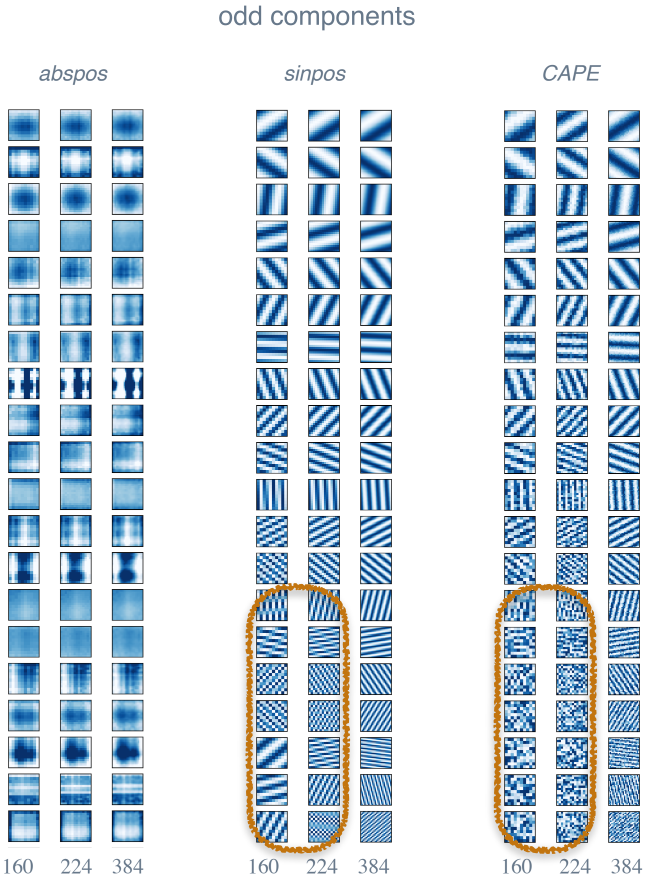
**Touvron H, et.al. Training data-efficient image transformers & distillation through attention. ICML 2021. PMLR.

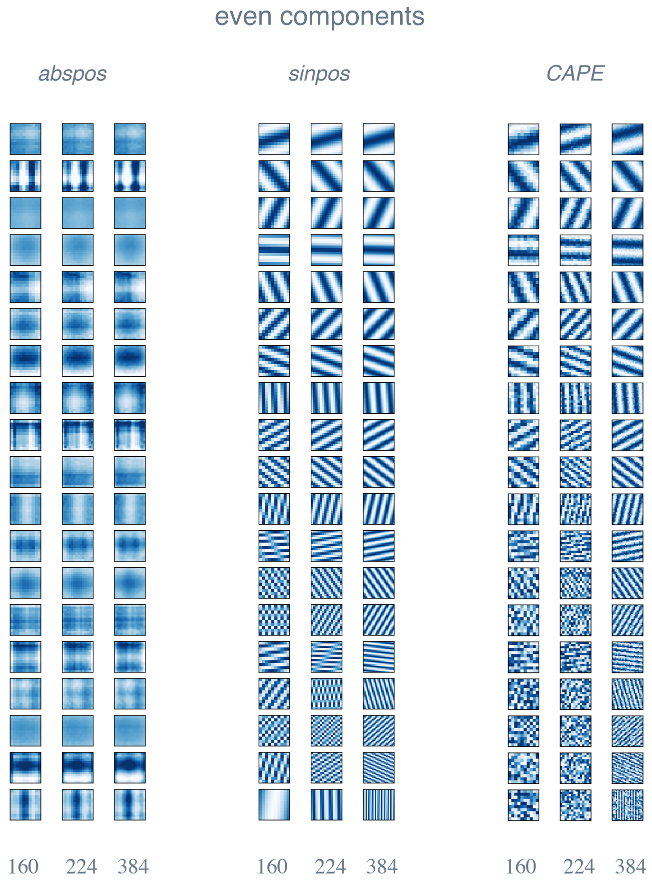
Result: Positional Embedding Generalization



- On resolutions different from training one, CAPE performs best, notably outperforming on high and low resolutions

Positional Embedding Visualization





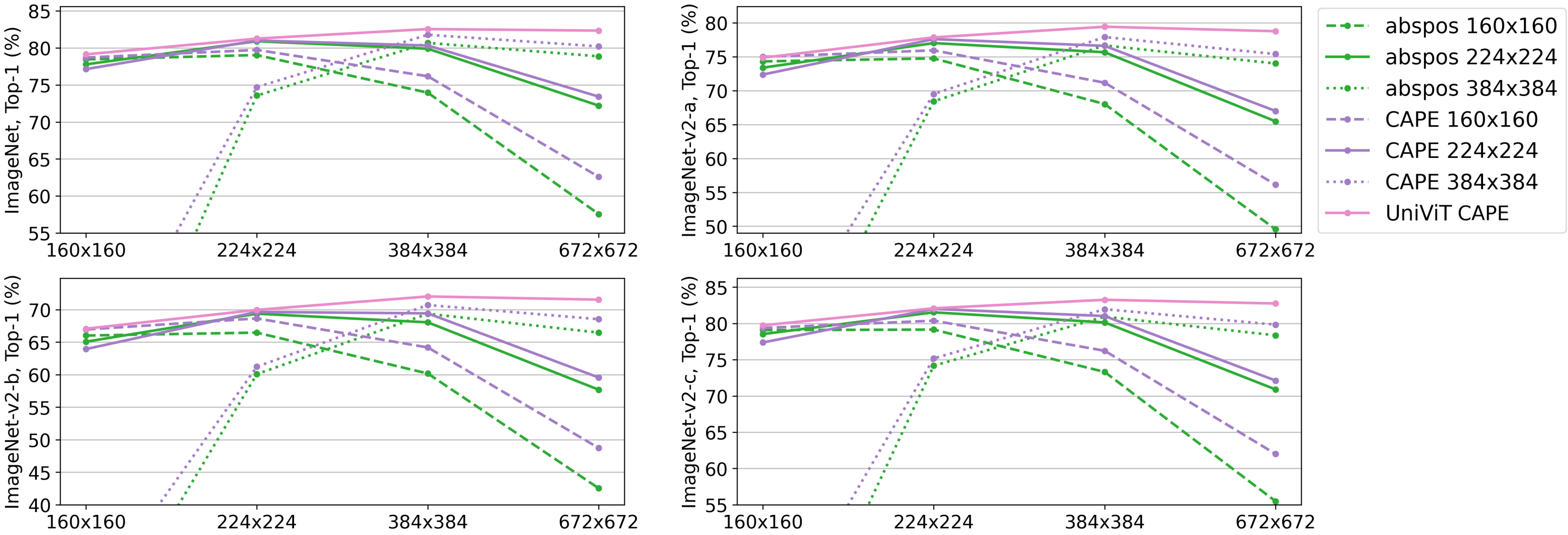
CAPE allows to train on images of *different resolutions*

UniViT — new training paradigm for ViT

New Training Paradigm: Universal ViT (UniViT)

- We propose training a single Universal Vision Transformer (UniViT) on different resolutions:
  - ViT model with proper positional embedding
  - Training is done on image batches which are randomly resized to {128, 160, 192, 224, 256, 288, 320}
- For experiments the rest of training configuration remains the same as for ViT

Result: VIT vs UniViT

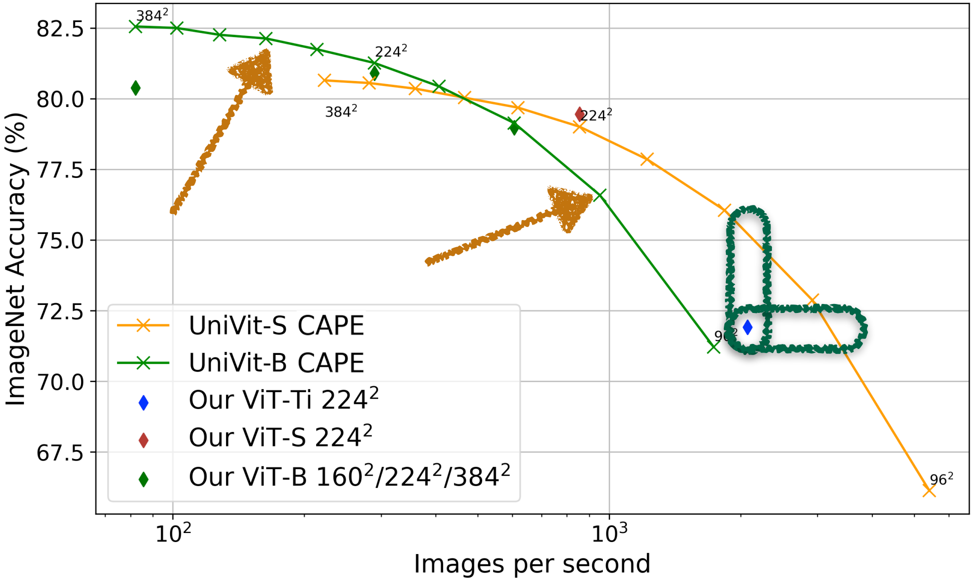


- CAPE generalizes better to other resolutions than *abspos*
- UniViT outperforms single-resolution ViT models
- UniViT does not need pre-training on lower resolution

Adjustable Inference: Resolution Scale vs Model Scale

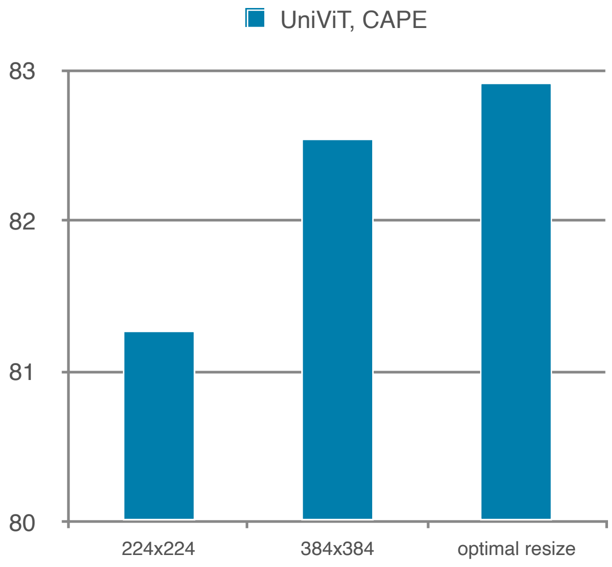
UniViT unlocks dynamically adjusting throughput at inference time, a practical alternative to improving model throughput via decreasing model size

Image resolution directly impacts throughput: computational complexity of attention $O(N^4)$



Optimal Resizing for Evaluation

Find an optimal resizing strategy for each image during evaluation:
mostly, the best strategy is to use the **original size**



Speech Recognition (ASR)

Setup

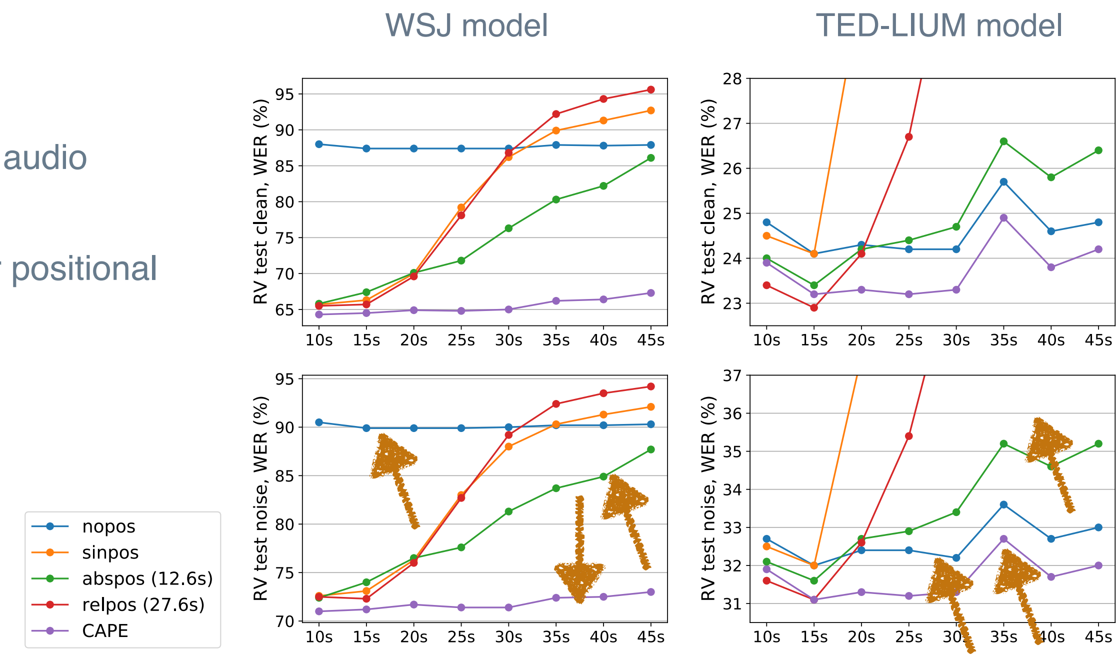
- Letter-based CTC Transformer model
- Train data: either WSJ (80h) or TED-LIUM v3 (450h)
- Vary **only** positional embedding
- Test on clean and noisy in-house data

  segment the same data with different durations: 10s, 15s, 20s, 25s, 30s, 35s, 40s, 45s

Note: *abspos* covers $N{=}13.8s$ and for $t > N \;\; E(t) = E(t \bmod N)$

Result: Generalization to Long Audio Duration

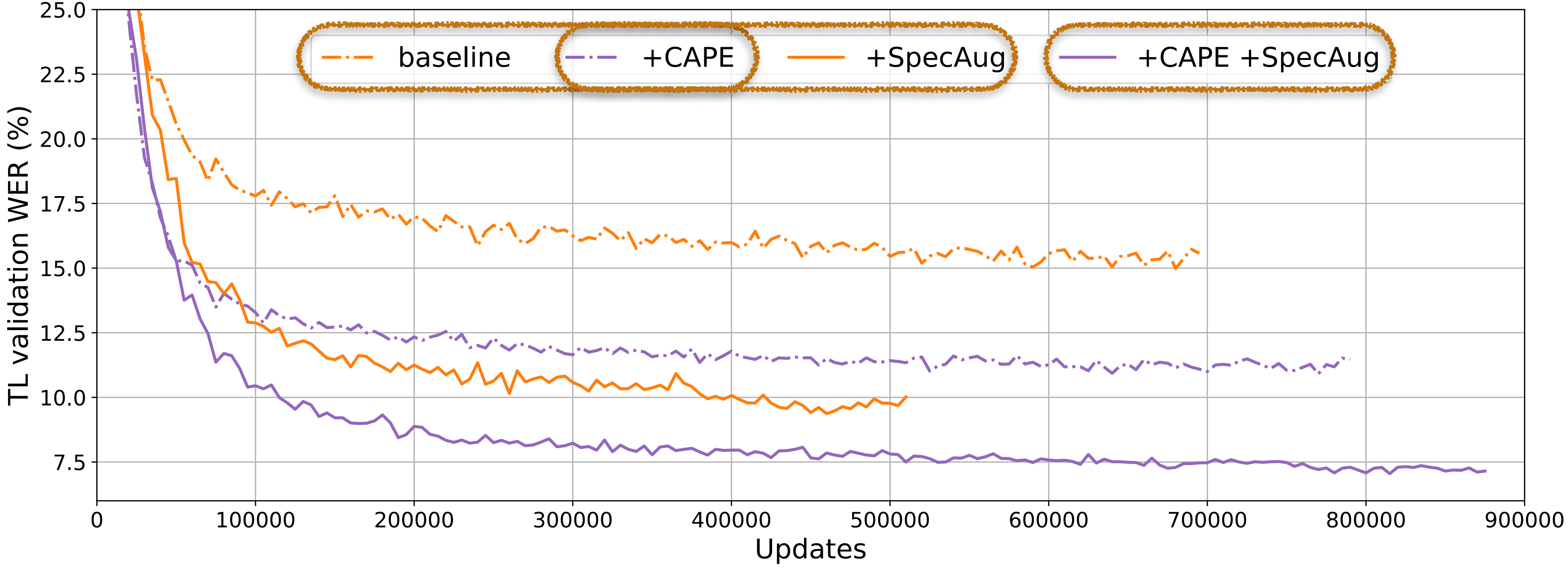
- CAPE performs uniformly well on different audio durations, including 45s duration
- CAPE behaves similar or outperform other positional embeddings for training-duration test sets



Note: CAPE covers 1min duration via global shift, while *relpos* covers 30s to the left/right (the whole training duration)

CAPE’s Augmentation Effect

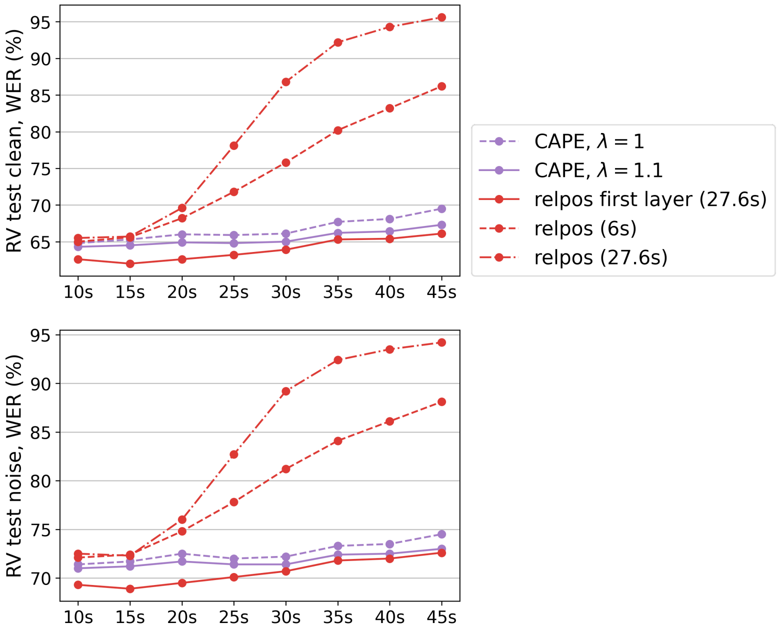
CAPE performs positions augmentation which is orthogonal to data augmentation (SpecAugment)



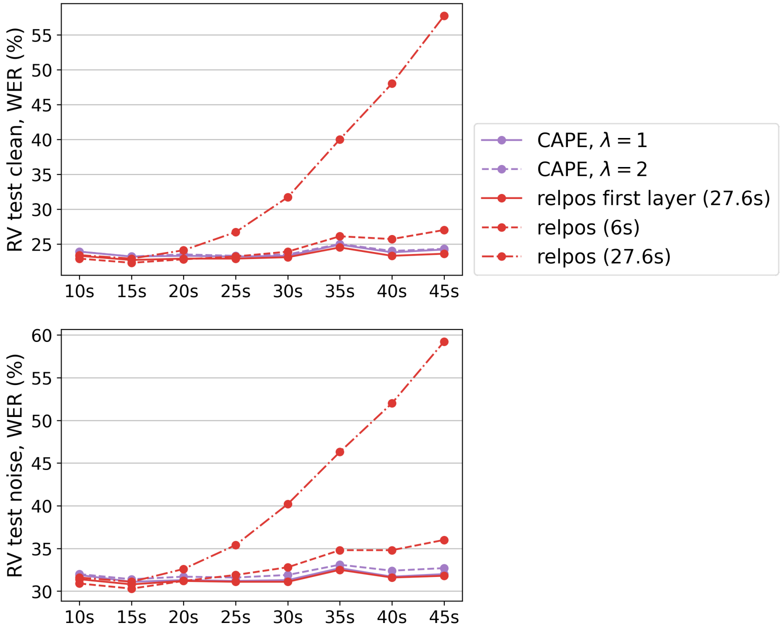
Where To Place Relative Positional Embedding?

CAPE's ability to learn spatial relations hints that *relpos* could be used only in the first Transformer layer

WSJ model



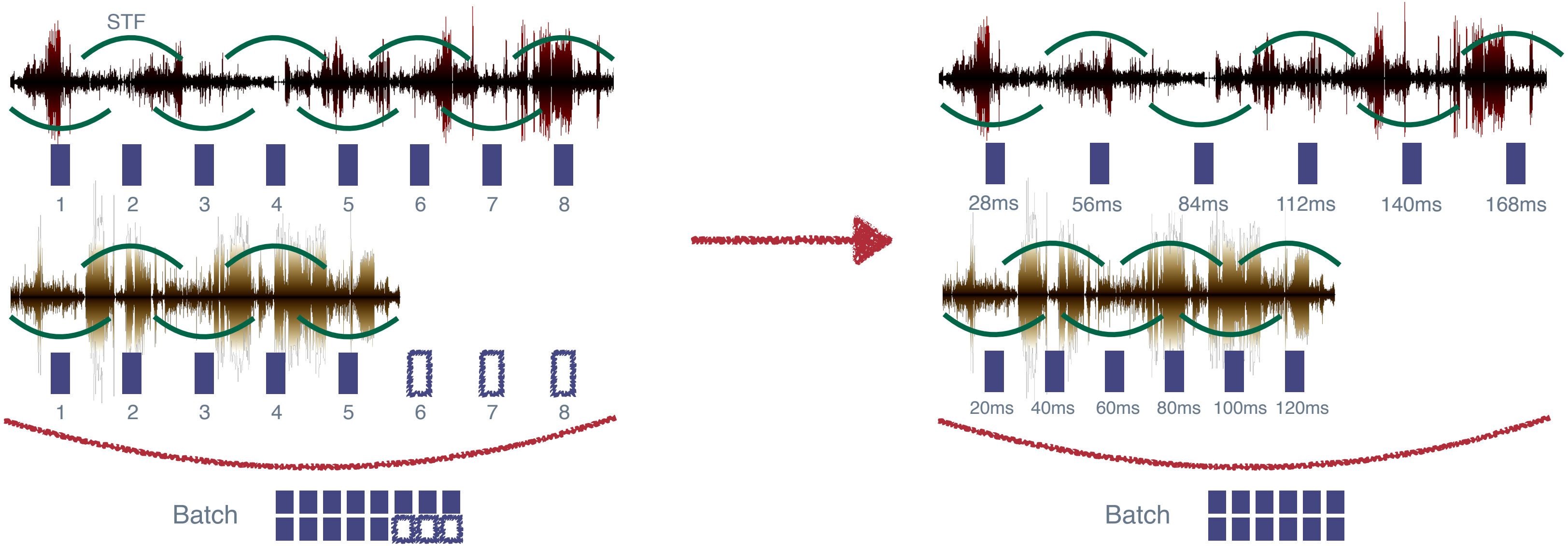
TED-LIUM model



CAPE allows *padding-free* pipeline by tying positions to timestamps

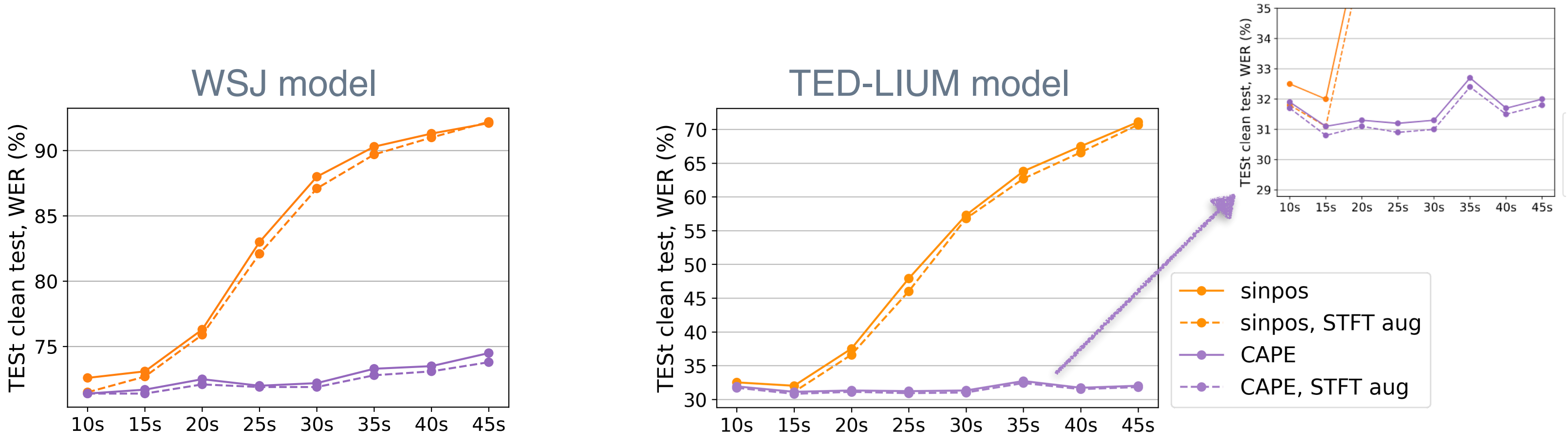
Padding-free ASR

- In ASR, when batching sequences of different length, padding tokens are used
- We propose pipeline simplification with CAPE:
  - CAPE embeddings remain tied to the original timestamp of the audio
  - For audio features perform time stretching augmentation by changing STFT hop distance



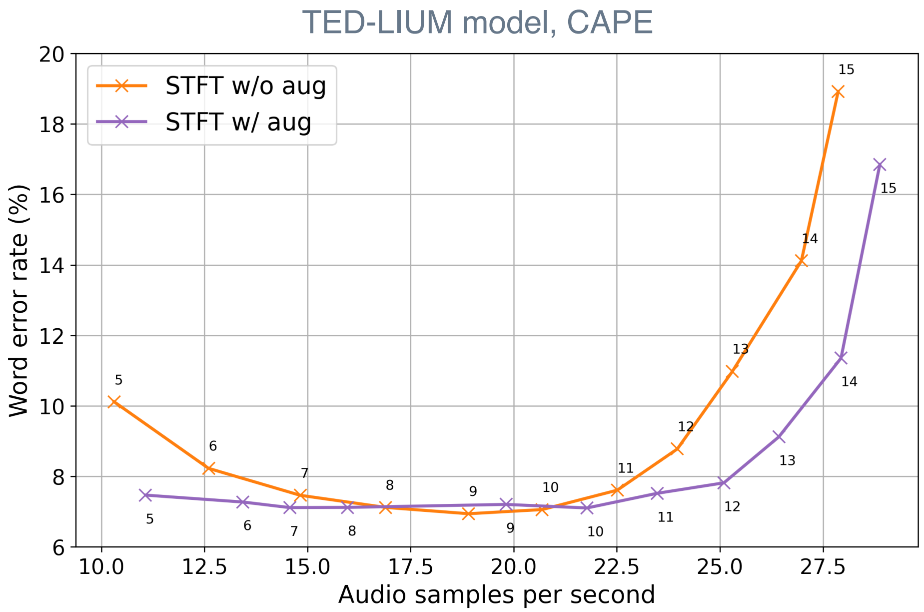
Padding-free ASR

- In ASR, when batching utterances of different sizes, padding tokens are used
- We propose pipeline simplification with CAPE:
  - CAPE embeddings remain tied to the original timestamp of the audio
  - For audio features perform time stretching augmentation by changing STFT hop distance



Adjusting Throughput via STFT

Model trained with STFT hop distance augmentation is less affected by varying STFT hop distance



Machine Translation

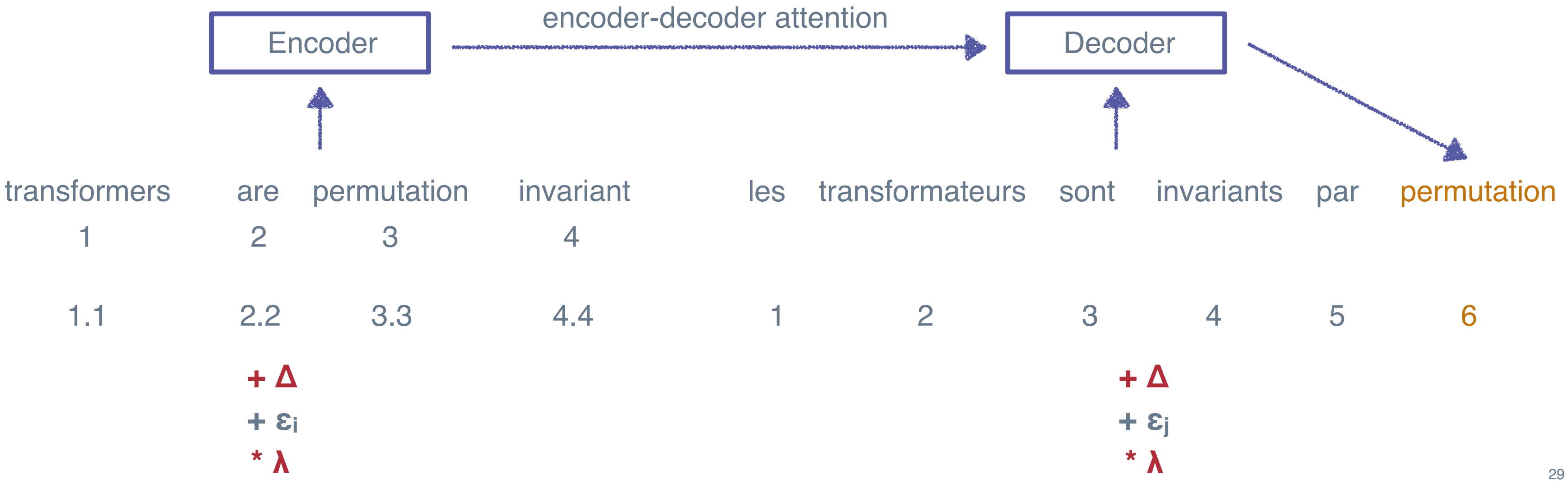
Setup

- Data: WMT'14, English-French (FR) and English-German (DE)
- Baseline: vanilla Transformer with ADMIN initialization scheme*
- Vary **only** positional embedding
- No back-translation or other specific domain data augmentations

*Liu L, Liu X, Gao J, Chen W, Han J. Understanding the Difficulty of Training Transformers. EMNLP 2020.

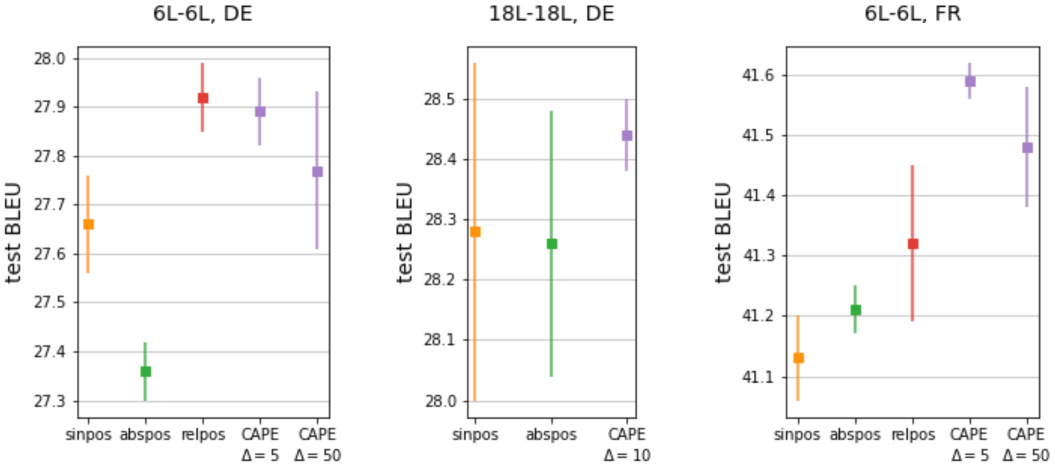
Encoder-Decoder Synchronization

- Scale positions of source language to match the length of target language (via train statistics)
- Apply the **same** global shift and global scaling for both encoder and decoder



Result

- CAPE outperforms *sinpos* and *abspos* for both DE and FR
- CAPE either outperforms *relpos* (FR) or is in the same ballpark (DE)



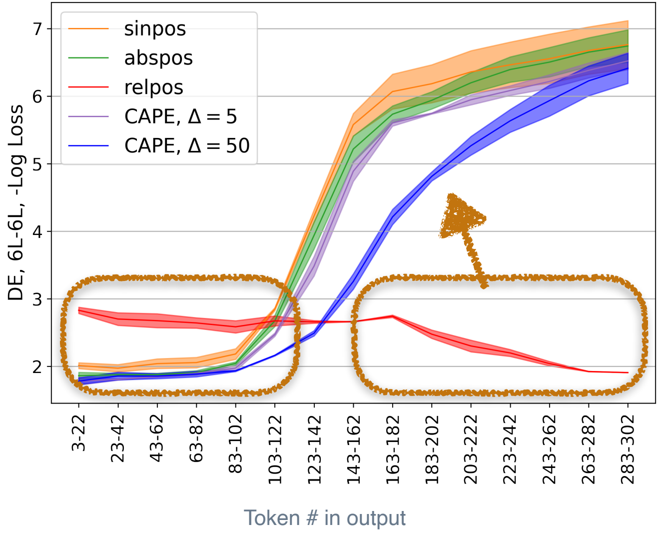
Result: Generalization to Long Sentences

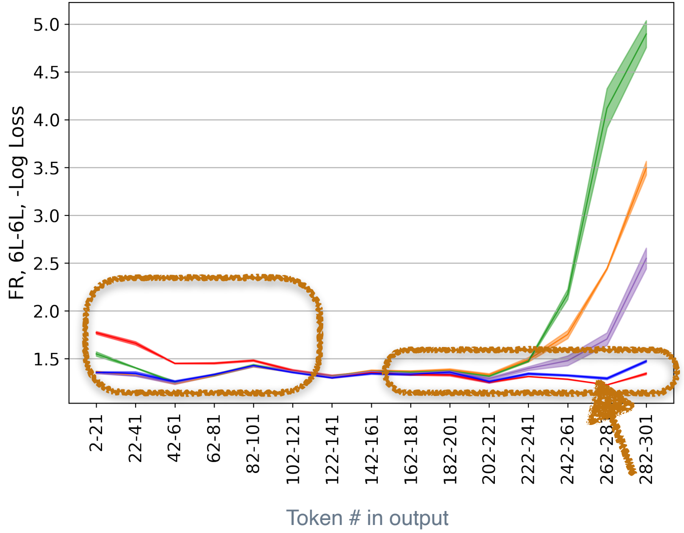
- WMT'14 validation and test sets do not have sentences longer than training
- To test generalization:
  - Stack sentences to form sentences with 300+ tokens
  - Compute average negative log likelihood per position

    (to estimate how well model works at particular position having a true prefix)

Result: Generalization to Long Sentences

- Positions < 100: CAPE, *sinpos*, and *abspos* are similar and outperform *relpos*
- Positions > 200: *relpos* outperforms others but CAPE is able to generalize well too with larger data (FR)





Note: *relpos* covers 150 left/right tokens (the whole training sequences) while CAPE covers only 100 tokens

Summary

- We proposed to augment positions by introducing a **simple** and **efficient** CAPE embedding
    - allows augmentations previously not possible, thanks to continuous nature
    - preserves relative positions between tokens
    - generalizes to input sizes across several domains
    - drop-in replacement for absolute positional embeddings
    - no additional costs compared to *relpos* attention mechanisms

- We introduced **new training and production pipelines**
    - Vision: UniViT — a universal model, able to adjust throughput by changing input resolution
    - ASR: padding-free training pipeline

Thank You