Productionization

Text Editing Advantages

Faithfulness

Constraining decoders in seq2seq is an active area of research

Control

We can control the word a model can add / remove. Can incorporate external knowledge (e.g., pronoun).

Text Editing Advantages

Faithfulness	Constraining decoders in seq2seq is an active area of research
Control	We can control the word a model can add / remove. Can incorporate external knowledge (e.g., pronoun).
Latency	Can be >10x faster inference.
Data efficient	Text Editing models need less training data.

Latency

Case study: EdiT5 vs T5

- Two GEC models:
  - EdiT5 base (12-layer-encoder, 1-layer-decoder)
  - T5 base (12-layer-encoder, 12-layer-decoder)
- Profiles obtained on GPU
  - Profiles obtained with Tensorflow Profiler
  - PyTorch has similar tools

GEC

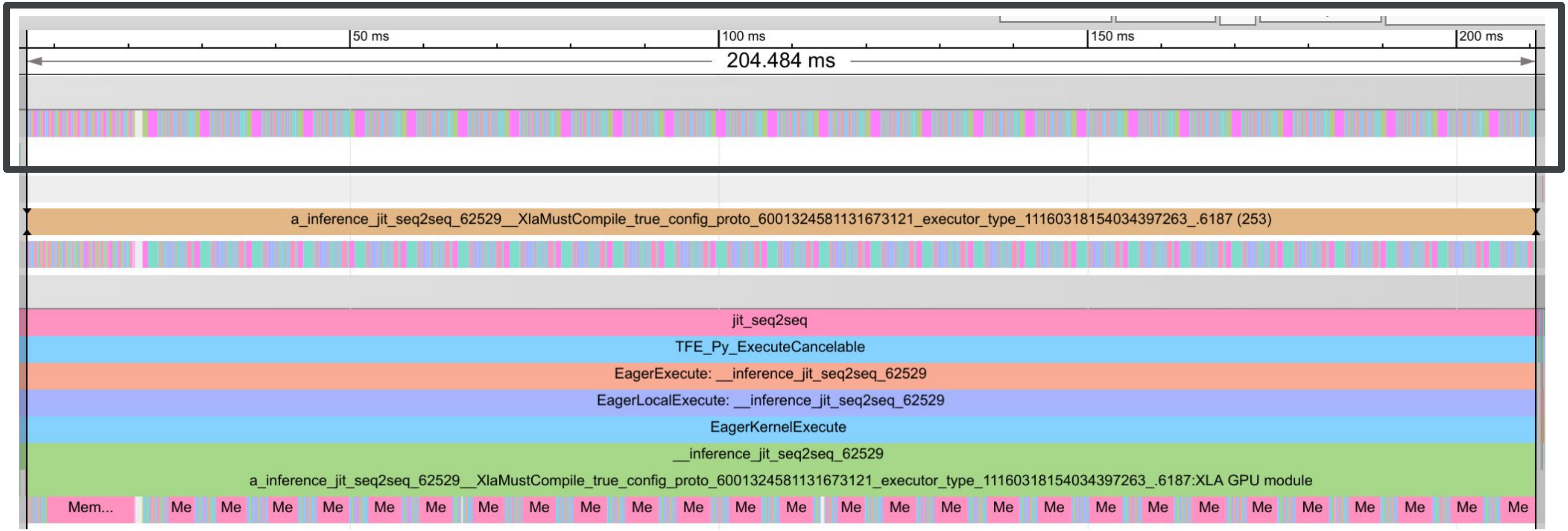
Input to correct (23 tokens):

i was walking through the park when struck by bicycle ... my arm hurts a little now .

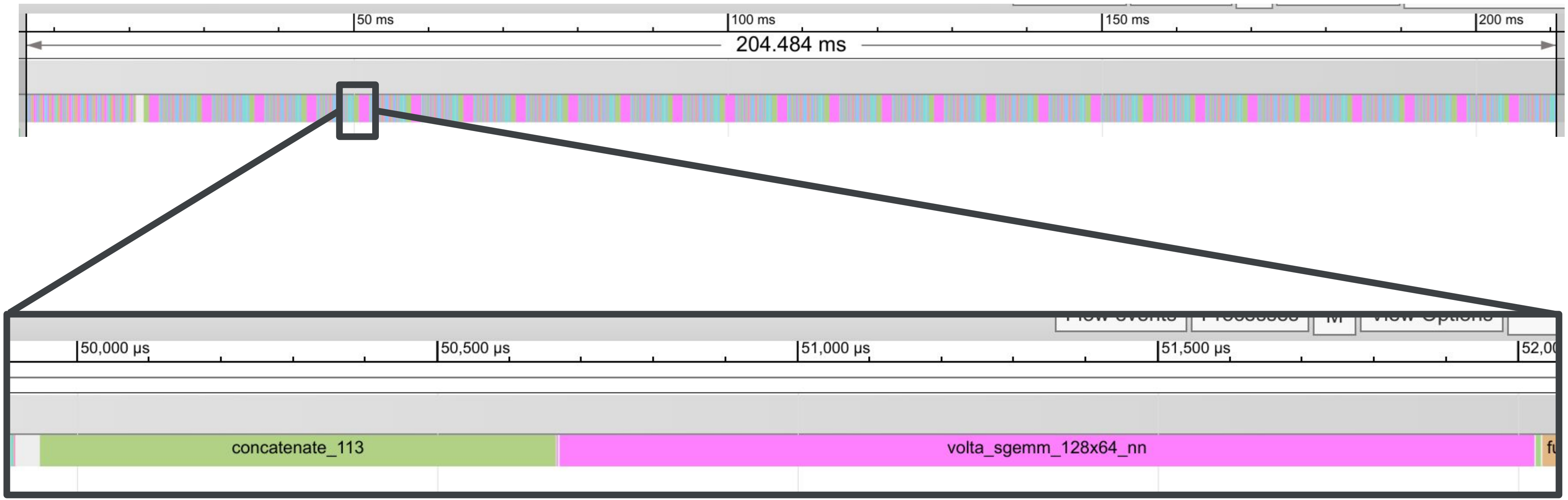
Decoder output Seq2seq (27 tokens):

`_I _was _walking _through _the _park _when _I _was _struck _by _ a _bicycle _ ... _my _arm _hurt s _ a _little _now _ . </s>`

Seq2Seq

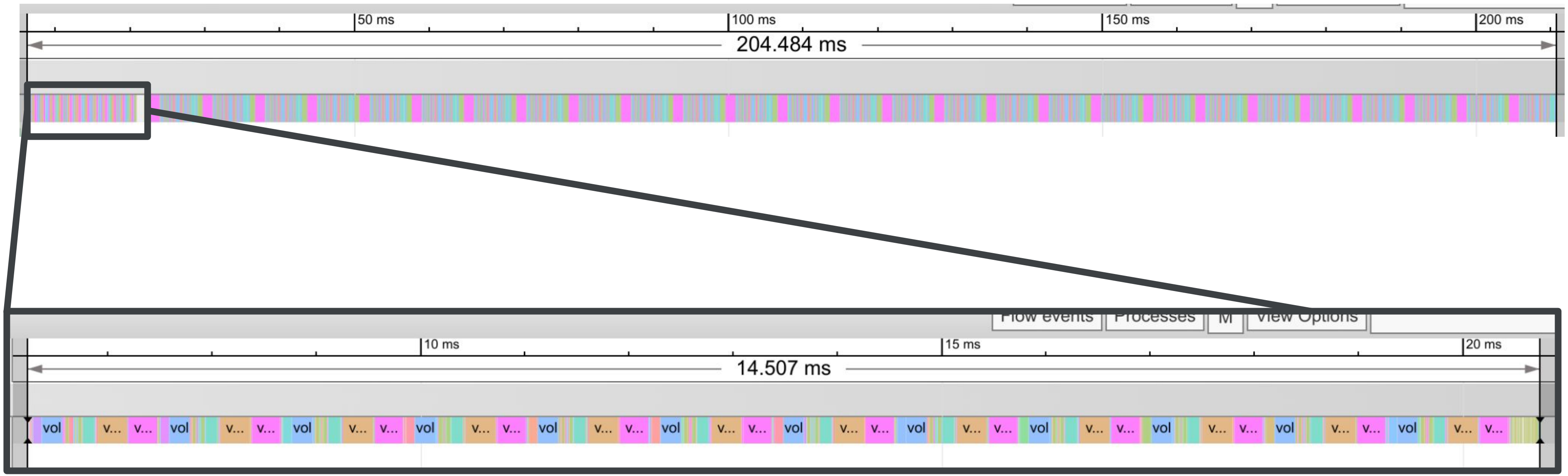

50 ms
100 ms
150 ms
200 ms
204.484 ms
a_inference_jit_seq2seq_62529__XlaMustCompile_true_config_proto_6001324581131673121_executor_type_11160318154034397263_.6187 (253)
jit_seq2seq
TFE_Py_ExecuteCancelable
EagerExecute: __inference_jit_seq2seq_62529
EagerLocalExecute: __inference_jit_seq2seq_62529
EagerKernelExecute
__inference_jit_seq2seq_62529
a_inference_jit_seq2seq_62529__XlaMustCompile_true_config_proto_6001324581131673121_executor_type_11160318154034397263_.6187:XLA GPU module
Mem...

Seq2Seq



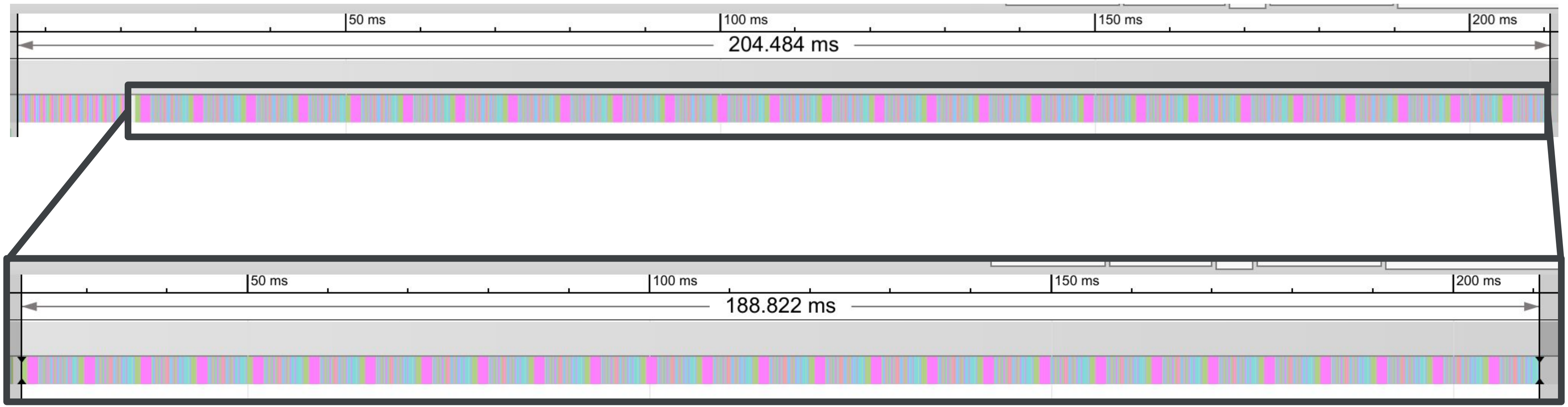
- End-to-end latency: 204ms
- Compiled with [XLA](XLA)
- Disabling compilation will increase latency, but make the profile more readable

Seq2Seq, encoder



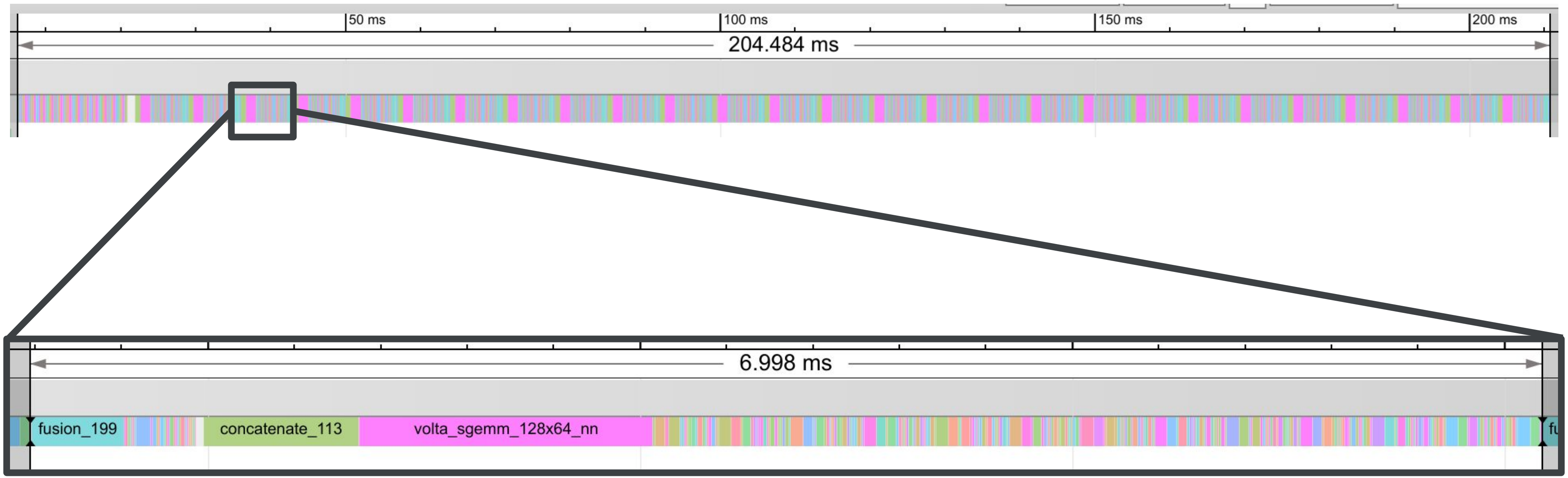
- Encoder takes 15ms

Seq2Seq, decoder



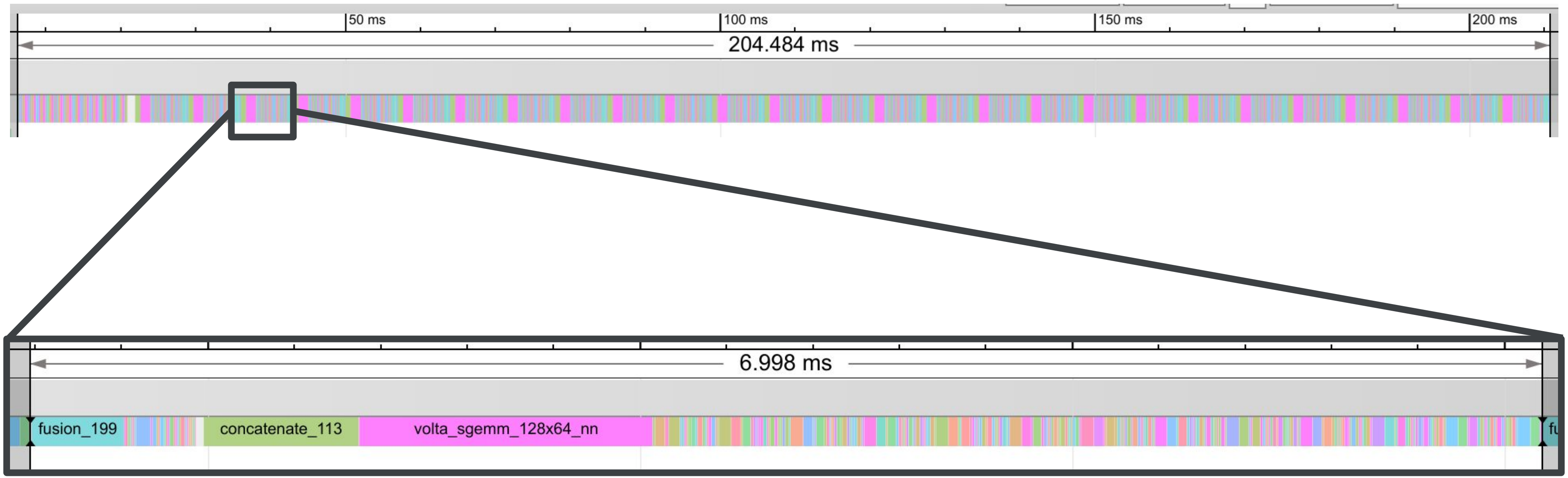
- Encoder takes 15ms
- Decoder takes 189ms

Seq2Seq, decoder step



- Encoder takes 15ms
- Decoder takes 189ms
- Single decoder step takes 7ms
  - 7 [ms/step] * 27 [steps] = 189ms

Seq2Seq, conclusions

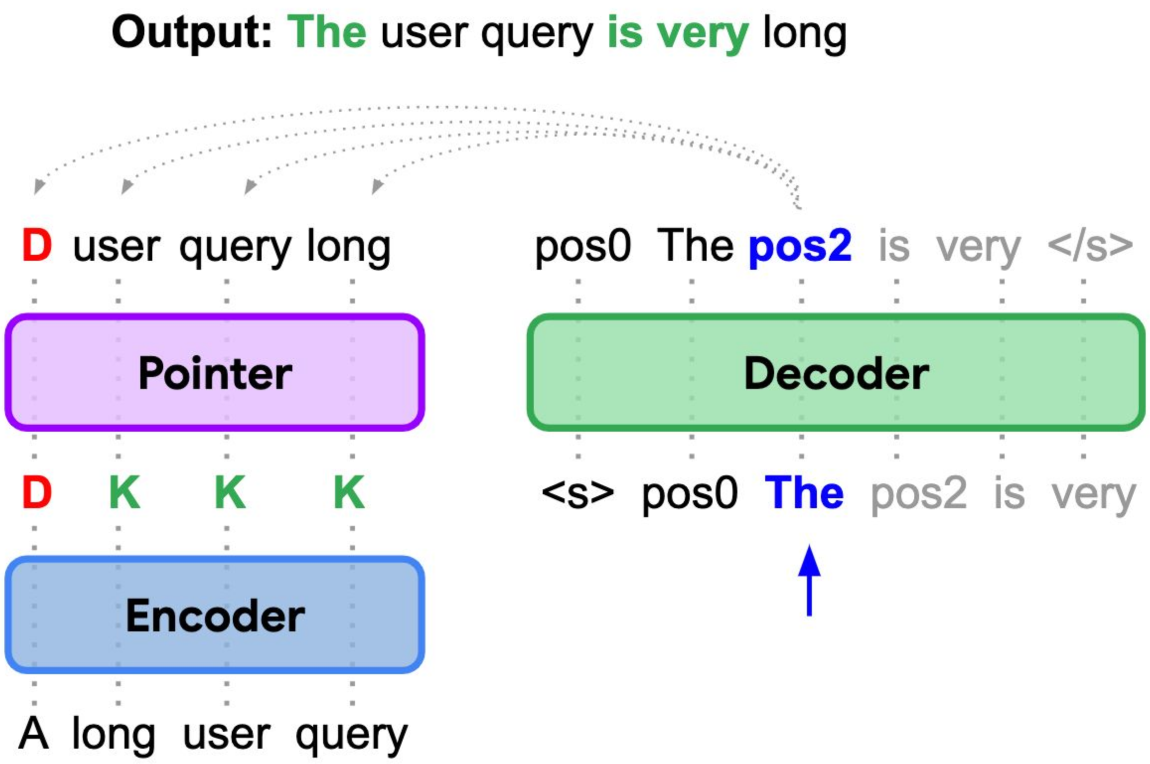


- Encoder takes 15ms
- Decoder takes 189ms
- Single decoder step takes 7ms
  - 7 [ms/step] * 27 [steps] = 189ms

If we want to reduce latency, target the decoder:

- Reduce the number of steps.
- Reduce the latency per step.

Refresher on EdiT5



Source: EdiT5 paper (Mallinson et al. 2022).

How does EdiT5 reduce latency?

- Use 1-layer decoder
    - Isn’t limited to text-editing models
- It moves work into the encoder
    - Tagging, Reordering
- Limit use of autoregressive decoder

GEC

Input to correct (21 tokens):

i was walking through the park when struck by bicycle... my arm hurts a little now.

Decoder output Seq2seq (27 tokens):

_I _was _walking _through _the _park _when **_I _was** _struck _by **_ a** _bicycle _ ... _my _arm _hurt s _ a _little _now _ . </s>

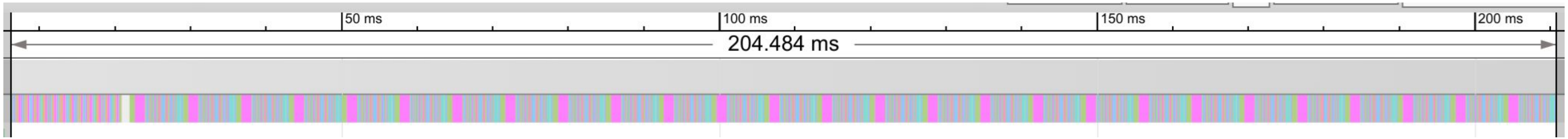
Decoder output EdiT5 (10 tokens)

<extra_id_1> **_I _was** <extra_id_6> **_I _was** <extra_id_8> **_ a** </s>

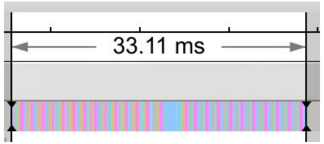
Note: extra ids are used to represent insertion positions.

EdiT5 vs Seq2Seq

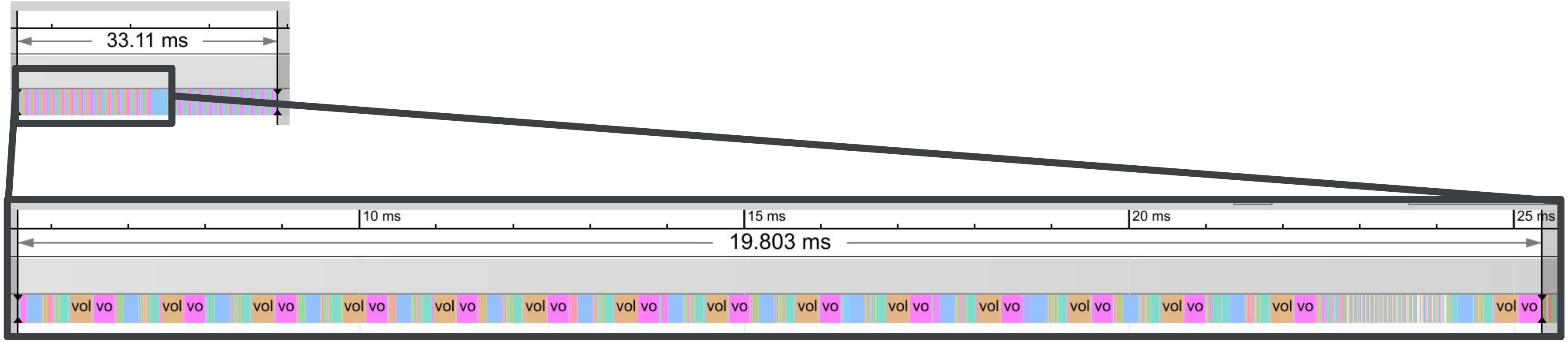
Seq2seq model:



EdiT5 model:

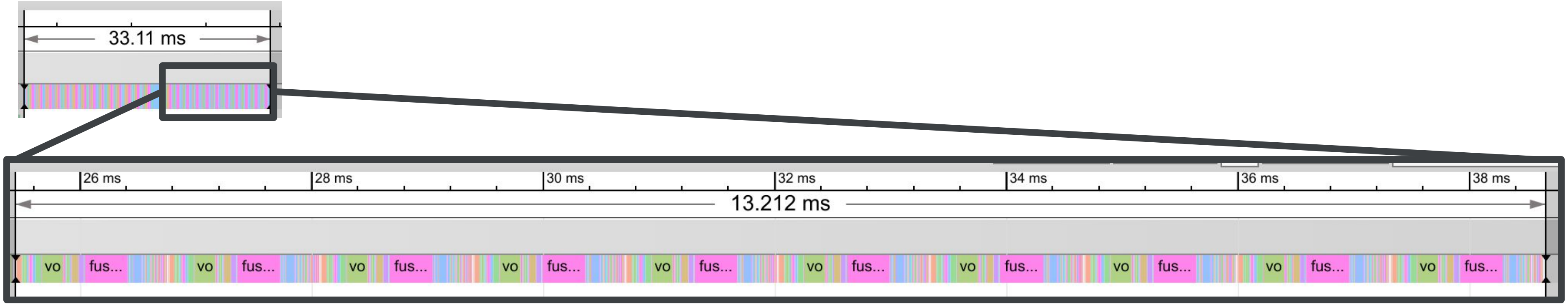


EdiT5 encoder and overhead



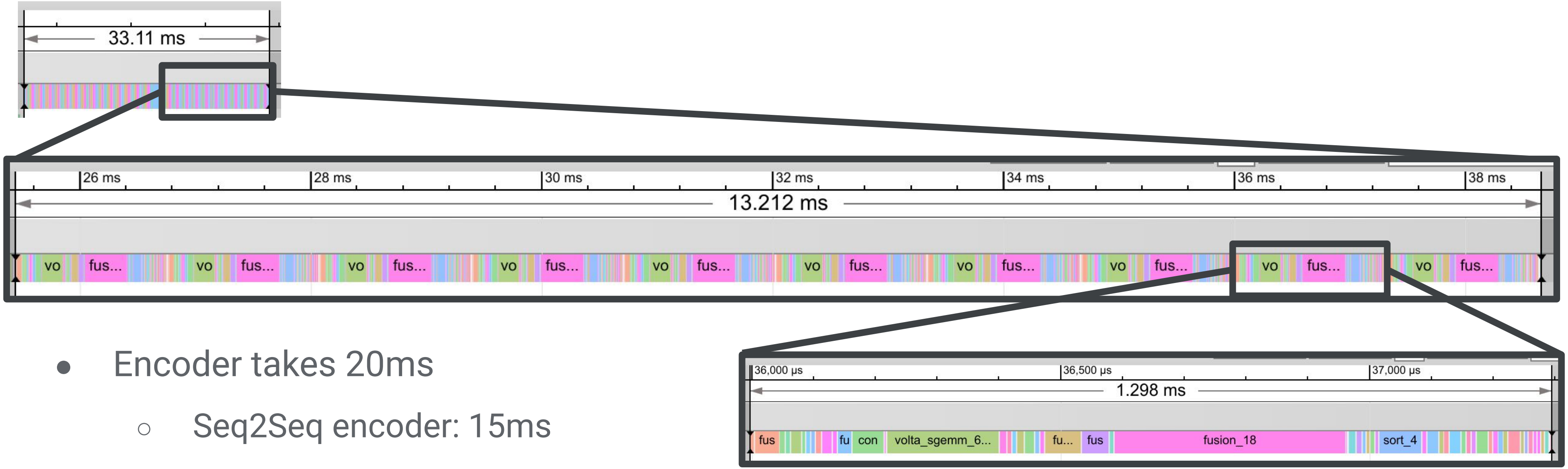
- Encoder takes 20ms
    - Seq2Seq encoder: 15ms
    - Pointing mechanism, extra layers

EdiT5 decoder



- Encoder takes 20ms
  - Seq2Seq encoder: 15ms
  - Pointing mechanism, extra layers
- Decoder takes 13ms

EdiT5 decoder step



- Encoder takes 20ms
  - Seq2Seq encoder: 15ms
  - Pointing mechanism, extra layers
- Decoder takes 13ms
  - Single step takes 1.3ms
  - Seq2Seq single step: 7ms

How does EdiT5 reduce latency?

- Decoder step takes 1.3ms compared to 7ms
  - **5.4x** reduction
- There are 10 decoder steps, compared to 27
  - Another **2.7x** reduction

In summary: **14.5x** reduction in decoder latency compared to Seq2Seq,

in exchange for **5ms** of overhead.

Text editing for latency reduction

Strategies:

- Parallel decoding
  - LaserTagger
- Iterative parallel decoding
  - GECToR, PIE, Levenshtein Transformer
- Semi-autoregression
  - Few-step decoder
    - Seq2Edits, EdiT5, others (e.g. [Chen et al., EMNLP 2020](#))
  - Combine with iterative decoding: Seq2Edits
- Pointing network for reordering: Felix, EdiT5

Iteration #	**P**	**R**	$\mathbf{F_{0.5}}$	**# corr.**
Iteration 1	72.3	38.6	61.5	787
Iteration 2	73.7	41.1	63.6	934
Iteration 3	74.0	41.5	64.0	956
Iteration 4	73.9	41.5	64.0	958

Table 4: Cumulative number of corrections and corresponding scores on CoNLL-2014 (test) w.r.t. number of iterations for our best single model.

GECToR – Grammatical Error Correction: Tag, Not Rewrite (Omelianchuk et al., BEA 2020)

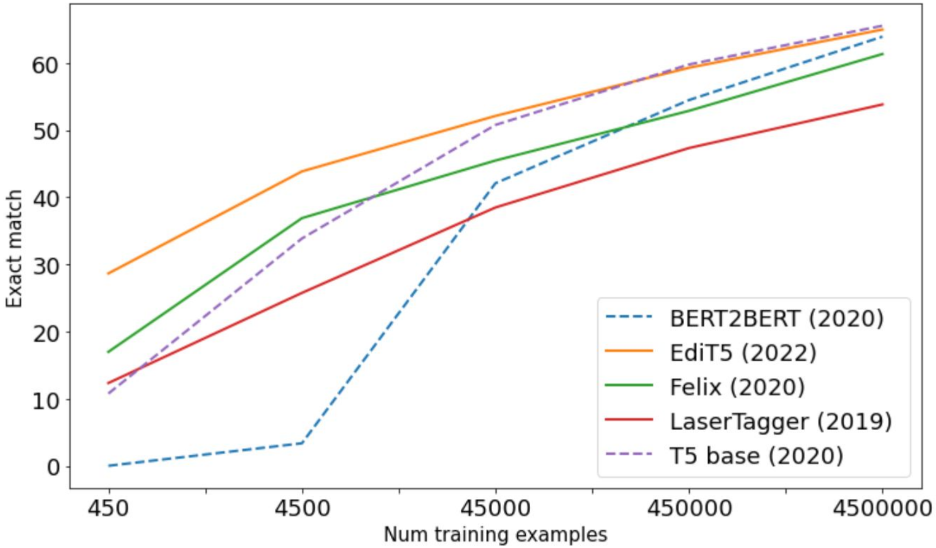
Data efficiency

Text Editing Advantages

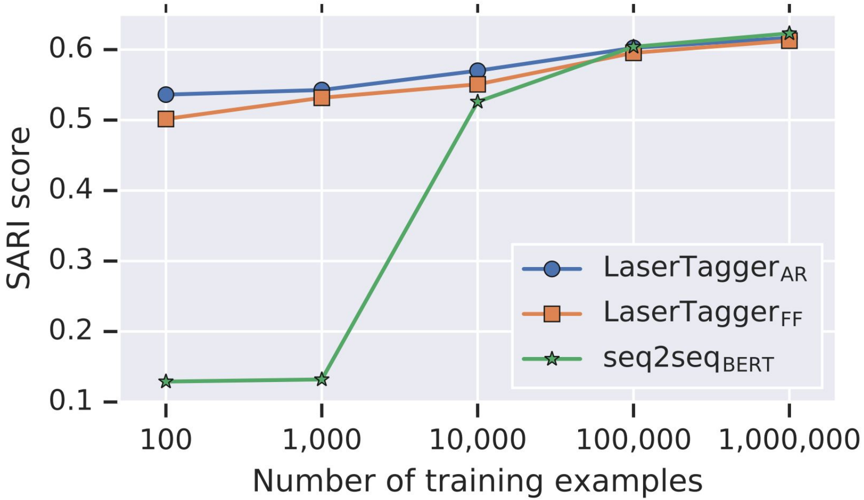
Faithfulness	Constraining decoders in seq2seq is an active area of research
Control	We can control the word a model can add / remove. Can incorporate external knowledge (e.g., pronoun).
Latency	Can be >10x faster inference.
Data efficient	Text Editing models need less training data.

Text editing for low resource settings

Sentence Fusion



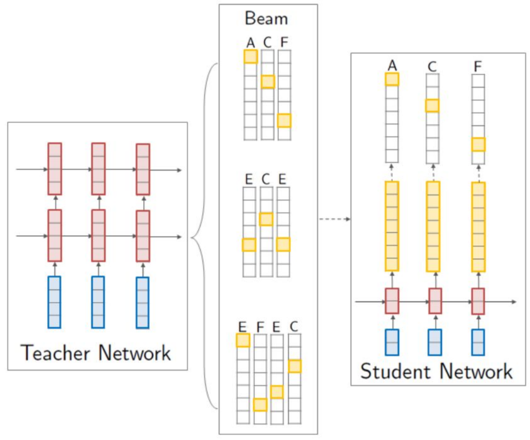
Sentence Splitting



[Encode, Tag, Realize: High-Precision Text Editing (Malmi et al., EMNLP 2019)]

Text editing models as distillation targets

- From seq2seq: Levenshtein Transformer
  - Text editing model to replicate oracle edits
  - Using seq2seq model instead of oracle improves scores
- From ensembles of text editing models: GECToR
  - Ensembles of GECToR models
  - Tarnavskyi et al., ACL 2022



Sequence-Level Knowledge Distillation

Sequence-Level Knowledge Distillation (Kim & Rush, EMNLP 2016)

Questions?