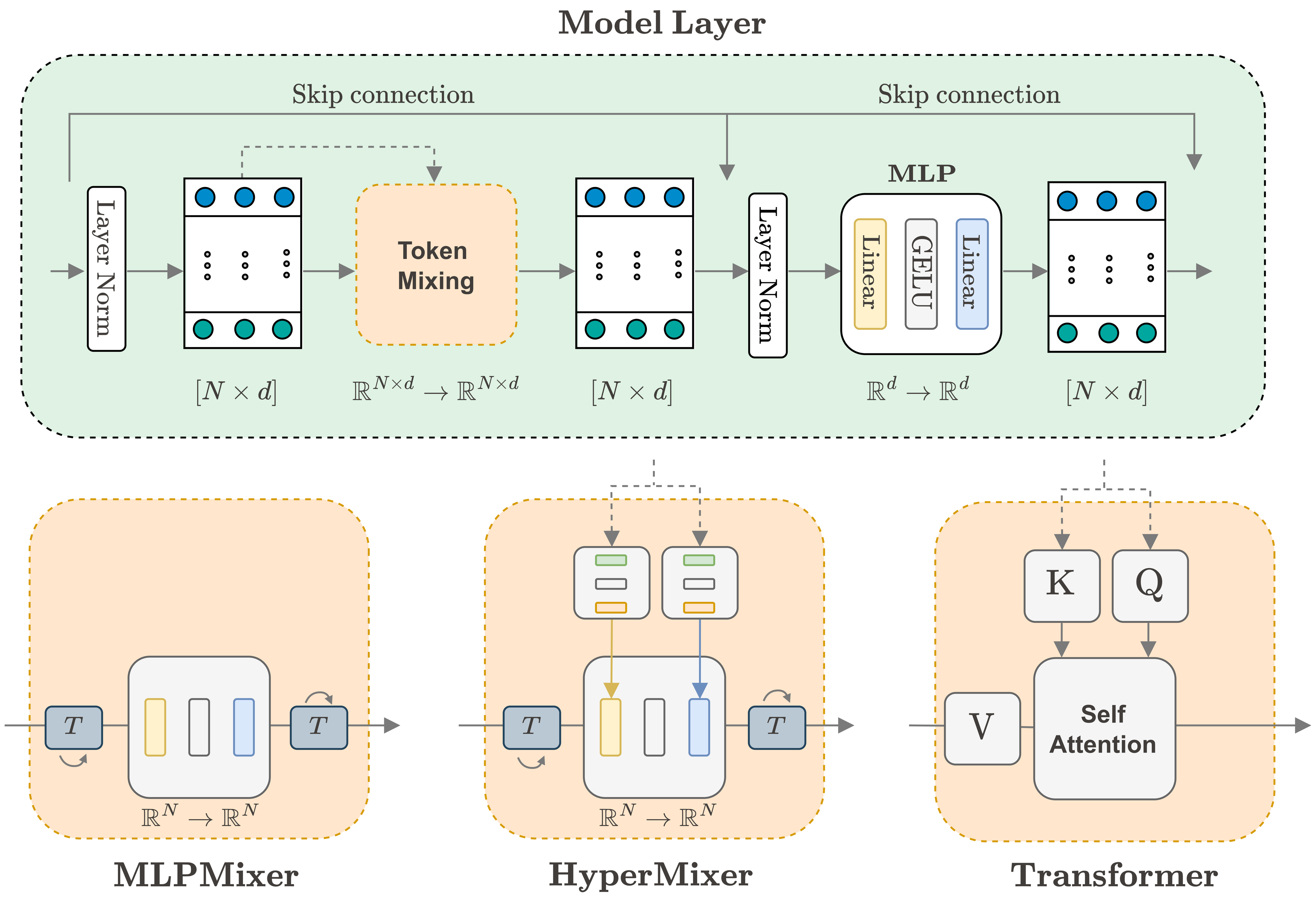
HyperMixer: An MLP-based Low Cost Alternative to Transformers

Florian Mai[1,2,4], Arnaud Pannatier[1,2], Fabio Fehr[1,2], Haolin Chen[1,2], François Marelli[1,2], François Fleuret[3,2,1], James Henderson[1]

[1]Idiap Research Institute [2]École Polytechnique Fédérale de Lausanne [3]Université de Genève [4]now at KU Leuven: florian.mai@kuleuven.be




```
class HyperMixerTokenMixing(nn.Module):
    def __init__(self, d, d_ff):

        self.hypernet_in = MLP([d, d, d_ff])
        self.hypernet_out = MLP([d, d, d_ff])
        self.pe = PositionalEncoder(d)
        self.ln = LayerNorm(d, dim=-1)

    def forward(self, queries, keys, values):
        # queries : [B, M, d]
        # keys / values : [B, N, d]

        # [B, N, d_ff]
        W1 = self.hypernet_in(self.pe(keys))

        # [B, M, d_ff]
        W2 = self.hypernet_out(self.pe(queries))

        # TM-MLP(x) = W_2 ( act ( W_1^T x) )
        # [B, d, N] -> [B, d, d_ff] -> [B, d, M]
        token_mixing_mlp = compose_MLP(W1, W2, act)

        values = values.transpose(1, 2) # [B, d, N]

        output = token_mixing_mlp(values) # [B, d, M]

        output = output.transpose(1,2) # [B, M, d]
        return self.ln(output)
```

1. Summary

- Conceptually simpler models like MLPs promise to be more sustainable because they are easier to train and require less data.
- We propose **HyperMixer**, an MLP-based neural architecture with inductive biases suited for natural language processing.
- HyperMixer is substantially better at text classification tasks than alternative MLP-based models.
- HyperMixer is less costly than Transformers in terms of processing time, training data, and hyperparameter tuning.

2. Motivation

- Simpler models promise to be less costly ⇒ MLPs!
- Existing models lack important inductive biases of Transformers: *variable binding*, *variable length* and *pos. invariance*.

	Variable binding	Pos. invariance	Variable-length
Transformer [7]	✓	✓	✓
	MLP-based models		
MLPMixer [6]	✓	✗	✗
gMLP [4]	✓	✗	✗
HyperMixer (ours)	✓	✓	✓

3. Model

See figure and pseudo-code at the top!

- General Transformer-like architecture: apply token mixing and feature mixing (FF-MLP) per token ⇒ variable binding
- MLPMixer: uses a *fixed* token mixing MLP to mix positions ⇒ fixed length and not position invariant
- HyperMixer: generate token mixing MLP with hypernetworks [2] ⇒ variable length, position invariance!

Code:

4. Experiments

Results:

1. HyperMixer performs better at text classification tasks than MLPMixer and similar MLP-based alternatives.
2. HyperMixer is less costly than Transformers in terms of processing time, training data, and hyperparameter tuning, while achieving competitive results.

Scope of results:

- Low-resource scenario: relatively small models, no pretraining, medium-size datasets
- We only cover text classification datasets (no text generation) mostly from the GLUE benchmark

4.1. Comparison to other models

Test set results on 5 tasks from the GLUE benchmark [8]:

Model	MNLI	SNLI	QQP	QNLI	SST	# Params
FNet [9]	59.8	75.3	78.4	59.6	80.0	9.5 M
Lin. Transformer [3]	**66.9**	**83.0**	82.3	61.7	80.8	11 M
Transformer [7]	65.8	80.7	82.4	73.2	79.4	11 M
MLPMixer [6]	62.9	80.1	83.5	70.5	81.2	11 M
gMLP [4]	61.2	80.9	82.5	60.2	79.5	11 M
HyperMixer (ours)	<u>66.1</u>	<u>81.7</u>	**<u>84.1</u>**	**<u>77.1</u>**	**<u>81.4</u>**	11 M

<u>underlined:</u> best MLP-based model. **bold:** best model overall.

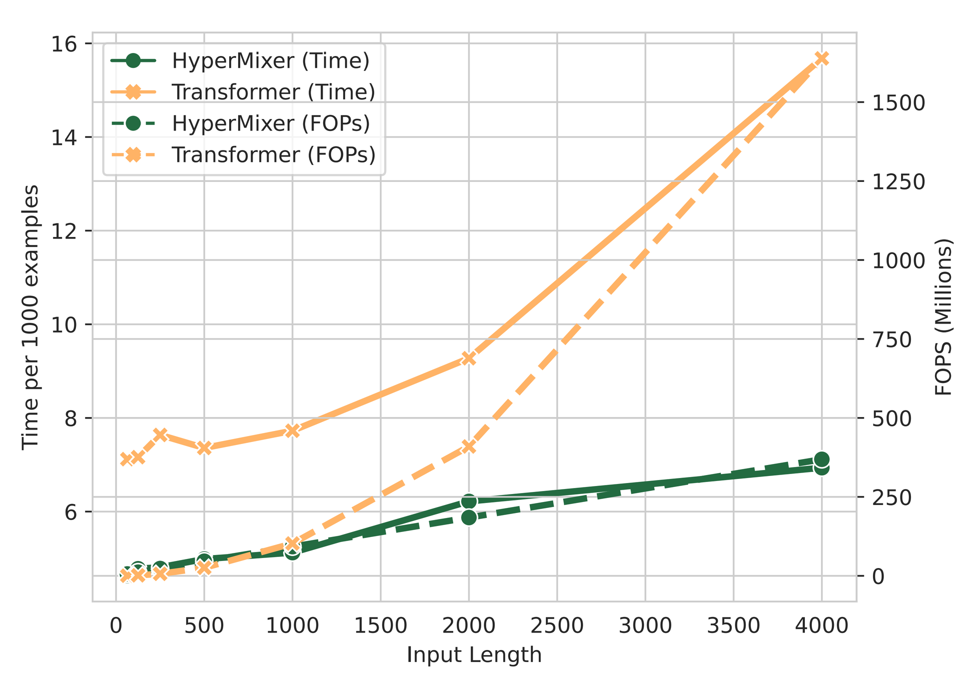
4.2. Cost comparison with Transformers

Cost of an AI result according to Schwartz et al. [5]:

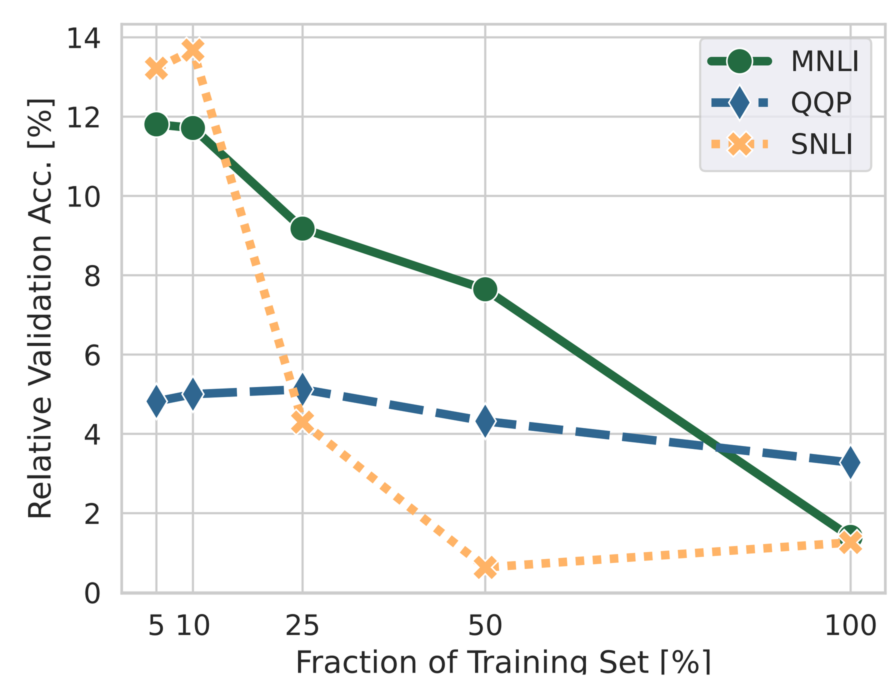
$$Cost(R) \propto E \cdot D \cdot H$$

E : processing time; D : dataset size; H : hyperparameters

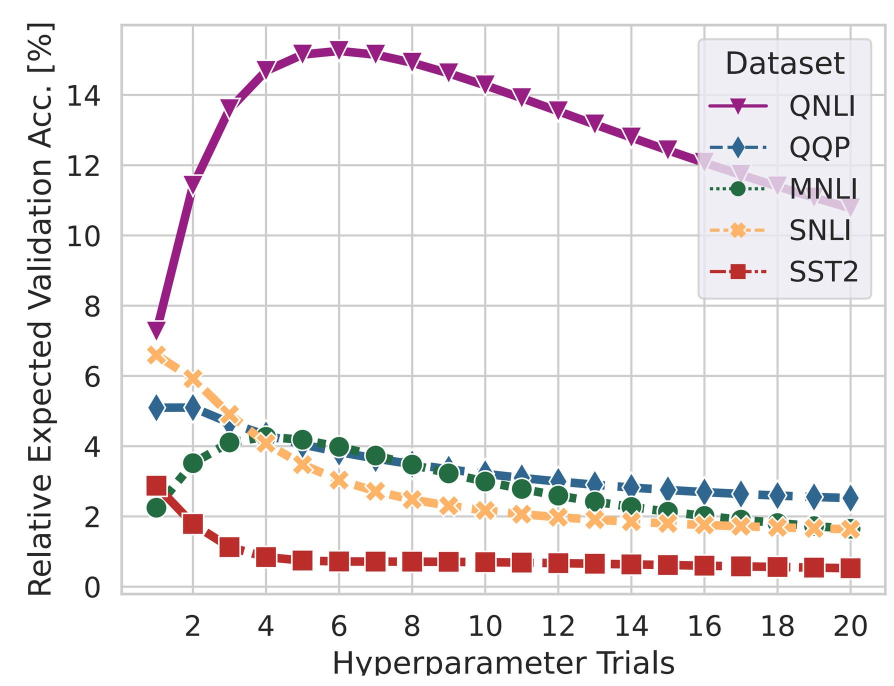
E**: HyperMixer has complexity of $\mathcal{O}(N)$ vs Transformers $\mathcal{O}(N^2)$**



D**: HyperMixer does better in the low-resource scenario** (graph shows HyperMixer's relative improvement over Transformers as a function of training data size)



H**: HyperMixer does better with small hyperparameter tuning** (graph shows HyperMixer's expected relative improvement[1] over Transformers as function of #trials in random hyperparameter search)



References

[1] Dodge et al. Show your work: Improved reporting of experimental results. In *EMNLP*, 2019.

[2] Ha et al. Hypernetworks. *arXiv*, 2016.

[3] Katharopoulos et al. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020.

[4] Liu et al. Pay attention to mlps. In *NeurIPS*, 2021.

[5] Schwartz et al. Green ai. *Communications of the ACM*, 2020.

[6] Tolstikhin et al. Mlp-mixer: An all-mlp architecture for vision. *NeurIPS*, 2021.

[7] Vaswani et al. Attention is all you need. *NeurIPS*, 2017.

[8] Wang et al. Glue: A multi-task benchmark and analysis platform for natural language understanding. *ICLR*, 2019.

[9] Yu et al. Metaformer is actually what you need for vision, 2021.

Acknowledgement

Florian Mai was supported by the Swiss National Science Foundation under grant number 200021_178862. Arnaud Pannatier was supported by the Swiss Innovation Agency Innosuisse under the project MALAT, grant number “32432.1 IP-ICT”. Fabio Fehr was supported by the Swiss National Centre of Competence in Research (NCCR) under the project Evolving Language, grant number “51NF40_180888”. Haolin Chen was supported by the Swiss National Science Foundation under the project NAST, grant number “185010”. François Marelli was supported by the Swiss National Science Foundation under the project COMPBIO, grant number “179217”.