HYPERGRID: Efficient Multi-Task Transformers with Grid-wise Decomposable Hyper Projections

Yi Tay, Zhe Zhao, Dara Bahri, Donald Metzler, Da-Cheng Juan
Google Research

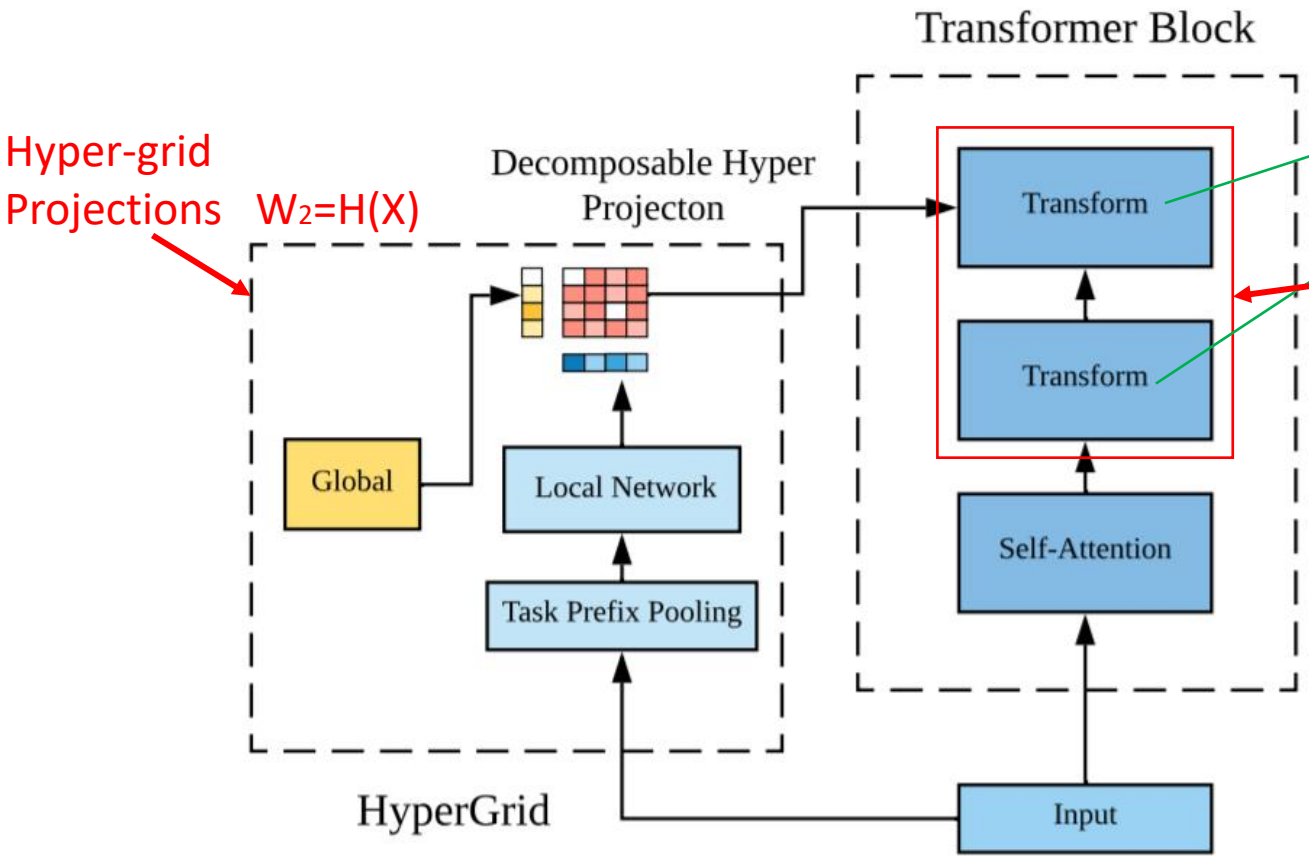
(Under review)

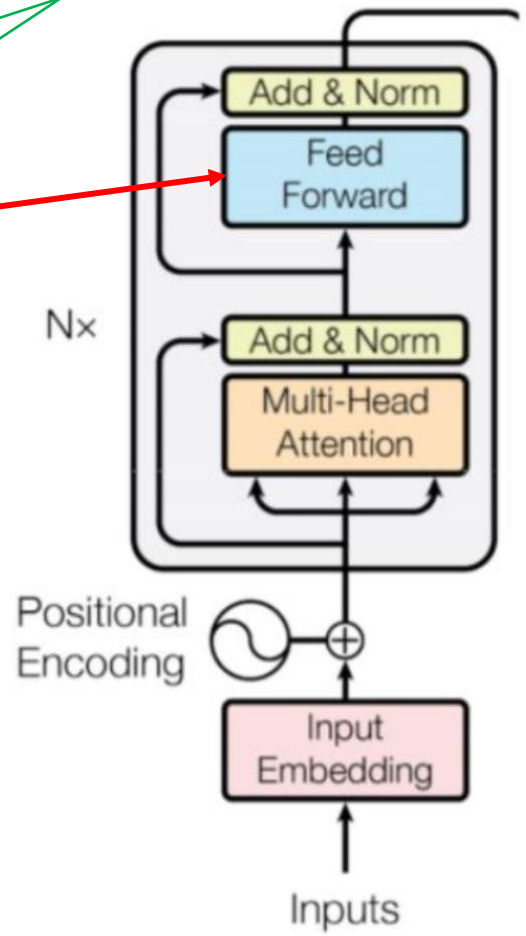
Overview

- This paper introduces a **hypernetwork**-based method for **generating weights** of a pretrained transformer (T5) to improve the **multi-task learning** on the GLUE and SuperGLUE benchmarks.
- The weights of the transformer are generated based on the pretrained weights and **each example** the model receives.
- The paper proposes grid-wise projection as a method for controlling the weight generating procedure.
- The proposed method achieves competitive results with regular methods which finetune separate models for each task of the GLUE and SuperGLUE benchmarks.

Overall Architecture

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$





Hyper-grid Projections

- HYPERGRID operates on weight matrices (linear transformations), i.e., Y = WX + b.
- Instead of letting W be free weights, W are generated using a parameterized side network H().

$$Y = \boldsymbol{W}x + b \quad \text{where} \quad \boldsymbol{W} = H(X)$$

Where $\boldsymbol{W} \in \mathbb{R}^{d_m \times d_f}$ and $X \in \mathbb{R}^{d_m}$ (prefix token of the input sequence).

- **Local only (L):**

$$H(X) = \sigma(\boldsymbol{U}X)\mathbf{1}^\top \odot \boldsymbol{W}$$

$$U \in \mathbb{R}^{d_m \times d_f}$$

Hyper-grid Projections

- HYPERGRID operates on weight matrices (linear transformations), i.e., Y = WX + b.
- Instead of letting W be free weights, W are generated using a parameterized side network H().

$$Y = \boldsymbol{W}x + b \quad \text{where} \quad \boldsymbol{W} = H(X)$$

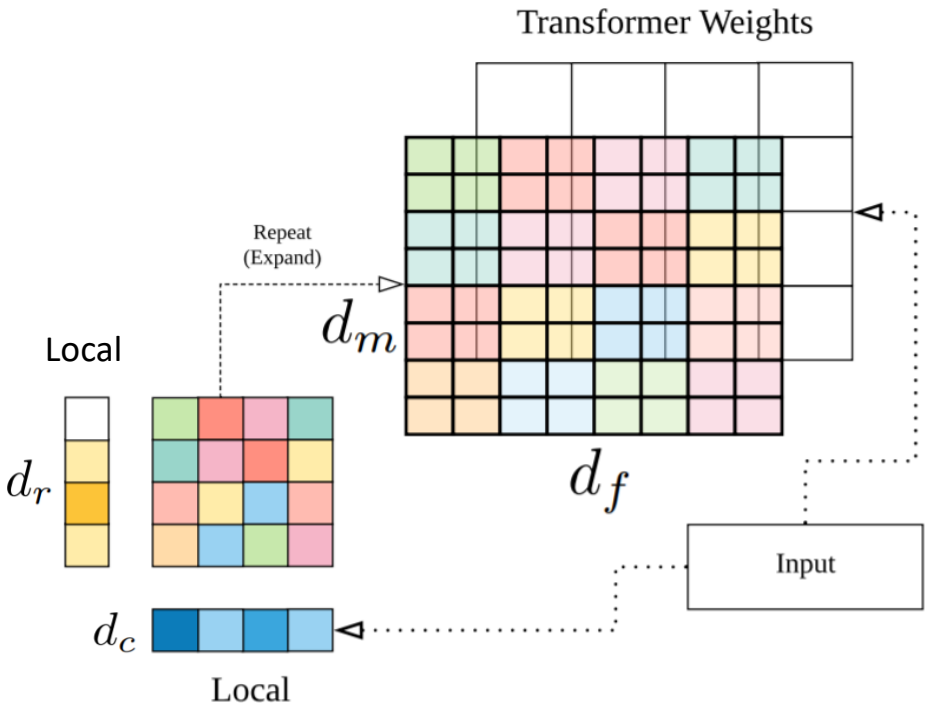
Where $\boldsymbol{W} \in \mathbb{R}^{d_m \times d_f}$ and $X \in \mathbb{R}^{d_m}$.

- **Local-Local (L2):**

$$H(X) = \psi(\sigma((\boldsymbol{L_r}X)(\boldsymbol{L_c}X)^\top)) \odot \boldsymbol{W}$$

Where: $(\boldsymbol{L_r}X)(\boldsymbol{L_c}X)^\top \in \mathbb{R}^{d_r \times d_c}$

$\psi(.)$ is a repeat function



Hyper-grid Projections

- HYPERGRID operates on weight matrices (linear transformations), i.e., Y = WX + b.
- Instead of letting W be free weights, W are generated using a parameterized side network H().

$$Y = \boldsymbol{W}x + b \quad \text{where} \quad \boldsymbol{W} = H(X)$$

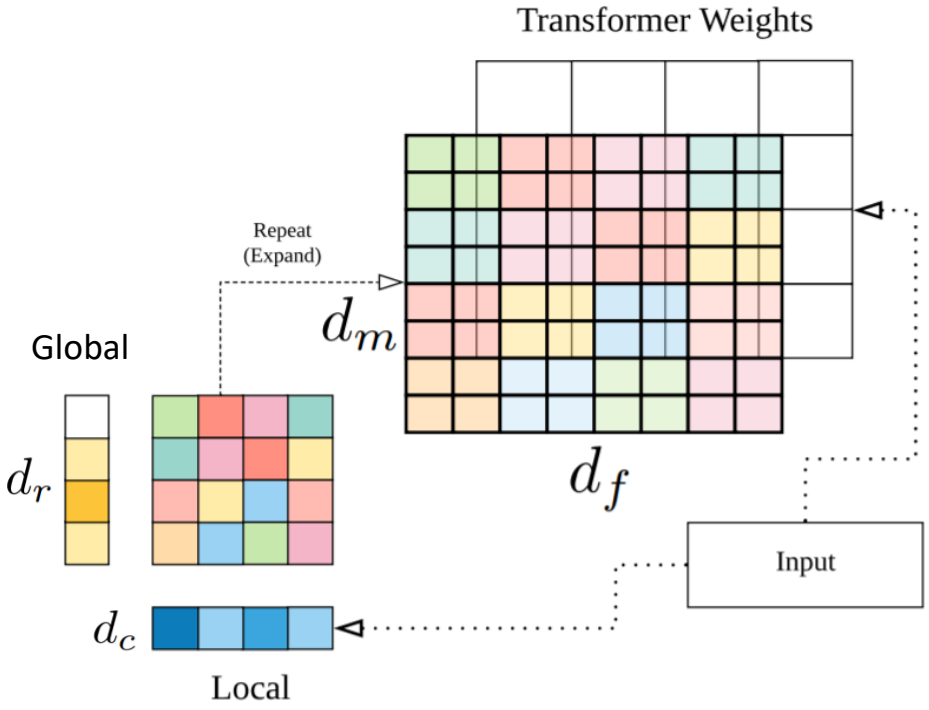
Where $\boldsymbol{W} \in \mathbb{R}^{d_m \times d_f}$ and $X \in \mathbb{R}^{d_m}$.

- **Local-Global (LG):**

$$H(X) = \psi(\sigma((\boldsymbol{L_r}X)\boldsymbol{G_c}^{\top})) \odot \boldsymbol{W}$$

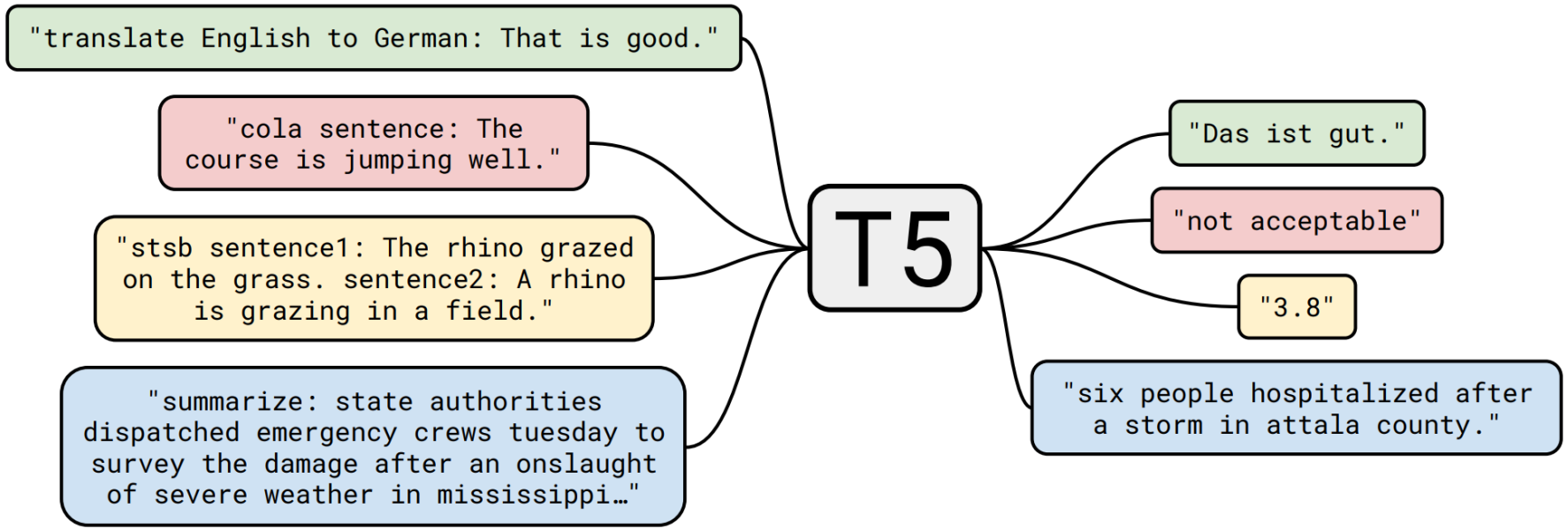
Where: $\boldsymbol{G_c} \in \mathbb{R}^{d_f}$

$\psi(.)$ is a repeat function



Seq2seq-based multi-task learning

- Formulate GLUE and SuperGLUE tasks in the form of predicting the target sequence given the input sequence:



Results

Model	$\|\theta\|$	Avg	CoLA	SST	MR	STS	QQP	MNLI	QNLI	RTE	WNLI
BERT*	-	80.5	60.5	94.9	84.5	86.5	89.3	86.7	92.7	70.1	65.1
RoBERTa*	-	88.1	67.8	96.7	89.8	91.9	90.2	90.8	95.4	88.2	89.0
ALBERT*	-	-	69.1	97.1	91.2	92.0	90.5	91.3	-	89.2	89.0
XLNet*	-	-	70.2	97.1	90.5	92.6	90.4	90.9	-	88.5	89.1
ELECTRA*	5B	89.4	71.7	97.1	90.7	92.5	90.8	91.3	95.8	88.5	92.5
T5 (3B)	48B	88.5	67.1	97.4	90.0	89.8	82.1	91.3	96.3	91.1	89.7
T5 (11B)	176B	89.7	70.8	97.1	90.0	92.1	82.5	90.9	96.7	92.5	93.2
Ours (3B)	3B	88.2	65.6	97.5	89.0	91.6	81.9	90.9	95.9	90.1	89.7
Ours (11B)	11B	89.4	69.0	97.6	89.2	92.6	82.0	91.3	96.4	91.5	93.2

Table 5: Test set performance on GLUE [Wang et al., 2018]. Models with $*$ are large ensembles. All models are single-tasked fine-tuned except ours. Parameter costs are reported considering ensembles and cost required to fit all of GLUE and SuperGLUE.

Model	$\|\theta\|$	Avg	BQ	CB	CP	MultiRC	Record	RTE	WiC	WSC
BERT++	2.7B	71.5	79.0	84.8/90.4	73.8	70.0/24.1	72.0/71.3	79.0	69.6	64.4
RoBERTa	56B	84.6	87.1	90.5/95.2	90.6	84.5/52.5	90.6/90.0	88.2	69.9	89.0
T5 (3B)	48B	86.4	89.9	90.3/94.4	92.0	86.8/58.3	91.2/90.4	90.7	72.1	90.4
T5 (11B)	176B	88.9	91.0	93.0/96.4	94.8	88.2/62.3	93.3/92.5	92.5	76.1	93.8
Ours (3B)	3B	84.7	89.2	81.7/90.4	89.6	86.6/58.7	91.1/90.3	90.8	70.6	87.7
Ours (11B)	11B	87.7	90.7	85.5/92.0	94.0	87.9/61.7	93.3/92.6	91.5	74.6	92.1

Table 6: Test set performance on SuperGLUE [Wang et al., 2019]. Our MTL approach achieves competitive performance to the state-of-the-art with a single multi-task model. Parameter costs refers to total number of parameters used to fit all GLUE and SuperGLUE tasks