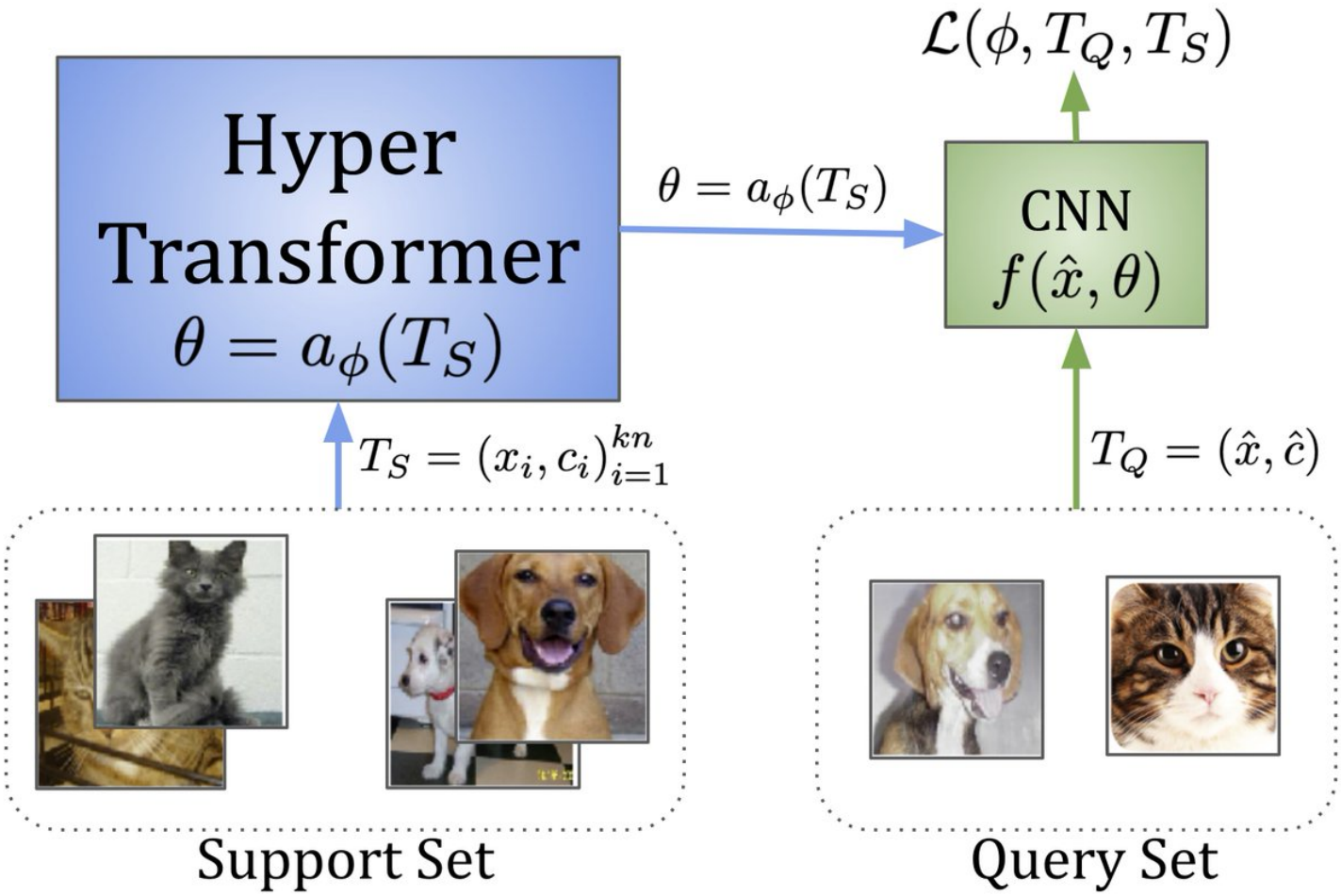
Twitter Thread by Max Vladymyrov

Max Vladymyrov
@mvladymyrov

I'm excited to share our new paper on HyperTransformers, a novel architecture for few-shot learning able to generate the weights of a CNN directly from a given support set. ■■

■: https://t.co/vcm67G6P6t with Andrey Zhmoginov and Mark Sandler.



2) We train a transformer model to `convert` a few-shot task description into a small CNN network specialized in solving it on new images.

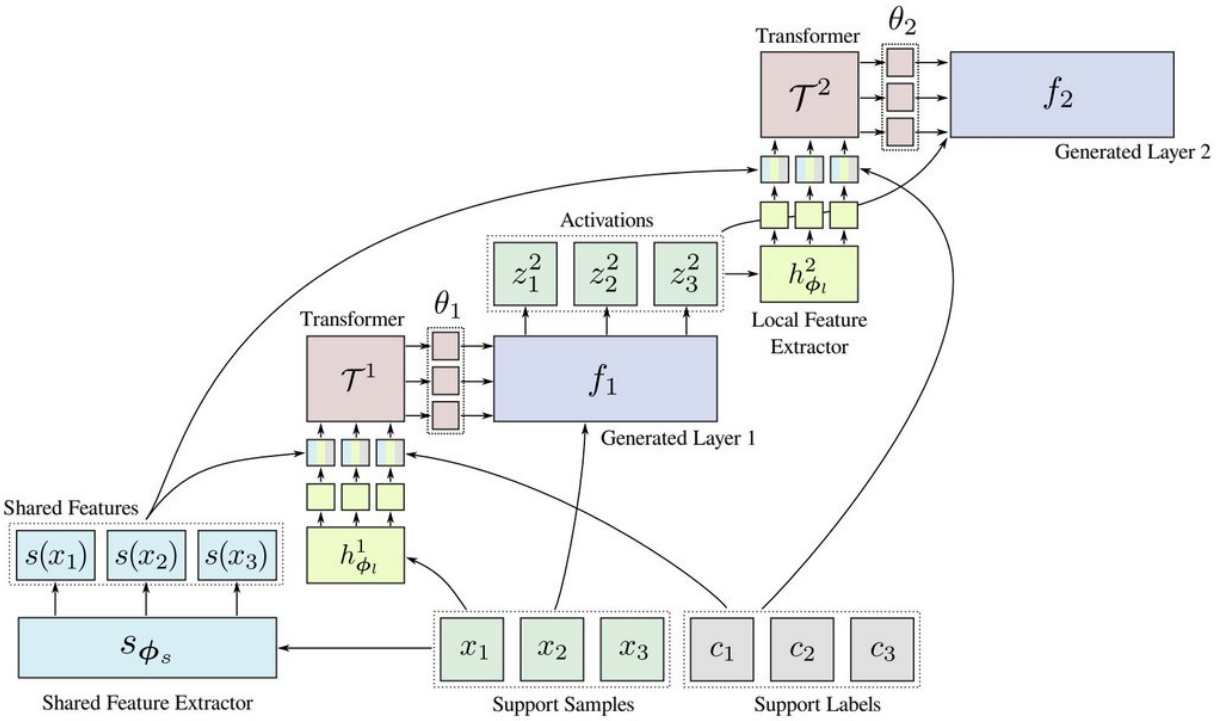


Figure 1: A diagram of our model showing the generation of two CNN layers: transformer-based weight generators receive global shared features $s_{\phi_s}(\cdot)$ and local features $h_{\phi_l}(\cdot)$ along with corresponding labels c_i, and produce CNN layer weights (θ_1 and θ_2). After being generated, the CNN model is used to compute the loss on the query set. The gradients of this loss are then used to adjust the weights of the entire weight generation model (ϕ_s, ϕ_l, transformer weights).

3) This effectively decouples a high-capacity transformer generator from a much smaller inference model. It is different from most of the existing methods, e.g. MAML where the generator and the executing model share the same architecture.

4) CNN weights are generated layer-by-layer from a combination of layer embedding (features from the last generated layer), and image w/ class embeddings (features directly from the data). The final weights are extracted from output of self-attention (similar to [CLS] tokens).

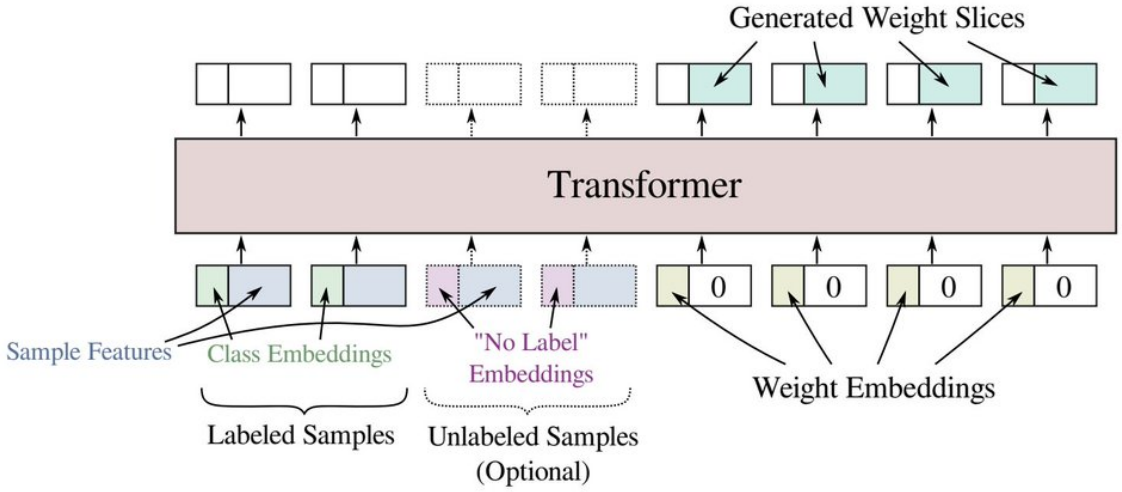


Figure 2: Structure of the tokens passed to and received from a transformer model. Both labeled and unlabeled (optional) samples can be passed to the transformer as inputs. Empty output tokens indicate ignored outputs.

5) What is cool is that we can also add unlabeled samples from the support set into the mix, effectively allowing for semi-supervised few-shot learning!

6) HyperTransformers are comparable in performance to many competing methods on miniImageNet and tieredImageNet datasets.

Table 2: Comparison of MINIIMAGENET and TIEREDIMAGENET 1-shot (1-S) and 5-shot (5-S) 5-way results for HT (underlined) and other widely known methods with a 64-64-64-64 model including (Tian et al., 2020): Matching Networks (Vinyals et al., 2016), IMP (Allen et al., 2019), Prototypical Networks (Snell et al., 2017), TAML (Jamal & Qi, 2019), SAML (Hao et al., 2019), GCR (Li et al., 2019a), KTN (Peng et al., 2019), PARN (Wu et al., 2019), Predicting Parameters from Activations (Qiao et al., 2018), Relation Net (Sung et al., 2018), MELR (Fei et al., 2021). We also include results for CNNs with fewer channels ("-32" for 32-channel models, etc.).

MINIIMAGENET						TIEREDIMAGENET		
Method	**1-S**	**5-S**	**Method**	**1-S**	**5-S**	**Method**	**1-S**	**5-S**
HT	54.1	68.5	HT-48	**55.1**	68.1	HT-32	**52.7**	**69.9**
MN	43.6	55.3	SAML	52.2	66.5	MAML-32	51.7	**70.3**
IMP	49.2	64.7	GCR	53.2	**72.3**	HT	**56.1**	**73.3**
PN	49.4	68.2	KTN	54.6	71.2	PN	53.3	72.7
MELR	**55.4**	**72.3**	PARN	**55.2**	71.6	MELR	**56.4**	**73.2**
TAML	51.8	66.1	PPA	54.5	67.9	RN	54.5	71.3

7) But our method especially shines for the case of small target CNN architectures, where the large capacity of the transformer model is the most useful and noticeable. For the 8-channels model we are seeing 5-10% improvement over MAML++!

Table 1: Comparison of HT with MAML++ on models of different sizes and different datasets: (a) 20-way OMNIGLOT and (b) 5-way MINIIMAGENET. Results for MAML++ were obtained using GitHub code accompanying Antoniou et al. (2019), those marked with † are from Antoniou et al. (2019). HT outperforms MAML++ on many few-shot tasks. Accuracy confidence intervals: OMNIGLOT – between 0.1% and 0.3%, MINIIMAGENET – between 0.2% and 0.5%.

Approach	*1-shot (channels)*					*5-shot (channels)*				
	8	16	32	48	64	8	16	32	48	64
OMNIGLOT:										
- MAML++	81.4	88.6	**95.6**	**95.8**	**97.7**†	83.2	94.9	**98.6**	**98.8**	**99.3**†
- HT	**87.2**	**93.7**	**95.5**	**95.7**	96.2	**94.7**	**98.0**	**98.6**	**98.8**	98.8
MINI:										
- MAML++	43.9	46.6	49.4	52.2†	–	**59.0**	**64.6**	66.8	**68.3**†	–
- HT	**45.5**	**50.2**	**53.8**	**55.1**	–	58.5	63.8	**67.1**	**68.1**	–

8) As it turns out, for small target models, where every neuron matters, it is important to generate the whole network from a given support set. For larger target models even generating only the last logits layers appears to be sufficient.

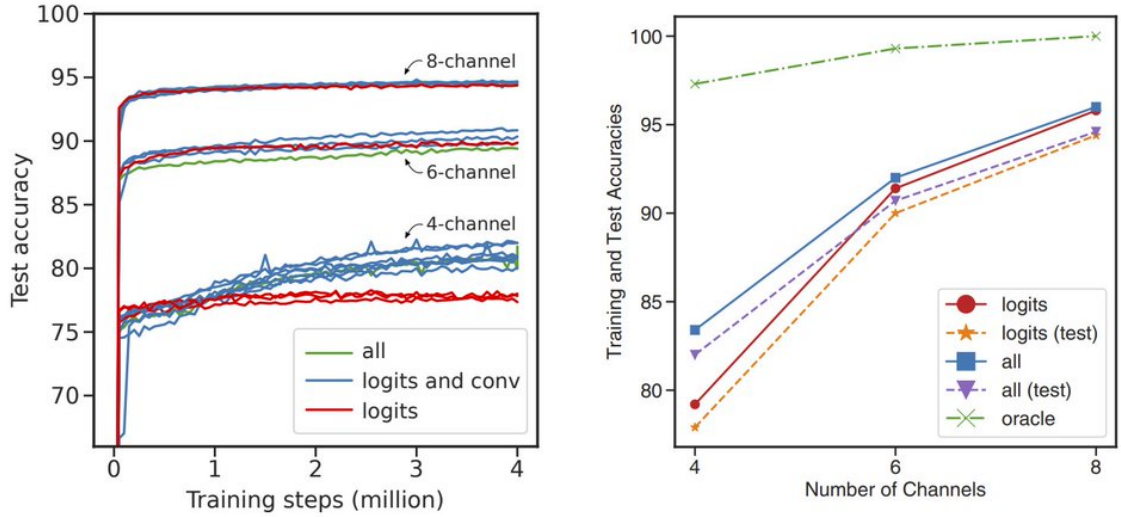


Figure 3: (**Left**) Test accuracies for the generated 4-, 6- and 8-channel CNN models on the 5-shot-20-way OMNIGLOT task. Models with only the last logits layer generated (*red*) are characterized by lower test accuracies compared to the models with some or all convolutional layers also being generated (*blue*, *green*). Similar plot for TIEREDIMAGENET can be found in Appendix (Fig. 13). (**Right**) 5-shot-20-way OMNIGLOT training/test accuracies (separate run) as a function of the CNN model complexity: only the final logits layer being generated (*logits*), all layers being generated (*all*), training the model on all available samples for a random set of few classes (*oracle*). A model that generates CNN weights by memorizing all samples (being able to determine their classes) and also memorizing optimal trained weights for any selection of classes would reach the *oracle* accuracy, but would not generalize.

9) We are really excited about the direction of using Transformers to guide the construction and performance of smaller specialized models e.g. in low-power settings. This has a lot of applications in the areas where high-performance compact personalized networks are being used.

