

# Mean Embeddings with Test-Time Data Augmentation for Ensembling of Representations

Arsenii Ashukha<sup>1</sup> Andrei Atanov<sup>2</sup> Dmitry Vetrov<sup>1</sup>

## Abstract

Averaging *predictions* over a set of models—an ensemble—is widely used to improve predictive performance and uncertainty estimation of deep learning models. At the same time, many machine learning systems, such as search, matching, and recommendation systems, heavily rely on embeddings. Unfortunately, due to misalignment of features of independently trained models, embeddings, cannot be improved with a naive deep ensemble like approach. In this work, we look at the ensembling of representations and propose mean embeddings with test-time augmentation (MeTTA) simple yet well-performing recipe for ensembling representations. Empirically we demonstrate that MeTTA significantly boosts the quality of linear evaluation on ImageNet for both supervised and self-supervised models. Even more exciting, we draw connections between MeTTA, image retrieval, and transformation invariant models.

We believe that spreading the success of ensembles to inference higher-quality representations is the important step that will open many new applications of ensembling.

## 1. Mean Embeddings

Our goal is to utilize ensembles to improve quality of embedding produced by a neural network. Conventional ensembling techniques e.g., deep ensembles (Lakshminarayanan et al., 2017) involve averaging predictions of several independently trained networks. Representations inside these networks are not coherent: activations that appear on the

<sup>\*</sup>Equal contribution <sup>1</sup>Samsung AI Center Moscow <sup>2</sup>Swiss Federal Institute of Technology Lausanne (EPFL), work done while at National Research University Higher School of Economics. Correspondence to: Arsenii Ashukha <ars.ashuha@gmail.com>, Andrei Atanov <ai.atanov@gmail.com>.

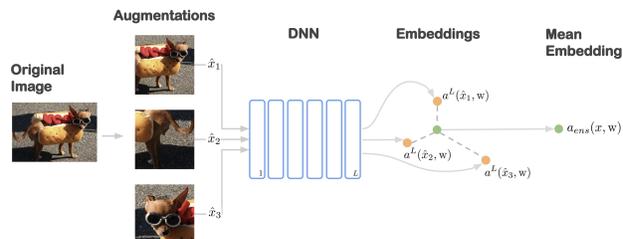


Figure 1. To produce a *mean embedding* MeTTA averages activations of a network across different augmentations of an object. The MeTTA does not affect training phase and can be applied to a pre-trained network.

same positions in different networks respond to different input signals, thus cannot be averaged. Below we explain how to get around this obstacle.

We consider a model  $p(y | x, w)$  represented by neural network with parameters  $w$  that is trained on dataset  $(x_i, y_i) \in \mathcal{D}$  of size  $K$ , where  $x$  denotes an object,  $y$  denotes a variable to be predicted, and  $w$  denotes parameters of the model. The model is a composition of transformations  $f(x, w) = f_{w^N}^N \circ \dots \circ f_{w^1}^1(x)$ . We denote intermediate representations or activations of  $L$ -th layer as  $a^L(x; w) = f_{w^L}^L \circ \dots \circ f_{w^1}^1(x)$ . We assume the model is trained by optimizing a loss function  $\mathcal{L}$  with a stochastic optimization and using data augmentation  $\hat{x} \sim p_{aug}(\hat{x} | x)$ .

$$\mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{\hat{x} \sim p_{aug}(\hat{x} | x)} \mathcal{L}(\hat{x}, w) \rightarrow \min_w. \quad (1)$$

We propose *mean embeddings with TTA (MeTTA)*—a simple method for representations ensembling. The method averages representations  $a^L(\cdot; w)$  of a single model over different augmentations of an object  $x$

$$a_{ens}(x; w) = \mathbb{E}_{\hat{x} \sim p_{aug}(\cdot | x)} a^L(\hat{x}; w) \cong \frac{1}{S} \sum_{s=1}^S a^L(\hat{x}_s; w), \text{ where } \hat{x}_s \sim p_{aug}(\cdot | x), \quad (2)$$

where  $S$  is number of samples of augmentations for a single image, and  $a_{ens}(x; w)$  is a mean embedding. In contrast to a deep ensembles like approach<sup>1</sup>, global image representa-

<sup>1</sup>We discuss possible steps to DE-like approaches in Sec. 5

Problem	Model	Width	SK	# Params (M)	Central crop	Mean embeddings with TTA	
					Embeddings	$N = 10$	$N = 32$
Self-supervised features (SimCLRv2)	ResNet50	1×	False	24	71.7	73.3 (+1.6%)	73.8 (+2.1%)
		1×	True	35	74.6	75.8 (+1.2%)	76.2 (+1.6%)
	ResNet101	2×	False	170	77.0	78.1 (+1.1%)	78.5 (+1.5%)
		2×	True	257	79.0	79.8 (+0.8%)	79.9 (+0.9%)
ResNet152	3×	True	795	79.8	80.3 (+0.4%)	80.7 (+0.9%)	
	Supervised features	ResNet50	1×	False	24	76.6	78.0 (+1.4%)
1×			True	35	78.5	79.7 (+1.2%)	80.2 (+1.7%)
ResNet101		2×	False	170	78.9	80.2 (+1.3%)	80.6 (+1.7%)
		2×	True	257	80.1	81.0 (+0.9%)	81.3 (+1.3%)
ResNet152	3×	True	795	80.5	81.4 (+0.9%)	81.9 (+1.4%)	

Table 1. Comparing of different methods of inference embeddings. Top-1 accuracy on ImageNet (Deng et al., 2009) for linear evaluation of embeddings with 100% labels. For self-supervised features we used models that were pre-trained with SimCLRv2 (Chen et al., 2020). We used both supervised and self-supervised pretrained models from <https://github.com/google-research/simclr> repository. SK states for selective kernels (Li et al., 2019). For log-likelihood score see Fig. 3

tions, so called non-spatial activations, of a single network have no misalignment between different augmentations of a single image, so activations can be safely averaged. The illustration of MeTTA is in Figure 1.

There are several reasons why MeTTA might work better than a conventional inference:

- 1. Transformation invariant representations** When a model is trained with data augmentations, it is forced to be invariant to a label preserving transformations. In practice, though, the models only partially are invariant to these transformations. Representations of different augmentations of one object may vary a lot, which over-complicates the embedding space and may significantly affect the model predictive performance. Mean embeddings, on the other hand, are invariant to such transformations by design.
- 2. Higher degree of parameter sharing during inference** Mean embeddings enjoys a higher degree of parameter sharing. It allows  $N$  times more compute per parameter during inference, and potentially, use parameters more efficiently and boost the performance of the model. Mean embeddings, however, sacrifice computational complexity of inference that is a common drawback for ensembles.
- 3. Train-test distribution shift** A model usually sees augmented data during training and clean non-augmented data during test, which can potentially introduce a domain shift, and result in a loss of performance. MeTTA does not suffer from the shift because it has exactly the same distribution of data for training and inference.

## 2. Classification with Mean Embeddings

We will use linear evaluation—a conventional protocol for evaluation of embedding that is widely used in self-

supervised learning (Chen et al., 2020). The main idea is to train a linear classifier on top of the embeddings produced by a pretrained network, while weights of the network remain fixed. Then one can compare pretrained models based on test performance of the trained linear classifier.

**But how to train a model with mean embeddings?** More formally, we are supposed to train the following model  $p(y | x)$  on top of the mean embeddings  $a_{ens}(x, w)$ :

$$p(y | x) = \text{softmax}(\theta^T a_{ens}(x, w)) = \text{softmax}(\theta^T \mathbb{E}_{\hat{x} \sim p_{aug}(\cdot | x)} a^L(\hat{x}; w)), \quad (4)$$

by minimizing a cross entropy loss

$$-\mathbb{E}_{(x,y) \sim \mathcal{D}} \log p(y | x) \rightarrow \min_{\theta}.$$

However, due to the intractable expectation inside  $p(y | x)$  that is presented under the nonlinear function, a sample-based gradients of this objective will be biased. We, therefore, substitute the loss objective with the following upper bound, whose unbiased gradients can be estimated:

$$-\log p(y | x) = -\log \text{sm}(\theta^T \mathbb{E}_{\hat{x} \sim p_{aug}(\cdot | x)} a^L(\hat{x}; w))_y \leq^2 \leq -\mathbb{E}_{\hat{x} \sim p_{aug}(\cdot | x)} \log \text{sm}(\theta^T a^L(\hat{x}; w))_y \rightarrow \min_{w, \theta}, \quad (5)$$

where  $\text{sm}$  denotes *softmax*. The equation 5 appears to be the conventional loss that is used for training a model with data augmentations. Thus, we can use a conventionally training with no additional changes needed.

**MeTTA seems to help in practice!** Using mean embeddings show stable improvement over conventional central crop embedding for both supervised and self-supervised pre-trained models (Table 1).

<sup>2</sup>Jensen’s inequality:  $f(\mathbb{E}x) \leq \mathbb{E}f(x)$  for concave  $f = -\log \text{softmax}$  (Boyd et al., 2004).

### 3. Understanding

**Intuition** Deep learning models are usually trained to raise predictions that are invariant to data augmentations. This principle is heavily used by supervised learning, and especially self-supervised learning methods based on contrastive losses e.g., SimCLR.

However, a network is only partially resistant to these transformations after training. Predictions can jitter depending on small changes in data (here we mean data augmentation, not any kind of adversarial attack), that especially pronounced on visually similar classes (Fig. 2). This effect might be caused by unstable decision boundaries.

The performance improvement obtained by averaging embeddings is, likely, caused by a step back from unstable borders in a local region of more stable predictions.

#### 1-d loss interpolations

Fig. 3 shows that mean embeddings perform better than central crop embeddings and embeddings of individual augmentations. It can be seen that loss decreases on average when approaching mean embeddings stating that this ME-direction in space of embedding is favorable.

**Relation to TTA & pre-softmax averaging** Conventional test-time augmentation (eq. 6) and mean embeddings with TTA (eq. 7) are extremely similar:

$$p(y | x) = \mathbb{E}_{\hat{x} \sim p_{aug}(\cdot | x)} sm(\theta^T a^L(\hat{x}; w)) \quad (6)$$

$$p(y | x) = sm(\theta^T \mathbb{E}_{\hat{x} \sim p_{aug}(\cdot | x)} a^L(\hat{x}; w)) \quad (7)$$

It is quite common that researches average activations of the last linear layer instead of probabilities (eq. 6). This, in the case of averaging feature from the last pre-softmax layer, is equivalent to MeTTA in terms of predictions:

$$\begin{aligned} p(y | x) &= sm(\theta^T \mathbb{E}_{\hat{x} \sim p_{aug}(\cdot | x)} a^L(\hat{x}; w)) = \\ &= sm(\mathbb{E}_{\hat{x} \sim p_{aug}(\cdot | x)} \theta^T a^L(\hat{x}; w)). \end{aligned} \quad (8)$$

An important difference is that MeTTA allows us to improve not only the final predictions but also representations, that can widen the scope of applications for ensembling techniques.

The evaluation on a classification task serves as a sanity check for the MeTTA, as the regular TTA performs (roughly) the same. But, as we will highlight in the next section there are successful use cases of MeTTA-like method in image retrieval.

### 4. Related Work

The main goal of this work is to draw the attention of ensembling and uncertainty estimation communities to the



Figure 2. Jittering of supervised ResNet50 prediction depending on a random sample of augmentation.

research on ensembling for enhancing the quality of representations.

Ensembles of representations have been used for image retrieval for a while. Specifically, aggregated multi-scale representations, is used to average representations of an image at multiple scales (Gordo et al., 2017; Radenović et al., 2018)<sup>4</sup>. Aggregated multi-scale representations is the successful use-cases of using a MeTTA-like method. It also shows that we might need different augmentation policies depending on a specific application.

Invariant DNNs heavily exploit the idea of using data aug-

<sup>4</sup>Code that uses multi-scale representations for image retrieval <https://github.com/filipradenovic/cnnimageretrieval-pytorch/blob/master/cirtorch/examples/test.py#L53>

mentations to develop invariant representations. Specifically, TI-Pooling (Laptev et al., 2016) computes the network predictions for multiple data augmentations in order to pool transformation-invariant features. The difference is that it does so during both training and inference, which makes its training step more expensive compared to MeTTA.

Independently, the idea of averaging representations has been proposed in Section 4 of (Foster et al., 2020). The proposed methods are identical, and share some motivation points e.g., a *transformation invariant* propriety. Theoretical explanations differ, and are complementary to each other e.g., we used lower-bound to motivate not using MeTTA during training, while Foster et al. (2020) provides a nice intuitive motivation. This work also contains experiments with large-scale self-supervised models.

## 5. Conclusion and Open Directions

We introduce MeTTA, a technique that uses a test-time augmentation ensemble in order to improve the representations quality. MeTTA improves the performance of both supervised and self-supervised features evaluated with linear evaluation benchmark. We also find that similar techniques have been used in image retrieval (Gordo et al., 2017; Radenović et al., 2018), as well as similar ideas were used to learn feature invariant networks (Laptev et al., 2016).

MeTTA-like methods can be potentially applied to many problems, but one should be aware of the following pitfalls:

- i)* MeTTA for spatial features, that occur in problems like detection or segmentation, needs special averaging that accounts for offsets of crops and other deformations;
- ii)* The conventional data augmentation will not, most likely, fit for any problem. For example, we found that the conventional `resize-randomcrop-flip` augmentation hurts performance of image retrieval systems, whereas a widely used handcrafted multi-scale "augmentation" improves it. In general, the issue can be resolved with a policy search for (test-time) data augmentation (Lim et al., 2019; Molchanov et al., 2020; Kim et al., 2020; Shanmugam et al., 2020).

There are the following perspectives for the usage of deep ensembles to inference better embeddings:

- i)* One way is to *synchronize* embedding spaces between different models. As all predictions in deep ensembles are synchronized by the same ground truth, so we can average predictions. There are many known tricks that can be used for synchronization, e.g., contrastive losses (Chen et al., 2020; He et al., 2020). This is still an open direction.
- ii)* The another way is to construct or learn an aggregation

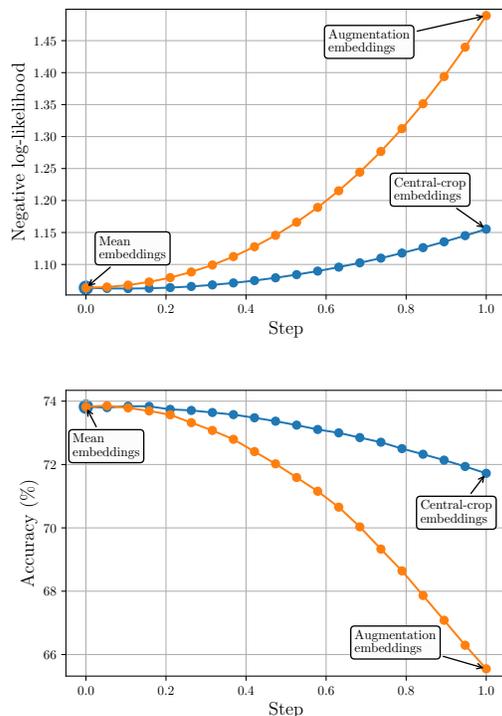


Figure 3. Negative log-likelihood (top) and accuracy (bottom) for linearly interpolated embeddings of the form  $(1 - \alpha) \cdot x + \alpha \cdot y$ , where  $x$  is the mean embedding, and  $y$  is the central-crop embedding for blue and an embeddings of an individual augmentation for orange. Metrics for each step size  $\alpha$  are averaged over validation images for both curves and additionally over different augmentations for the orange one. The embeddings are taken from the ResNet50 ( $1 \times$ , without SK) trained with SimCLRv2.

function for embeddings from non-synchronized networks. This, however, will most likely require the use of additional data, piece-wise training, as well as a the smart design of the function.

Ti-Pooling can be recognized to do both *i)* & *ii)*, but it shares the same weights across all models and train networks jointly which may hurt the predictive performance (Havasi et al., 2020).

During a review we received the following question: "Instead of only taking the mean of the embeddings, do you think it could be useful to also utilize other statistics (e.g., the variance)? Does the variance of the embedding relate to the uncertainty of the prediction?". We think it is a nice idea! It might be the way to introduce uncertainty to many problems.

We believe that spreading the success of ensembles to inference higher-quality representations is important, and will allow many new applications of ensembling. MeTTA provides a small step in this exciting direction.

## References

- Boyd, S., Boyd, S. P., and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004. URL [https://web.stanford.edu/~boyd/cvxbook/bv\\_cvxbook.pdf](https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf).
- Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- Foster, A., Pukdee, R., and Rainforth, T. Improving transformation invariance in contrastive representation learning. *arXiv preprint arXiv:2010.09515*, 2020.
- Gordo, A., Almazan, J., Revaud, J., and Larlus, D. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017.
- Havasi, M., Jenatton, R., Fort, S., Liu, J. Z., Snoek, J., Lakshminarayanan, B., Dai, A. M., and Tran, D. Training independent subnetworks for robust prediction. *arXiv preprint arXiv:2010.06610*, 2020.
- He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.
- Kim, I., Kim, Y., and Kim, S. Learning loss for test-time augmentation. *arXiv preprint arXiv:2010.11422*, 2020.
- Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pp. 6402–6413, 2017.
- Laptev, D., Savinov, N., Buhmann, J. M., and Pollefeys, M. Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 289–297, 2016.
- Li, X., Wang, W., Hu, X., and Yang, J. Selective kernel networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 510–519, 2019.
- Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. Fast autoaugment. *arXiv preprint arXiv:1905.00397*, 2019.
- Molchanov, D., Lyzhov, A., Molchanova, Y., Ashukha, A., and Vetrov, D. Greedy policy search: A simple baseline for learnable test-time augmentation. *arXiv preprint arXiv:2002.09103*, 2020.
- Radenović, F., Tolias, G., and Chum, O. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7): 1655–1668, 2018.
- Shanmugam, D., Blalock, D., Balakrishnan, G., and Gutttag, J. When and why test-time augmentation works. *arXiv preprint arXiv:2011.11156*, 2020.