Non-local NetVLAD Encoding for Video Classification

Yongyi Tang[†], Xing Zhang[‡], Jingwen Wang[†], Shaoxiang Chen[‡],
Lin Ma[†], Yu-Gang Jiang[‡]
[†]Tencent AI Lab [†]Fudan University

Outline

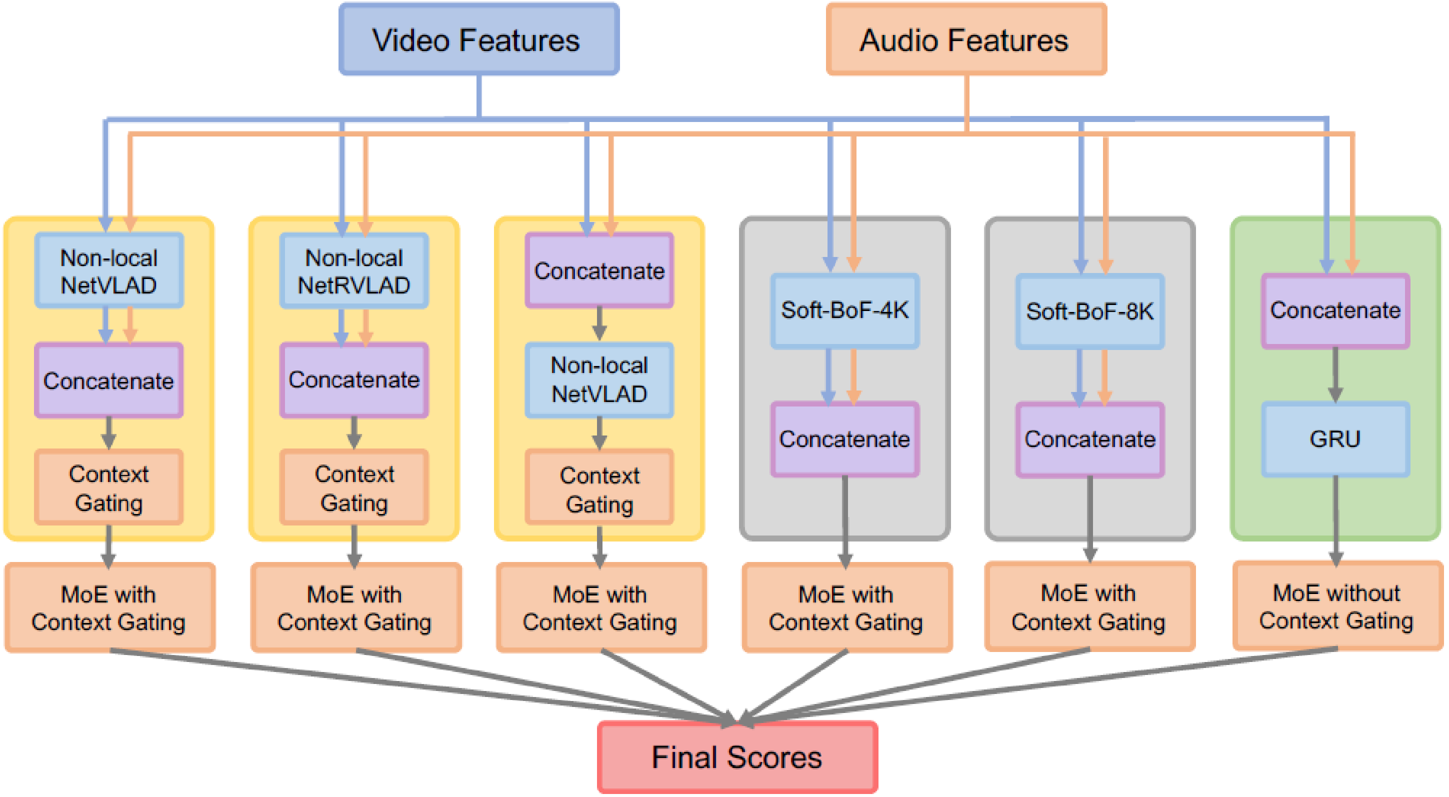
- Introduction
- Proposed Framework
- Non-local NetVLAD
- Experimental Results
- Tips and Tricks

Introduction

- Goal:
  - Achieving compatible classification result on the 2nd YouTube-8M dataset under model sized constraints given the video and audio features.
- Motivation
  - Exploring relations between features for improving single model results.
  - Seeking for complementary models and compact ensemble method.
- Our method:
  - Non-local NetVLAD.
  - Integration of NL-NetVLAD, Soft-BoF and GRU.
  - ‘bfloat-16’ format for model compression.
- Results:
  - Final ranks at the 4th place in the final announcement.
  - The proposed framework is of 995M.
  - Achieving the 0.88763 and 0.88704 GAP@20 on the public and private test set.

Proposed Framework

- Video representation learning:
  LFNL-NetVLAD, LFNL-NetRVLAD, EFNL-NetVLAD, Soft-BoF, GRU.
- Classifiers:
  Mixture of Experts, Context Gating.

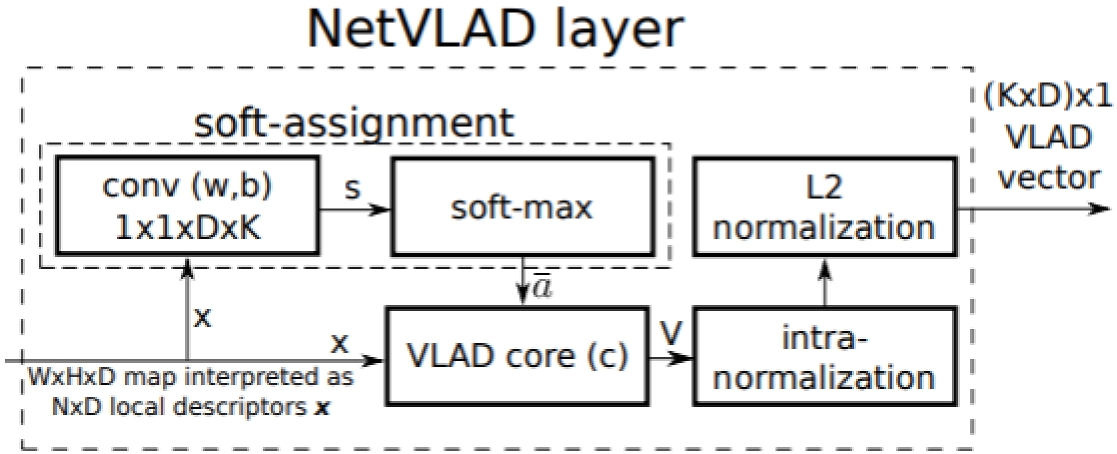


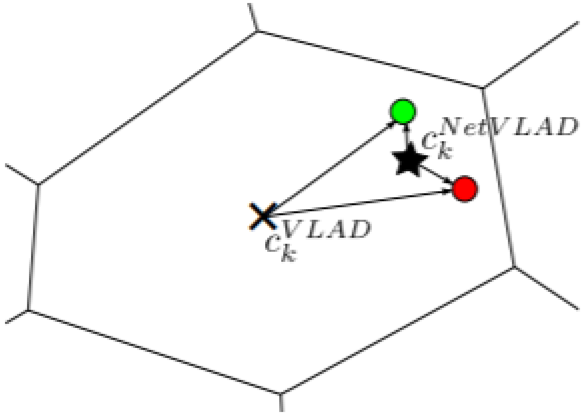
Non-local NetVLAD

NetVLAD[1] descriptor $V(j,k)$ is computed based on differentiable soft assignment $\bar{a}_k(\mathbf{x}_i)$.

$$V(j,k) = \sum_{i=1}^{N} a_k(\mathbf{x}_i)(x_i(j) - c_k(j)),$$

$$\bar{a}_k(\mathbf{x}_i) = \frac{e^{\mathbf{w}_k^T \mathbf{x}_i + b_k}}{\sum_{k'} e^{\mathbf{w}_{k'}^T \mathbf{x}_i + b_{k'}}}$$





[1] NetVLAD: CNN architecture for weakly supervised place recognition. Arandjelovic et al. CVPR2016

Non-local NetVLAD

Non-local NetVLAD descriptor models the relations between different local cluster centers with the non-local block. The non-local relations are computed with the embedded Gaussian function:

$$f(\mathbf{v}_i, \mathbf{v}_j) = e^{\theta(\mathbf{v}_i)^T \phi(\mathbf{v}_j)}$$

where $\theta(\cdot)$ and $\phi(\cdot)$ are linear transformations.

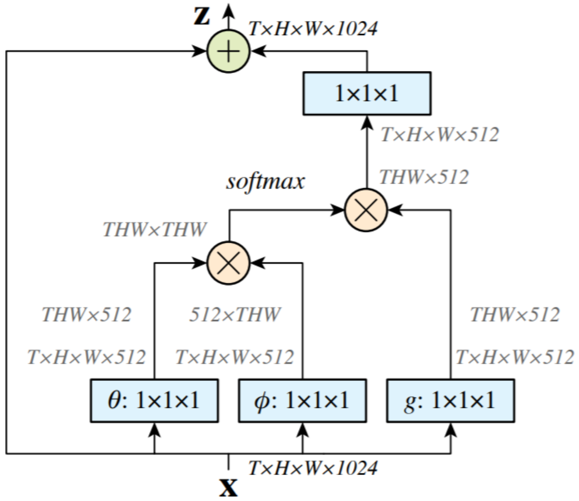


Fig1. Non-local block[1].

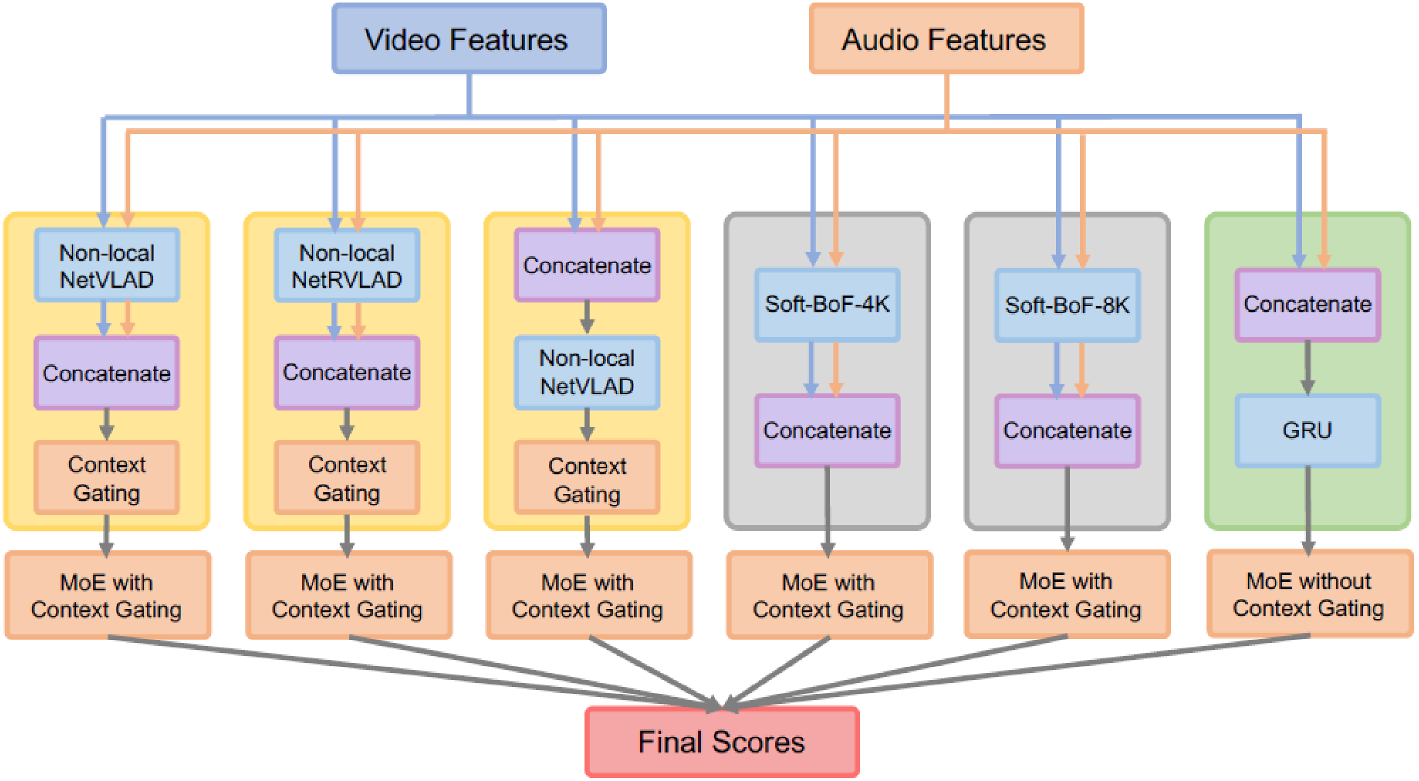
The non-local NetVLAD is formulated as:

$$\hat{\mathbf{v}}_i = \mathbf{W}\mathbf{y}_i + \mathbf{v}_i, \qquad \mathbf{y}_i = \frac{1}{Z(\mathbf{v})} \sum_{\forall j} f(\mathbf{v}_i, \mathbf{v}_j) g(\mathbf{v}_j)$$

[1] Non-local Neural Networks. Wang et al. CVPR2018

Proposed Framework

1.Late Fusion Non-local NetVLAD (64 clusters, 8MoE, 593M)
2.Late Fusion Non-local NetRVLAD (64 clusters, 4MoE, 472M)
3.Early Fusion Non-local NetVLAD (64 clusters, 2MoE, 478M)
4.Soft Bag of Features (4k clusters, 2MoE, 109M; 8k clusters, 2MoE, 143M)
5.Gate Recurrent Units (1024 hidden units, 2MoE, 243M)



Tips and Tricks

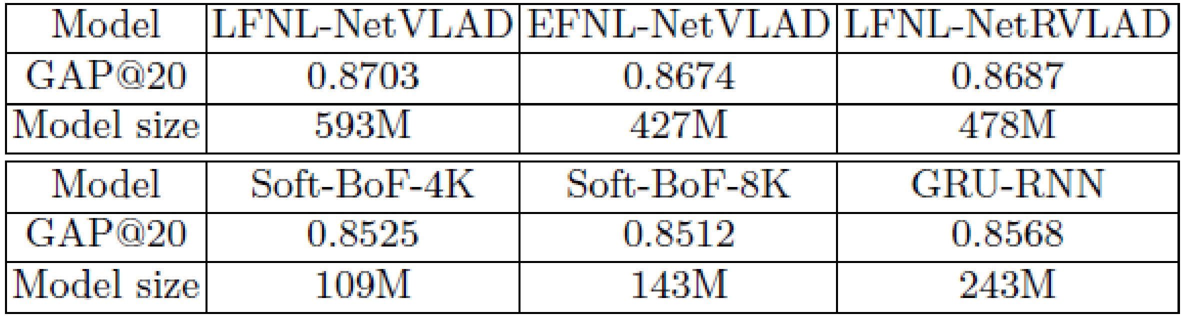
- Ensembling diverse models.
- Using ‘bfloat16’ format for model compression.
- Averaging model parameters of checkpoints gains performance.
- Multiple sampling of video frames for feature encoding.

Experimental Results

The Late Fusion Non-local NetVLAD (LFNL-NetVLAD) performs best achieving 0.8716 GAP@20 on our validation set.

Experimental results show that the NL-NetVLAD, Soft-BoF and GRU models are complimentary.

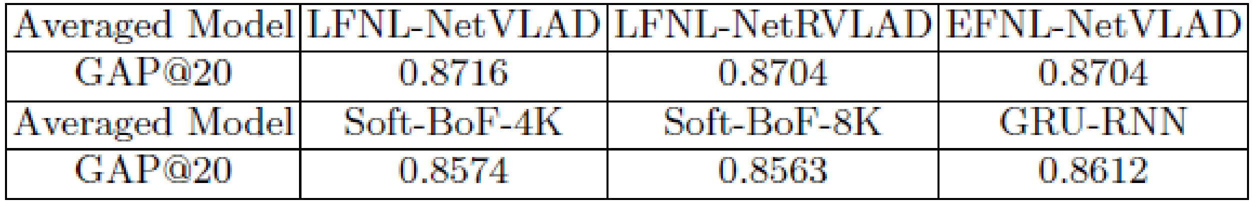
Table 1. Single model performances on our split validation set.



Model	LFNL-NetVLAD	EFNL-NetVLAD	LFNL-NetRVLAD
GAP@20	0.8703	0.8674	0.8687
Model size	593M	427M	478M

Model	Soft-BoF-4K	Soft-BoF-8K	GRU-RNN
GAP@20	0.8525	0.8512	0.8568
Model size	109M	143M	243M

Table 2. Single averaged model performances on our split validation set.



Averaged Model	LFNL-NetVLAD	LFNL-NetRVLAD	EFNL-NetVLAD
GAP@20	0.8716	0.8704	0.8704
Averaged Model	Soft-BoF-4K	Soft-BoF-8K	GRU-RNN
GAP@20	0.8574	0.8563	0.8612

Experimental Results

The Late Fusion Non-local NetVLAD (LFNL-NetVLAD) performs best achieving 0.8716 GAP@20 on our validation set.
Experimental results show that the NL-NetVLAD, Soft-BoF and GRU models are complimentary.

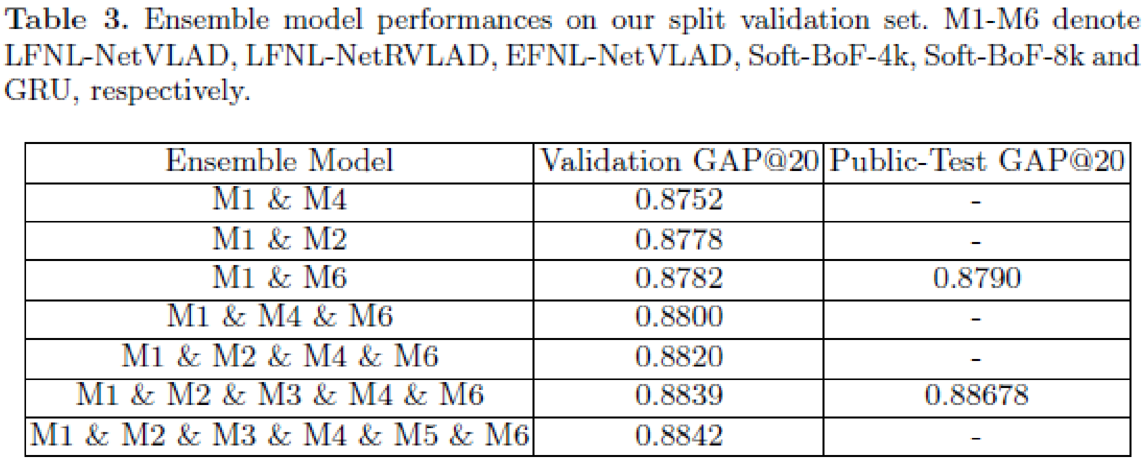


Table 3. Ensemble model performances on our split validation set. M1-M6 denote LFNL-NetVLAD, LFNL-NetRVLAD, EFNL-NetVLAD, Soft-BoF-4k, Soft-BoF-8k and GRU, respectively.

Ensemble Model	Validation GAP@20	Public-Test GAP@20
M1 & M4	0.8752	-
M1 & M2	0.8778	-
M1 & M6	0.8782	0.8790
M1 & M4 & M6	0.8800	-
M1 & M2 & M4 & M6	0.8820	-
M1 & M2 & M3 & M4 & M6	0.8839	0.88678
M1 & M2 & M3 & M4 & M5 & M6	0.8842	-

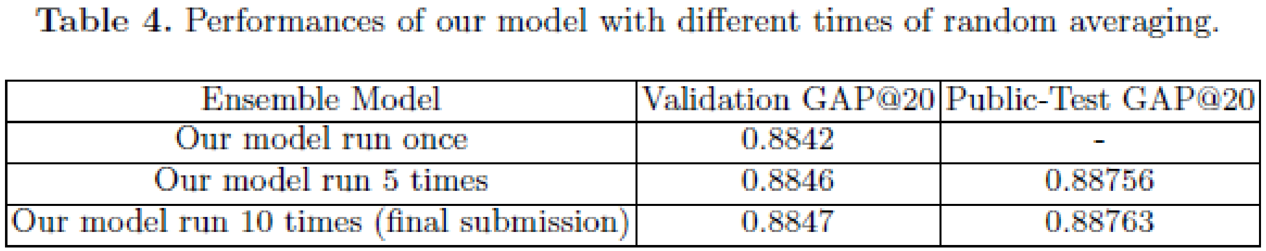
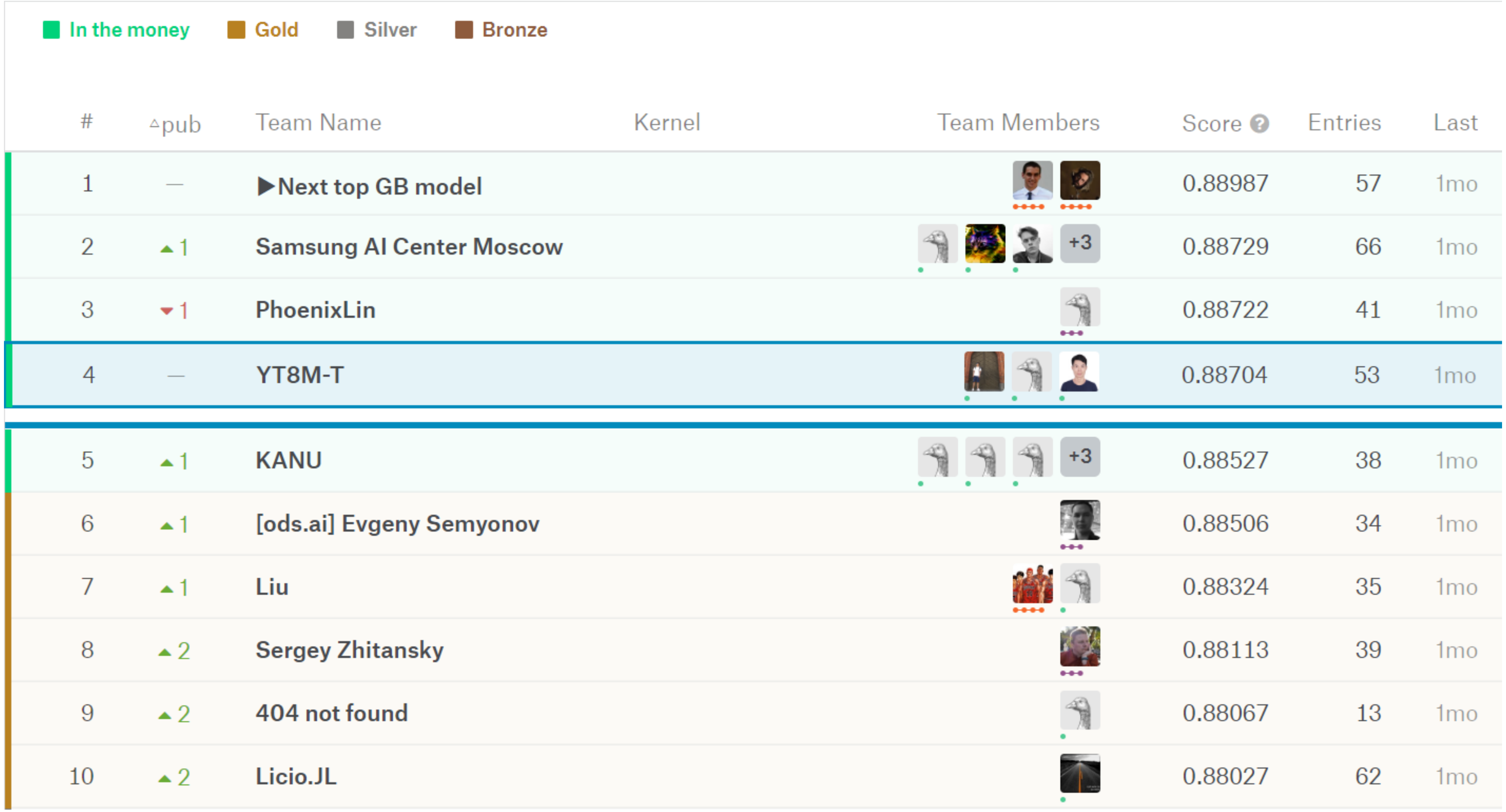


Table 4. Performances of our model with different times of random averaging.

Ensemble Model	Validation GAP@20	Public-Test GAP@20
Our model run once	0.8842	-
Our model run 5 times	0.8846	0.88756
Our model run 10 times (final submission)	0.8847	0.88763

Experimental Results



🟩 In the money 🟨 Gold ⬜ Silver 🟫 Bronze

#	▵pub	Team Name	Kernel	Team Members	Score ?	Entries	Last
1	—	▶Next top GB model			0.88987	57	1mo
2	▲1	Samsung AI Center Moscow		+3	0.88729	66	1mo
3	▼1	PhoenixLin			0.88722	41	1mo
4	—	YT8M-T			0.88704	53	1mo
5	▲1	KANU		+3	0.88527	38	1mo
6	▲1	[ods.ai] Evgeny Semyonov			0.88506	34	1mo
7	▲1	Liu			0.88324	35	1mo
8	▲2	Sergey Zhitansky			0.88113	39	1mo
9	▲2	404 not found			0.88067	13	1mo
10	▲2	Licio.JL			0.88027	62	1mo

Thanks

Q&A