Supplementary Material for Self-supervised Video Representation Learning with Cross-Stream Prototypical Contrasting

1. Example code for ViCC

Here, we provide pseudocode in PyTorch-like style for the implementation of the cross-stream stage of ViCC-RGB. For the definition of the function `sinkhorn` that describes the Sinkhorn-Knopp algorithm we refer to [6].

Pseudocode for ViCC-RGB-2 in PyTorch-like style

```
# rgb_model: encoder network for RGB
# flow_model: encoder network for flow, frozen
# temp: temperature
for rgb, flow in loader: # B samples
  # two augmented versions for two streams
  rgb_i, flow_i = aug(rgb_i, flow_i)
  rgb_j, flow_j = aug(rgb_j, flow_j)
  # get RGB and flow embeddings: 2B x D
  z_rgb = cat(rgb_model(rgb_i), rgb_model(rgb_j))
  z_flow = cat(flow_model(flow_i), flow_model(flow_j))
  # get similarity with prototypes C_rgb, C_rgb in D x K
  sim_rgb_i, sim_rgb_j = mm(z_rgb, C_rgb)
  sim_flow_i, sim_flow_j = mm(z_flow, C_rgb)
  # compute assignments
  with torch.no_grad():
    q_rgb_i, q_rgb_j, q_flow_i, q_flow_j =
    sinkhorn(sim_rgb_i), sinkhorn(sim_rgb_j),
    sinkhorn(sim_flow_i), sinkhorn(sim_flow_j)
  # convert similarity scores to probabilities
  p_rgb_i, p_rgb_j, p_flow_i, p_flow_j =
  softmax(sim_rgb_i / temp), softmax(sim_rgb_j / temp),
  softmax(sim_flow_i / temp), softmax(sim_flow_j / temp)

  # predict cluster assignments using three other views
  l_rgb_i = q_rgb_i * log(p_rgb_j)
            + q_rgb_i * log(p_flow_i)
            + q_rgb_i * log(p_flow_j)
  l_rgb_j = q_rgb_j * log(p_rgb_i)
            + q_rgb_j * log(p_flow_i)
            + q_rgb_j * log(p_flow_j)
  l_flow_i = q_flow_i * log(p_rgb_i)
            + q_flow_i * log(p_rgb_j)
            + q_flow_i * log(p_flow_j)
  l_flow_j = q_flow_j * log(p_rgb_i)
            + q_flow_j * log(p_rgb_j)
            + q_flow_j * log(p_flow_i)
  # combine for total loss for rgb model
  loss = - 1/4 * (1/3 * l_rgb_i + 1/3 * l_rgb_j +
                  1/3 * l_flow_i + 1/3 * l_flow_j)
  # optimizer update and normalize prototypes
  loss.backward()
  update(rgb_model.params), update(C_rgb)
  with torch.no_grad():
    C_rgb = normalize(C_rgb, dim=0, p=2)
```

2. Implementation Details

2.1. Implementation and Training

SGD with LARS [35] is used as the optimizer. A learning rate of 0.6, a weight decay of 10^{-6} and a cosine learning rate schedule with a final learning rate of 6×10^{-4} are chosen. The temperature τ is set to 0.1, the Sinkhorn regularization parameter ϵ is set to 0.05 and we perform 3 iterations of the Sinkhorn-Knopp algorithm. We use batch shuffle [17] to avoid the model exploiting local intra-batch information leakage for trivial solutions. For single-stream, the prototypes are frozen during the first 100 epochs of training. For cross-stream, the prototypes are directly updated from the start of the training.

2.2. Queue

To store additional features for use in the assignment to prototypes, we employ a queue in line with [6]. With 4 GPUs and a total batch size of $48 \times 4 = 192$, we adopt a queue of size 1920 to store features from the last 10 batches. The queue is introduced when the evolution of features is slowing down, *i.e.* when the decrease of the loss function is moderate. For single-stream RGB (RGB-1) we introduce the queue at 150 epochs and for Flow-1 we introduce the queue at 200 epochs. For the cross-stream stage, we introduce the queue at 25 epochs in each alternation.

3. Additional results

3.1. Analysis of Prototypes

This section focuses on further analysis of the prototypes. The main purpose of the prototype sets in ViCC is to guide the contrasting of groups of views from streams in each iteration. In combination with the relatively stable performance observed when varying the number of prototypes, we conjecture that the prototypes are not a pseudo-labeling approach similar to other methods [3, 2, 11, 5, 33]. Despite this intuition and our use of soft assignments, we investigate the prototypes by visualizing video samples assigned to the same prototypes when rounding the assignments. We also evaluate the rounded prototype assignments from several of our self-supervised stages on standard cluster evaluation metrics.

3.1.1 Visualization of Prototypes

In Figure 1 we show the hard assignment of video samples to random prototypes. Video samples with the highest similarity scores to the prototype clusters are visualized. Prototype scores are indicated on the samples and the ground truth class labels of the samples are indicated below the groups. We can observe that video samples assigned to the same prototypes share semantic similarity and even belong to the same action class, despite the fact that class labels are not used during ViCC training. The prototypes seem effective at grouping together views from the same semantic class label, as the samples visualized are all from the same class. These semantically similar sets in ViCC thereby provide an advantage for video representation learning over methods that use contrastive instance learning.

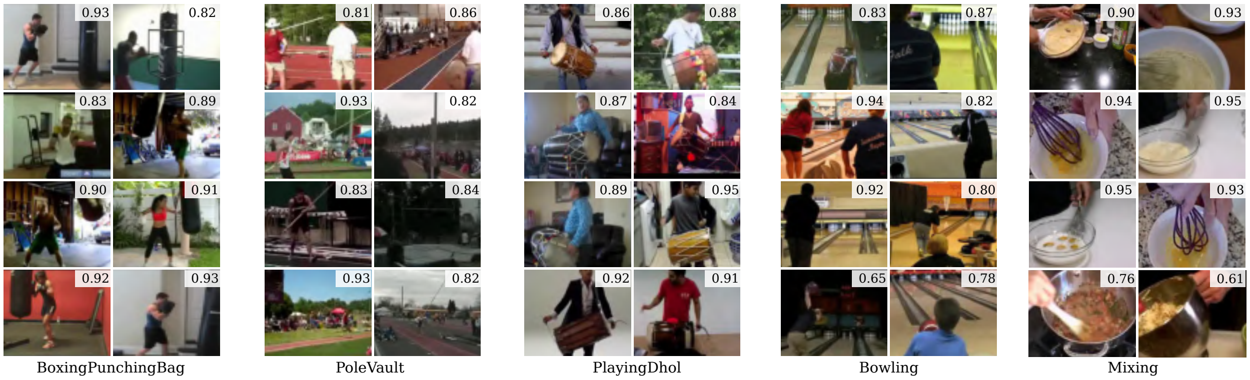

Figure 1. **Visualization of rounded assignments to random ViCC prototypes** using videos from UCF101. Samples with high similarity scores (visualized on the samples) to the prototypes are shown. The ground truth labels of all the video samples are included below (not used during training).

Method	Acc	NMI	ARI	Entropy	Max Purity
ViCC-RGB-1	32.3	62.5	16.4	1.6	36.8
ViCC-Flow-1	34.4	63.1	17.6	1.5	39.1
ViCC-RGB-2	40.8	67.8	24.5	1.4	45.1
ViCC-Flow-2	40.3	67.0	23.5	1.4	45.3

Table 1. **Cluster evaluation of ViCC prototypes** when rounding the assignments evaluated on the UCF101 test set.

3.1.2 Cluster evaluation

In this section, we evaluate the hard assignment of our prototype sets with standard cluster evaluation measures as done in [5, 3]. Although the ground truth number of clusters is not known in advance for self-supervised training, we set the number of prototypes to K=101 for evaluation purposes only to match the number of class labels for UCF101. The Hungarian algorithm [21] is then used to match self-supervised labels to the ground truth labels to obtain accuracy (Acc). We also report the Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), mean entropy per cluster (where the optimal number is 0) and mean maximal purity per cluster as defined in [3]. For example, the NMI ranges from 0 (no mutual information) to 100% (implying perfect correlation between self-supervised labels and the ground truth labels). Table 1 shows that our prototypes from the cross-stream stage (RGB-2 and Flow-2) obtain better performance on all measures compared to prototypes learned only on their own stream (RGB-1 and Flow-1), achieving *e.g.* a higher NMI, lower mean entropy per cluster and higher mean maximal purity.

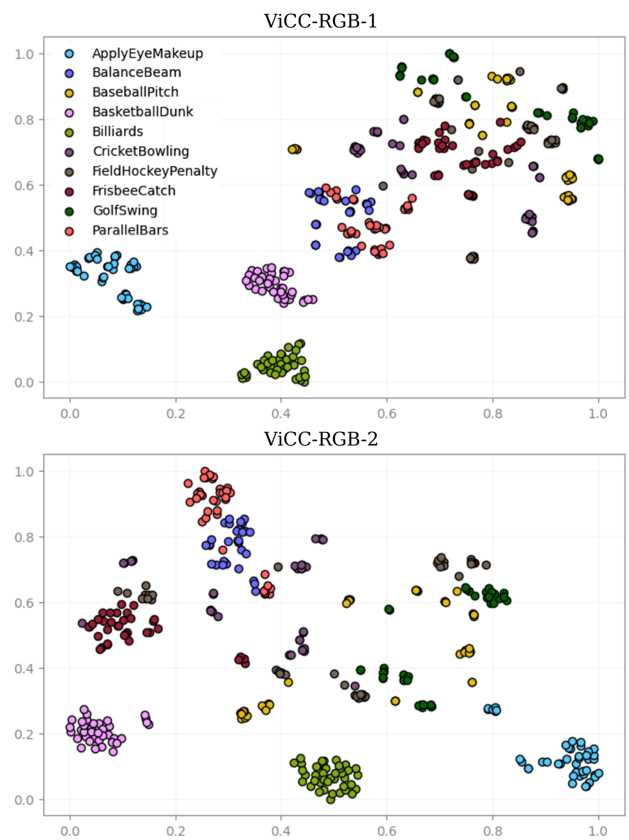

Figure 2. **T-SNE visualization** of the feature representations of UCF101 test set after 500 epochs of ViCC training. On the top RGB-1 single-stream is shown and on the bottom RGB-2 cross-stream.

3.2. T-SNE Visualization

In this section, we visualize ViCC representations of the UCF101 test set using the t-SNE clustering algorithm [30] to project features to 2D. For clarity, only 10 random action classes are visualized with a limited amount of random features for each class. Figure 2 shows the t-SNE visualization of features extracted from single-stream (RGB-1)

Method	Queue size 3840	1920	0
ViCC-RGB-2	84.5	84.3	84.7
ViCC-R+F-2	90.4	90.5	90.2

Table 2. **Impact of queue size.** We report Top-1 accuracy on action recognition finetuning on UCF101.

and cross-stream (RGB-2) trained using the same number of epochs (500). It can be observed that the inter-class distance between certain classes such as *CricketBowling* and *GolfSwing* is increased from RGB-1 to RGB-2. Moreover, the intra-class distance is reduced for classes *FrisbeeCatch*, *BasketballDunk* and *ApplyEyeMakeup*, which can be attributed to the benefit of motion learning from the flow encoder in cross-stream.

3.3. Impact of queue size

We investigate the effect of the queue size on performance. The queue is used in the assignment of features to K prototypes. In theory, using more features in each iteration on top of the current batch should result in a more accurate assignment for the Sinkhorn-Knopp algorithm. Results for queue sizes $\{3840, 1920, 0\}$ are shown in Table 2. We report Top-1 accuracy on action recognition on UCF101 finetuning. For queue size 3840, we observe that the larger queue size is not necessary or beneficial for UCF101 self-supervised pretraining, as the differences in performance are minimal. We also find that using no queue almost performs on par with our default queue size of 1920. We conjecture that our mini-batches may already provide enough features for ViCC self-supervision on UCF101.

3.4. More comparison with self-supervised works on action recognition

In Table 3 we list more results from self-supervised methods evaluated on action recognition. Results for the additional backbone R3D-18 [16] are included. We achieve better performance than several methods that use the R3D backbone. Our overall best result on the S3D backbone still outperforms almost all methods pretrained on UCF101. We also outperform several methods pretrained on the larger dataset K-400, and achieve competitive performance compared to CVRL [28].

References

[1] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-Supervised Learning by Cross-Modal Audio-Video Clustering. In *NeurIPS*, 2020.

[2] Yuki M. Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*, 2020.

[3] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

[4] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. SpeedNet: Learning the Speediness in Videos. In *CVPR*, 2020.

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV*, 2018.

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*, 2020.

[7] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A Short Note about Kinetics-600. *ArXiv preprint arXiv:1808.01340*, 2018.

[8] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*, 2017.

[9] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A Large-Scale Audio-Visual Dataset. In *ICASSP*, 2020.

[10] H. Cho, Tae-Hoon Kim, H. J. Chang, and Wonjun Hwang. Self-Supervised Spatio-Temporal Representation Learning Using Variable Playback Speed Prediction. *ArXiv preprint arXiv:2003.02692*, 2020.

[11] Kirill Gavrilyuk, Mihir Jain, Ilia Karmanov, and Cees G. M. Snoek. Motion-Augmented Self-Training for Video Recognition at Smaller Scale. *ArXiv preprint arXiv:2105.01646*, 2021.

[12] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio Set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017.

[13] Tengda Han, Weidi Xie, and Andrew Zisserman. Video Representation Learning by Dense Predictive Coding. In *ICCV Workshop*, 2019.

[14] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented Dense Predictive Coding for Video Representation Learning. In *ECCV*, 2020.

[15] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised Co-training for Video Representation Learning. In *NeurIPS*, 2020.

[16] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In *CVPR*, 2018.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*, 2020.

[18] S. Jenni, Givi Meishvili, and P. Favaro. Video Representation Learning by Recognizing Temporal Transformations. In *ECCV*, 2020.

[19] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-Supervised Spatiotemporal Feature Learning via Video Rotation Prediction. *ArXiv preprint arXiv:1811.11387*, 2019.

	Pretrain stage							Linear		Finetune	
Method	Year	Dataset	Backbone	Param	Res	Frames	Modality	UCF101	HMDB51	UCF101	HMDB51
OPN [22]	2017	UCF101	VGG	8.6M	80	16	V	-	-	59.8	23.8
VCOP [32]	2019	UCF101	R(2+1)D	14.4M	112	16	V	-	-	72.4	30.9
Var. PSP [10]	2020	UCF101	R(2+1)D	14.4M	112	16	V	-	-	74.8	36.8
Pace Pred [31]	2020	UCF101	R(2+1)D	14.4M	112	16	V	-	-	75.9	35.9
VCP [23]	2020	UCF101	R(2+1)D	14.4M	112	16	V	-	-	66.3	32.2
PRP [34]	2020	UCF101	R(2+1)D	14.4M	112	16	V	-	-	72.1	35.0
RTT [18]	2020	UCF101	R(2+1)D	14.4M	112	16	V	-	-	81.6	46.4
Pace Pred [31]	2020	K-400	R(2+1)D	14.4M	112	16	V	-	-	77.1	36.6
MotionFit [11]	2021	K-400	R(2+1)D	14.4M	112	32	V	-	-	88.9	61.4
XDC [1]	2020	K-400	R(2+1)D	14.4M	224	32	V+A	-	-	86.8	52.6
SeLaVi [2]	2020	VGG-sound [9]	R(2+1)D	14.4M	112	30	V+A	-	-	87.7	53.1
GDT [26]	2020	Audioset [12]	R(2+1)D	14.4M	224	32	V+A	-	-	92.5	66.1
ViCC-RGB (*ours*)		UCF101	R(2+1)D	14.4M	128	16	V	**74.4**	**30.8**	**82.8**	**52.4**
ViCC-R+F (*ours*)		UCF101	R(2+1)D	14.4M	128	16	V	**78.3**	**45.2**	**88.8**	**61.5**
DPC [13]	2019	UCF101	R2D3D	14.2M	128	40	V	-	-	60.6	-
MemDPC [14]	2020	UCF101	R2D3D	14.2M	224	40	V	-	-	84.3	-
VCOP [32]	2019	UCF101	R3D	14.2M	112	16	V	-	-	64.9	29.5
Var. PSP [10]	2020	UCF101	R3D	14.2M	112	16	V	-	-	69.0	33.7
VCP [23]	2020	UCF101	R3D	14.2M	112	16	V	-	-	66.0	31.5
PRP [34]	2020	UCF101	R3D	14.2M	112	16	V	-	-	66.5	29.7
RTT [18]	2020	UCF101	R3D	14.2M	112	16	V	-	-	77.3	47.5
RotNet3D [19]	2019	K-400	R3D	33.6M	224	16	V	-	-	62.9	33.7
ST-Puzzle [20]	2019	K-400	R3D	33.6M	224	16	V	-	-	65.8	33.7
DPC [13]	2019	K-400	R3D	14.2M	128	40	V	-	-	68.2	34.5
VIE [36]	2020	K-400	R3D	14.2M	112	40	V	-	-	72.3	44.8
CVRL [28]	2021	K-400	R3D-50	36.1M	224	16	V	-	-	92.1	65.4
ViCC-RGB (*ours*)		UCF101	R3D	14.2M	128	16	V	**69.0**	**44.2**	**78.2**	**44.7**
ViCC-R+F (*ours*)		UCF101	R3D	14.2M	128	16	V	**73.3**	**46.7**	**85.7**	**53.2**
Pace Pred [31]	2020	UCF101	S3D-G	9.6M	224	64	V	-	-	87.1	52.6
CoCLR [15]	2020	UCF101	S3D	8.8M	128	32	V	70.2	39.1	81.4	52.1
CoCLR † [15]	2020	UCF101	S3D	8.8M	128	32	V	72.1	40.2	87.3	58.7
CoCLR † [15]	2020	K-400	S3D	8.8M	128	32	V	77.8	52.4	90.6	62.9
SpeedNet [4]	2020	K-400	S3D-G	8.8M	128	32	V	-	-	81.1	48.8
MIL-NCE [24]	2020	HTM [25]	S3D	8.8M	224	32	V+T	82.7	53.1	91.3	61.0
CBT [29]	2019	K-600 [7]	S3D	8.8M	112	16	V+T	54.0	29.5	79.5	44.6
ELo [27]	2020	K-400	S3D	8.8M	224	32	V+T	-	-	93.8	67.4
ViCC-RGB (*ours*)		UCF101	S3D	8.8M	128	32	V	**72.2**	**38.5**	**84.3**	**47.9**
ViCC-R+F (*ours*)		UCF101	S3D	8.8M	128	32	V	**78.0**	**47.9**	**90.5**	**62.2**

Table 3. **Comparison with prior self-supervised works on video action recognition** on UCF101 and HMDB51 for finetuning and linear probe. We report Top-1 accuracy, compare with self-supervision pretraining on UCF101 and additionally report results on backbone R3D [16]. In grey color we show larger pretraining datasets such as K-400 [8] and multi-modal datasets (where T is text, A is audio).

[20] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles. In *AAAI*, 2019.

[21] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[22] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised Representation Learning by Sorting Sequences. In *ICCV*, 2017.

[23] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video Cloze Procedure for Self-Supervised Spatio-Temporal Learning. In *AAAI*, 2020.

[24] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*, 2020.

[25] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019.

[26] Mandela Patrick, Yuki M. Asano, Polina Kuznetsova, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal Self-Supervision from Generalized Data Transformations. *ArXiv preprint arXiv:2003.04298*, 2020.

[27] A. J. Piergiovanni, Anelia Angelova, and Michael S. Ryoo. Evolving Losses for Unsupervised Video Representation Learning. In *CVPR*, 2020.

[28] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal Contrastive Video Representation Learning. In *CVPR*, 2021.

[29] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning Video Representations using Contrastive Bidirectional Transformer. *ArXiv preprint arXiv:1906.05743*, 2019.

[30] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[31] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised Video Representation Learning by Pace Prediction. In *ECCV*, 2020.

[32] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-Supervised Spatiotemporal Learning via Video Clip Order Prediction. In *CVPR*, 2019.

[33] Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. ClusterFit: Improving Generalization of Visual Representations. In *CVPR*, 2020.

[34] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video Playback Rate Perception for Self-supervisedSpatio-Temporal Representation Learning. In *CVPR*, 2020.

[35] Yang You, Igor Gitman, and Boris Ginsburg. Large Batch Training of Convolutional Networks. *ArXiv preprint arXiv:1708.03888*, 2017.

[36] Chengxu Zhuang, Tianwei She, Alex Andonian, Max Sobol Mark, and Daniel Yamins. Unsupervised Learning from Video with Deep Neural Embeddings. In *ArXiv Preprint arXiv:1905.11954*, 2020.