Unbiased Scene Graph Generation in Videos
Supplementary Material

Sayak Nag[1], Kyle Min[2], Subarna Tripathi[2], Amit K. Roy-Chowdhury[1]
[1]University of California, Riverside, USA, [2]Intel Corporation, USA
snag005@ucr.edu, kyle.min@intel.com, subarna.tripathi@intel.com, amitrc@ece.ucr.edu

The supplementary material provides more details, results, and visualizations to support the main paper. In summary, we include additional implementation details, more experiments and ablation studies, analysis of our results, more qualitative visualizations, and a discussion on future works.

1. Additional Implementation Details

Predicate Class Distribution. We define the *HEAD*, *BODY* and *TAIL* relationship classes in Action Genome (Action Genome) [3] as shown below,

- *HEAD* $\geq$ 100000 training samples
- 8000 training samples $\leq$ *BODY* $<$ 100000 training samples
- *TAIL* $<$ 8000 training samples

Experimental Setup. The architecture of the PEG is kept the same as [1]. The Faster-RCNN object detector [7] is first trained on Action Genome [3] following [1, 5, 10]. Following prior work per-class non-maximal suppression at 0.4 IoU is applied to reduce region proposals provided by the Faster-RCNN's RPN. The sequence encoder in the OSPU is designed with 3 layers, each having 8 heads for its multi-head attention. The dimension of its FFN projection is 1024. During training, we reduce the initial learning rate by a factor of 0.5 whenever the performance plateaus. All codes are run on a single NVIDIA RTX-3090.

Evaluation Metrics. We follow the official implementation of [8] for the mean-Recall@K (mR@K) metric. Different from the standard Recall@K (R@K), mR@K is computed by first obtaining the recall values of each predicate class and then averaging them over the total number of predicates. Therefore if an SGG model consistently fails to detect any of the visual relationships i.e., predicates, the mR@K value will drop considerably. This makes it a much more balanced metric compared to R@K, which is obtained by averaging recall values over the entire dataset. Therefore improvement on the high-frequent classes alone is sufficient for high R@K values. For all experiments, the reported results are in terms of image-based R@K and mR@K. Since the video-based R@K is a simple averaging of the per-frame measurements, most existing works adopt the image-based metrics [1, 4, 5, 10].

Table I. Importance of uncertainty attenuation and memory guided meta-debiasing for PREDCLS.

Uncertainty Attenuation	Memory guided Debiasing	With Constraint		No Constraints	
		mR@10	mR@20	mR@10	mR@20
-	-	37.8	40.1	51.4	67.7
✓	-	40.2	44.0	55.1	77.3
-	✓	41.1	44.8	57.0	82.9
✓	✓	**42.9**	**46.3**	**61.5**	**85.1**

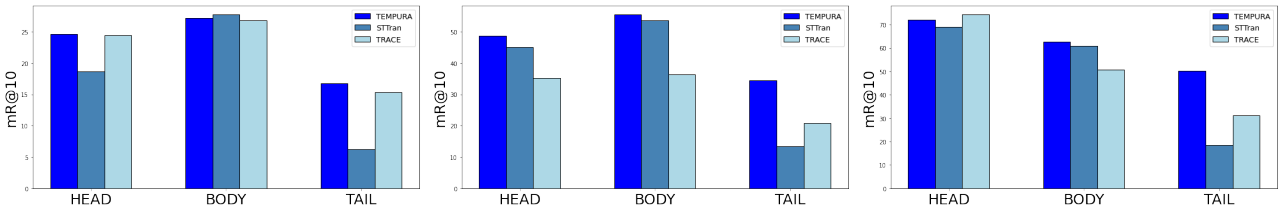
Table II. Impact of $\mathcal{L}_{intra}$.

$\mathcal{L}_{intra}$	With Constraint				No Constraints			
	SGCls		SGDet		SGCls		SGDet	
	mR@10	mR@20	mR@10	mR@20	mR@10	mR@20	mR@10	mR@20
-	32.1	33.2	17.9	22.1	46.5	60.4	23.5	32.8
✓	**34.0**	**35.2**	**18.5**	**22.6**	**48.3**	**61.1**	**24.7**	**33.9**

Baseline Performance. For the baselines STTran [1], TRACE [9], and RelDN [11], we used their official code implementation to obtain the respective mR@K values. We obtained the mR@K values of STTran-TPI [10] from email discussions. The performance of HCRD supervised [2], and ISGG [4] are taken from the reported values in [4]. As explained in Section 4.1 of the main paper, SGG performance is typically evaluated under two different setups **With Constraint** and/or **No Constraints**. All the baselines we compare with either follow the **With Constraint** [2, 4, 10] setup or the **No Constraints** setup [9] or both [1, 5].

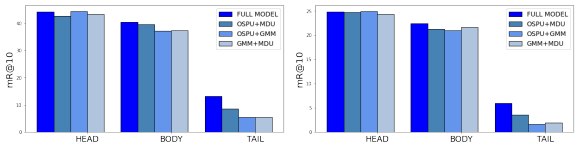
2. Additional Comparative Results

Comparison of *HEAD*, *BODY* and *TAIL* class performance, with SOTA, under No Constraints. Comparative performance on the *HEAD*, *TAIL* and *BODY* classes of Action Genome under the **No Constraints** setup are shown in Fig I.



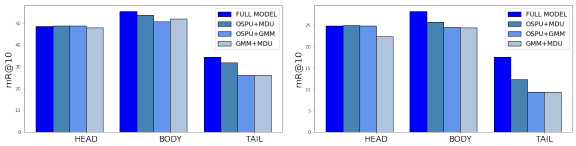
(a) SGDET (b) SGCLS (c) PREDCLS

Figure I. Comparison of mR@10 for the HEAD, BODY and TAIL classes in Action Genome [3] under the "No constraints" setup.



(a) SGCLS (b) SGDET

Figure II. Comparison of mR@10 for the HEAD, BODY, and TAIL classes in Action Genome [3] for different ablation setups of Table 4. Performances reported under the "with constraint" setup.



(a) SGCLS (b) SGDET

Figure III. Comparison of mR@10 for the HEAD, BODY, and TAIL classes in Action Genome [3] for different ablation setups of Table 4. Performances reported under the "no constraints" setup.

Similar to the **With Constraint** results shown in Fig 7 of the main paper, TEMPURA outperforms both TRACE [9] and STTran [1] in improving performance on the *TAIL* and *BODY* classes without significantly compromising performance on the *HEAD* classes. While TRACE performs well under the **No Constraints** setup, its performance under the **With Constraints** setup is lacking (Fig 7 main paper). TEMPURA, on the other hand, shows consistent performance for both setups, beating TRACE and STTran in generating more unbiased scene graphs.

3. Additional Ablations

Ablations for PREDCLS task. The impact of memory guided debiasing and uncertainty attenuation for the *PREDCLS* task can be seen in Table I. Similar to the other SGG tasks (Table 4 of the main paper), incorporating both principles gives the best results. Since, for PREDCLS, the object bounding boxes and classes are already provided, the OSPU is inactive for this SGG task.

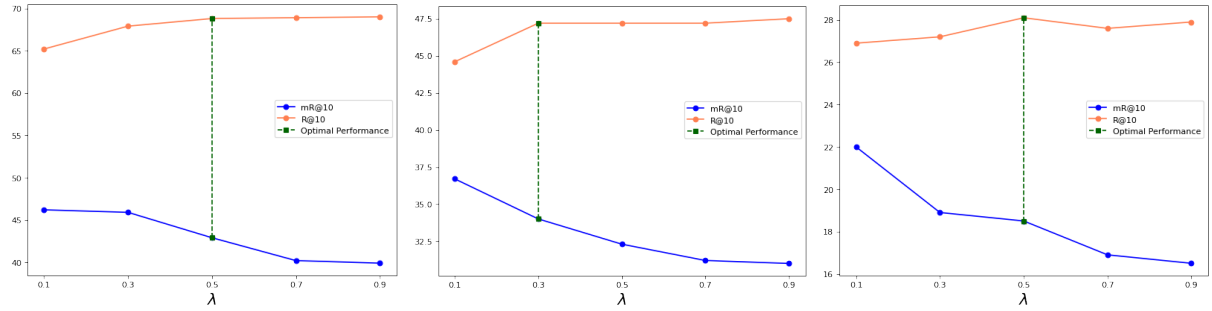
Comparison of *HEAD*, *BODY* and *TAIL* class perforamnce for SGCLS and SGDET ablations. From Fig II and Fig III, we can observe that using the full model (OSPU+MDU+GMM) gives the best performance for the *BODY* and *TAIL* classes for both SGCLS and SGDET tasks.

Impact of $\mathcal{L}_{intra}$. The impact of $\mathcal{L}_{intra}$ can ascertained from Table II. The results show that utilizing the intra-video contrastive loss $\mathcal{L}_{intra}$ boosts the sequence processing capability of the OSPU, leading to more consistent object classification and, consequently, more unbiased scene graphs.

Table III. Comparitive performance of TEMPURA for three different settings of λ. No λ corresponds to when the weighted residual operation of Eq 11 is replaced with a standard residual connection. The optimal values of λ are 0.5, 0.3, and 0.5 for PREDCLS, SGCLS, and SGDET, respectively.

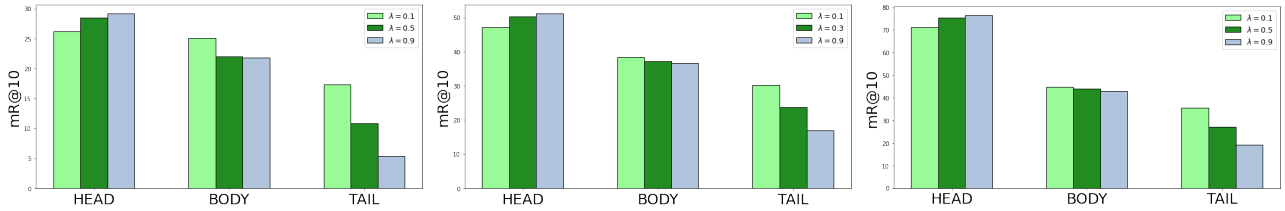
λ*Setting*	PredCLS		SGCls		SGDET	
	mR@10	mR@20	mR@10	mR@20	mR@10	mR@20
λ=0	31.7	36.9	25.0	26.2	13.4	17.9
No λ	39.4	43.1	30.8	32.2	17.3	21.6
Optimal λ	**42.9**	**46.3**	**34.0**	**35.2**	**18.5**	**22.6**

Ablations on λ. The gradient scaling factor λ regulates the influence of the direct PEG embedding $\boldsymbol{r}^j_{tem}$ of the j^{th} subject-object pair and the compensatory information of the diffused memory feature $\boldsymbol{r}^j_{mem}$ as shown in Eq 11 in the main paper. $\lambda \in (0,1]$ i.e. $0 < \lambda \leq 1$. To obtain the optimal value of λ, we vary it within $[0.1, 0.3, 0.5, 0.7, 0.9]$ and observe the corresponding With Constraint R@10 and mR@10 values as shown in Fig IV. As observed in Fig IV, increasing the value of λ causes the R@10 values to also increase before stagnating after a certain point. However, the corresponding mR@10 values start falling for higher λ values. Since the Recall@K metric is an indicator of how well an SGG model is performing on the data-rich predicates, this indicates that for higher λ values, the compensatory effect of $\boldsymbol{r}^j_{tem}$ is drastically reduced, and the PEG fails to generate more unbiased representations. On the other hand, if λ is set to small values like 0.1, high mR@10 values can be observed, but this comes at the expense of R@10 performance, indicating a drop in performance of the *HEAD* classes due to excessive knowledge being transferred from these data-rich classes to the data-poor ones. Since the goal of unbiased SGG is not to perform well on the data-poor



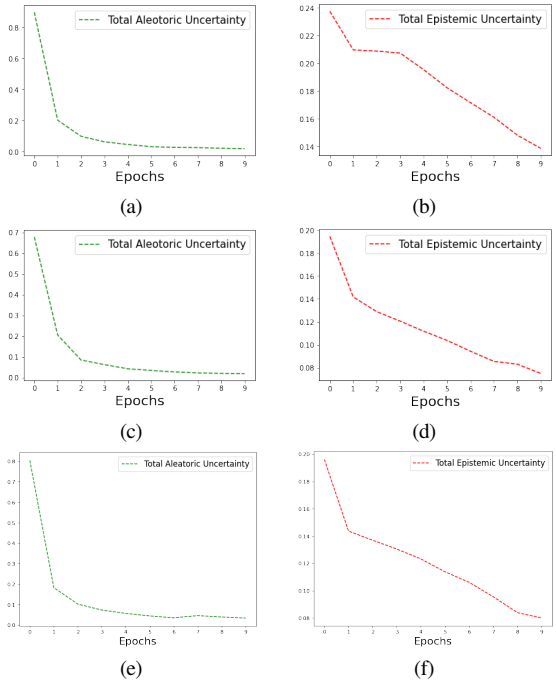
(a) PREDCLS (b) SGCLS (c) SGDET

Figure IV. Comparison of R@10 and mR@10 performance of TEMPURA for different values of λ.



(a) SGDET (b) SGCLS (c) PREDCLS

Figure V. Comparison of mR@10 for the HEAD, BODY and TAIL classes in Action Genome [3] for different λ values.



(a) (b) (c) (d) (e) (f)

Figure VI. Top to Bottom: Predictive uncertainty for *PREDCLS* (a,b), Predictive Uncertainty for *SGCLS* (c,d) and Predictive Uncertainty for *SGDET* (e,f).

classes at the expense of data-rich classes, it is necessary to set λ to an optimal value that gives the best balance between recall and mean-recall performance. As shown in Fig IV the optimal λ is 0.5 for *PREDCLS* and *SGDET*, and 0.3 for *SGCLS*. From Fig V we can observe that setting λ to the optimal values gives the best balance in performance over the *HEAD*, *BODY* and *TAIL* classes as opposed to setting λ to very low and very high values.

If $\lambda = 1$, there is no impact of $\boldsymbol{r}^j_{mem}$, and the model relies solely on the uncertainty attenuation of the GMM head. As explained in section 3.5 of the main paper, $\boldsymbol{r}^j_{mem}$ essentially hallucinates information relevant to the data-poor classes otherwise missing from the original PEG embedding $\boldsymbol{r}^j_{tem}$ and the weighted residual operation of Eq 11 acts as a mechanism to diffuse this compensatory information back to $\boldsymbol{r}^j_{tem}$ in order to make it more balanced. Therefore, λ can never be 0; otherwise, there will be no PEG embedding to debias, and the MDU will never be able to teach the framework how to generate more unbiased embeddings. On the other hand, if λ is not used in the diffusion operation of Eq 11 i.e., if a standard non-weighted residual operation is used, the biased information from $\boldsymbol{r}^j_{tem}$ tends to overpower the effect of $\boldsymbol{r}^j_{mem}$. To verify this, we set up two experiments. In the first case, we set $\lambda = 0$ and train the model, and in the second case, we replace the weighted residual operation of Eq 11 with a simple residual operation i.e. $\hat{\boldsymbol{r}}^j_{tem} = \boldsymbol{r}^j_{tem} + \boldsymbol{r}^j_{mem}$ and then train the model. It can be observed from the With Constraint results shown in Table III that setting λ to 0 results in a significant drop in mR@K performance since the MDU is unable to diffuse the compensatory information back to the original PEG embedding rendering it ineffective in regularizing the model towards generating more unbiased predicate embeddings. Additionally, by comparing rows 2 and 3 in Table III, we can infer that utilization of λ for the weighted residual operation of Eq 11 is necessary to get the best performance in terms of mean-recall.

Table IV. Comparison of performance when the memory bank $\mathbf{\Omega}_R$ and MDU is used and not used during inference. In both cases, the same model is used during inference which has been trained using the MDU. The results in the first row correspond to when MDU acts as a network module, and those in the second row correspond to when MDU is used as a meta-regularization unit which is its intended purpose. It can be observed from the results that incorporating the training memory bank $\mathbf{\Omega}_R$ for the test videos can bias the relationship representations towards the training distribution, defeating the purpose of the MDU.

MDU used during Inference	With Constraint						No Constraints					
	PredCLS		SGCls		SGDet		PredCLS		SGCls		SGDet	
	mR@10	mR@20	mR@10	mR@20	mR@10	mR@20	mR@10	mR@20	mR@10	mR@20	mR@10	mR@20
✓	38.5	42.0	29.9	31.2	16.1	20.4	53.6	80.0	42.5	57.5	19.4	29.6
-	**42.9**	**46.3**	**34.0**	**35.2**	**18.5**	**22.6**	**61.5**	**85.1**	**48.3**	**61.1**	**24.7**	**33.9**

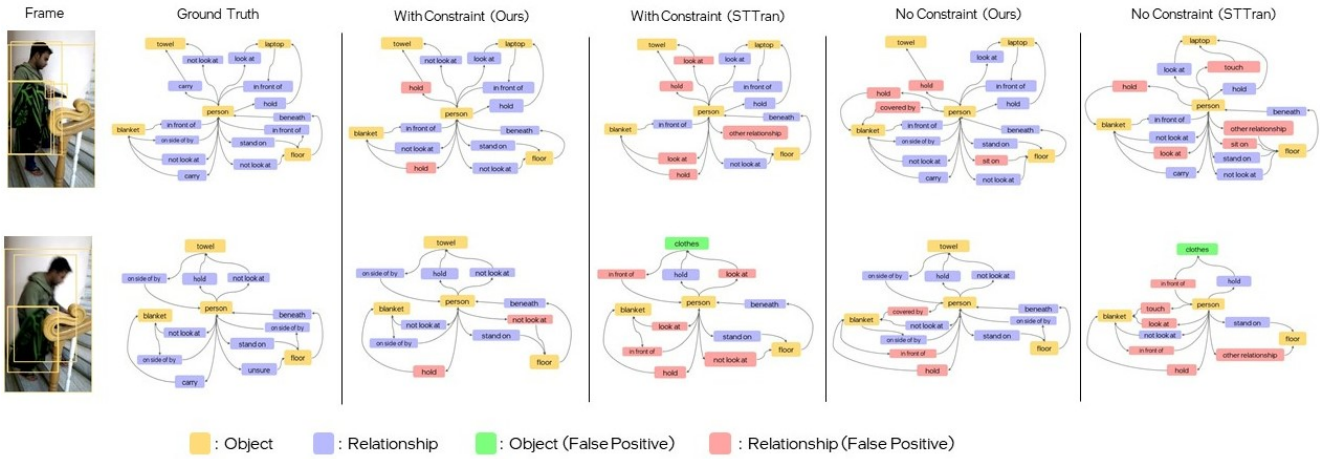


Figure VII. **Comparative qualitative results**. From left to right: input video frames, ground truth scene graphs, scene graphs generated by TEMPURA and the scene graphs generated by the baseline STTran. Incorrect object and predicate predictions are shown in green and pink, respectively.

4. Additional Analysis

Are the effects of high uncertainty being attenuated? The predicate classification loss, $\mathcal{L}_p$ (Eq 16) is designed to penalize the model if it predicts high uncertainty for any sample. This means the model progressively becomes more efficient in attenuating the effect of noisy samples, which inherently decreases its predictive uncertainty with the number of epochs. This can be visualized in Fig VI, which shows the total predictive uncertainty of the full model for each SGG task. Both the epoch-specific *aleatoric* and *epistemic* uncertainties are obtained by averaging across all samples (subject-object pairs) over all classes.

Role of Memory Diffusion Unit. As explained in section 3.5, the MDU and the predicate class-centric memory bank $\mathbf{\Omega}_R$ are used during the training phase as a structural meta-regularizer to debias the direct PEG embeddings and inherently teach the PEG how to learn more unbiased predicate embeddings. One might ask why the MDU and the training memory bank $\mathbf{\Omega}_R$ cannot be used as a network module to forward pass through during the inference like many memory-based works on long tail image recognition [6, 12]. This is because of the distributional shift between training and testing sets in the video SGG dataset. Such distributional shift also exist in standard image recognition datasets but is very minimal. That is not the case for video SGG data. For instance, unlike an image recognition dataset each sample of a visual relationship is not i.i.d. The visual relationship between a subject-object pair at each frame depends on the visual relationships (between the same pair) in the previous frames, and this temporal evolution is captured by $TempDec$ based on the motion information coming from the proposal features, shifting bounding boxes and union features of the subject-object pair. The temporal evolution of many visual relationships in the test videos can differ greatly from those in the training videos. This spatio-temporal information of each predicate class, *in the entire training set*, is compressed into their respective memory prototypes $\boldsymbol{\omega}_p \in \mathbf{\Omega}_R$. Therefore utilizing these predicate memory prototypes (for the MDU operation) during inference biases the framework towards the training distribution which is antithetical to the purpose of the MDU. Additionally, the issue of triplet variability, shown in Fig 2 of the main paper, can further deepen the distribution shift since certain triplets associated

with a relationship class can occur only during inference. For example, the triplets $<person-above-refrigerator>$ and $<person-lying\ on-bag>$ associated with the $above$ and $lying\ on$ predicates occur in only the test set videos of Action Genome [3]. The information associated with these unique triplets is never incorporated in $\mathbf{\Omega}_R$, consequently impacting the predicate embedding if $\mathbf{\Omega}_R$ is used during inference which can lead to a drop in performance. We verify this by conducting a short experiment the results of which are shown in Table IV.

More Qualitative Results. Some more qualitative results are shown in Fig VII. It can be seen that TEMPURA prevents fewer false positives compared to the baseline STTran [1].

5. Limitations and Future Work

The progressive computation of the memory bank as a set of prototypical centroids does increase the training time, but as we showed in our results, this memory-guided training approach can result in more unbiased predicate representations that inherently help in the generation of more unbiased scene graphs. In future works, we aim to explore a parallel memory computation approach that can perform at par with our current method.

References

[1] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. Spatial-Temporal Transformer for Dynamic Scene Graph Generation. In *Proceedings of the International Conference on Computer Vision (ICCV)*, October 2021. 1, 2, 5

[2] Raghav Goyal12 and Leonid Sigal123. A simple baseline for weakly-supervised human-centric relation detection. 2021. 1

[3] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action Genome: Actions as Composition of Spatio-temporal Scene Graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 3, 5

[4] Siddhesh Khandelwal and Leonid Sigal. Iterative scene graph generation. 2022. 1

[5] Yiming Li, Xiaoshan Yang, and Changsheng Xu. Dynamic scene graph generation via anticipatory pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13874–13883, June 2022. 1

[6] Sarah Parisot, Pedro M Esperança, Steven McDonagh, Tamas J Madarasz, Yongxin Yang, and Zhenguo Li. Long-tail recognition via compositional knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6939–6948, 2022. 4

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 91–99, Cambridge, MA, USA, 2015. MIT Press. 1

[8] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[9] Yao Teng, Limin Wang, Zhifeng Li, and Gangshan Wu. Target adaptive context aggregation for video scene graph generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13688–13697, 2021. 1, 2

[10] Shuang Wang, Lianli Gao, Xinyu Lyu, Yuyu Guo, Pengpeng Zeng, and Jingkuan Song. Dynamic scene graph generation via temporal prior inference. In *ACM International Conference on Multimedia (MM '22)*, 2022. 1

[11] Ji Zhang, Kevin J Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. Graphical contrastive losses for scene graph parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11535–11543, 2019. 1

[12] Linchao Zhu and Yi Yang. Inflated episodic memory with region self-attention for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4344–4353, 2020. 4