MeronymNet: A Unified Framework for Part and Category Controllable Generation of Objects

Rishabh Baghel
baghelrishabha@gmail.com

Abhishek Trivedi
abhishek.trivedi@research.iiit.ac.in

Tejas Ravichandran
venkatakrishnan164@gmail.com

Ravi Kiran Sarvadevabhatla
Centre for Visual Information Technology (CVIT)
IIIT Hyderabad, INDIA
ravi.kiran@iiit.ac.in

Abstract

We introduce MeronymNet, a novel hierarchical approach for controllable, part-based generation of multi-category objects using a single unified model. We adopt a guided coarse-to-fine strategy involving semantically conditioned generation of bounding box layouts, pixel-level part layouts and ultimately, the object depictions. We use Graph Convolutional Networks, Deep Recurrent Networks along with custom-designed Conditional Variational Autoencoders to enable flexible, diverse and category-aware generation of 2-D objects in a controlled manner. The performance scores and generations reflect MeronymNet's superior performance compared to scene generation baselines and ablative variants.

1. Introduction

Alongside recent successes for controllable scene generation [12, 18, 25, 19, 17, 1], generative models for individual objects have also found a degree of success [3, 10, 7, 22, 24, 4]. The bulk of object generation approaches specialize for a single category of objects with weakly aligned part configurations (e.g. faces [3, 10, 7]) and for objects with associated text description (e.g. birds [22, 24]). Although models such as as BigGAN [4] go beyond a single category, conditioning is possible only at a category level. Models which afford a more finer degree of control (e.g. part-level) for multiple object categories have not been explored. In addition, existing approaches induce contextual bias by necessarily generating background along with the object. As a result, the generated object cannot be utilized as a sprite (mask) for downstream tasks [5] (e.g. compositing the object within a larger scene). Re purposing controllable scene generative models, though seemingly reasonable, is not a viable alternative due to unique challenges in part-controlled object generation (as we shall show).

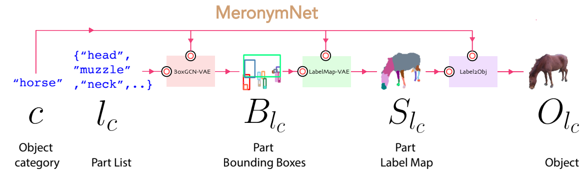

Figure 1: Given an object category c and an associated list of plausible object parts l_c, a part-labelled bounding box layout B_{l_c} is stochastically generated (Sec. 2.1). The box layout and category is used to guide the generation of a pixel-level label map S_{l_c} (Sec. 2.2). This layout map is translated into the final object depiction O_{l_c} (Sec. 2.3). Black-red circles indicate conditioning by object attributes during generation.

To address these shortcomings, we introduce MeronymNet[1], a novel unified part and category-aware generative model for object sprites (Fig. 1). Conditioned on a specified object category and an associated part list, a first-level generative model stochastically generates a part bounding box layout (Sec. 2.1). The category and generated part bounding boxes are used to guide a second-level model which generates semantic part maps (Sec. 2.2). Finally, the semantic part label maps are conditionally transformed into object depictions via the final level model (Sec. 2.3). Our unified approach enables diverse, part-controllable object generations across categories using a *single* model, without the necessity of multiple category-wise models (Sec. 4). Please visit our project page http://meronymnet.github.io/ for source code, generated samples and full version of this paper [2].

[1] Meronym: Linguistic term for expressing part-to-whole relationships.

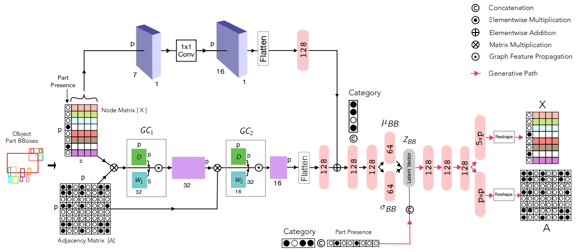

Figure 2: Architecture for BoxGCN–VAE which generates part-labelled bounding box object layouts (Sec. 2.1). The numbers within the rounded light pink rectangles indicate dimensionality of fully connected layers. Generative path is shown in dark pink.

2. MeronymNet

See Fig. 1. Suppose we wish to generate an object from category c ($1 \leqslant c \leqslant M$), with an associated maximal part list L_c. The category (c) and a list of parts $l_c \subseteq L_c$ is used to condition BoxGCN–VAE, the first-level generative model which stochastically generates part-labelled bounding boxes. The generated part-labeled box layout B_{l_c} and c are used to condition LabelMap–VAE which generates a category-specific per-pixel part-label map S_{l_c}. Finally, the label map is conditionally transformed into an RGB object depiction O_{l_c} via the final-level Label2obj model.

2.1. BoxGCN–VAE

Representing the part bounding box layout: We design BoxGCN–VAE (Fig. 2) as a Conditional VAE which learns to stochastically generate part-labelled bounding boxes. We model the object layout as an undirected graph. Let p be the maximum possible number of parts across all object categories, i.e. $\mathsf{p} = \max_c length(L_c), 1 \leqslant c \leqslant M$. X is a p $\times$ 5 feature matrix where each row r corresponds to a part. A binary value $p_r \in \{0, 1\}$ is used to record the presence or absence of the part in the first column. The next 4 columns represent bounding box coordinates. For categories with part count less than p and for absent parts, the rows are filled with 0s. The p $\times$ p binary matrix A encodes the connectivity relationships between the object parts in terms of part bounding box overlap. Thus, we obtain the object part bounding box representation $\mathbb{G} = (X, A)$.
Encoding the graph representation: The feature representation of graph $\mathbb{G}$ obtained from GCN is then mapped to the parameters of a diagonal Gaussian distribution $(\mu_{BB}(\mathbb{G}), \sigma_{BB}(\mathbb{G}))$, i.e. the approximate posterior. This mapping is guided via category-level conditioning (see Fig. 2). In addition, the mapping is also conditioned using skip connection features. These skip features are obtained via 1×1 convolution along the spatial dimension of bounding box sub-matrix of input $\mathbb{G}$ (see top part of Fig. 2). In addition to part-focused guidance for layout encoding, the skip-connection also helps avoid imploding gradients.

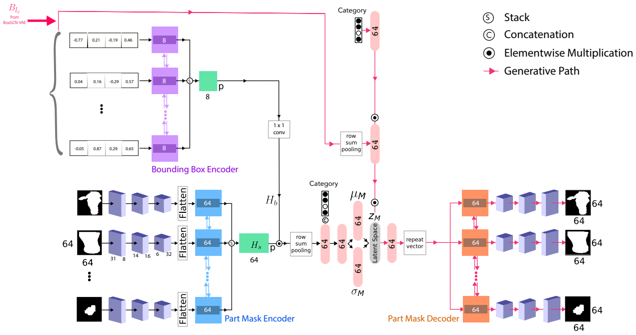

Figure 3: The architecture for LabelMap–VAE which generates per-pixel map for each part conditioned on object class and the part-labeled bounding box layout (Sec. 2.2). After generation, the bounding boxes are used to compose the object in terms of appropriately warped part masks. The pink arrows indicate the data flow for the generative model.

Reconstruction: The sampled latent variable z, conditioned using category and part presence variables (c, l_c), is mapped by the decoder to the components of $\mathbb{G} = (X, A)$. Denoting $X = [X_1 | X_{bb}]$, the binary part-presence vector X_1 is modelled as a factored multivariate Bernoulli distribution. To encourage accurate localization of part bounding boxes X_{bb}, we use two per-box instance-level losses: mean squared error $L_i^{MSE} = \sum_{j=1}^{4}(X_{bb}^i[j] - \hat{X}_{bb}^i[j])^2$ and Intersection-over-Union between the predicted $(\hat{X}_{bb})$ and ground-truth (X_{bb}) bounding boxes $L_i^{IoU} = -ln(IoU(X_{bb}^i, \hat{X}_{bb}^i))$ [23]. To impose additional structural constraints, we also use a pairwise MSE loss defined over distance between centers of bounding box pairs. Denoting the ground-truth Euclidean distance between centers of m-th and n-th bounding boxes as $d_{m,n}$, the pairwise loss is defined as $L_{m,n}^{MSE-c} = (d_{m,n} - \hat{d}_{m,n})^2$ where $\hat{d}_{m,n}$ refers to predicted between-center distance. For the adjacency matrix (A), we use binary cross-entropy $L_{m,n}^{BCE}$ as the per-element loss.

It is important to note that unlike scene graphs [21, 9], spatial relationships between nodes (parts) in our object graphs are not explicitly specified. This makes our part graph generation problem setting considerably harder compared to scene graph generation. Also note that the decoder architecture is considerably simpler compared to encoder. As our experimental results shall demonstrate (Sec. 3), the conditioning induced by category and part-presence, combined with the connectivity encoded in the latent representation z, turn out to be adequate for generating the object bounding box layouts despite the absence of graph unpooling layers in the decoder.

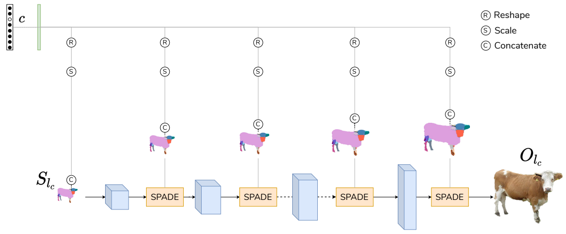

Figure 4: Architecture for Label2obj (Sec. 2.3) which translates part label map S_{l_c} to corresponding 2-D object depiction O_{l_c} conditioned on category c's embedding.

2.2. LabelMap–VAE

To generate dense per-part label maps, we design LabelMap-VAE as a Conditional VAE which learns to stochastically generate per-part spatial masks (Fig. 3). To guide mask generation in a category-aware and layout-aware manner, we use feature embeddings of object category c and bounding box layout B_{l_c} generated by BoxGCN–VAE (Sec. 2.1). During encoding, the binary mask for each part is resized to fixed dimensions. The per-part CNN-encoded feature representations of individual part masks are aggregated using a bi-directional Gated Recurrent Unit (GRU) (color-coded blue in Fig. 3). The per-part hidden-state representations from the unrolled GRU are stacked to form a $\mathsf{p} \times h_s$ representation H_s. This representation is modulated using a transformed representation of part bounding boxes H_b to induce bounding-box based conditioning. The result is pooled across rows and further gated using category information. In turn, the obtained feature representation is ultimately mapped to the parameter representations of a diagonal Gaussian distribution (μ_M, σ_M).

Generating the part label map: The decoder transforms the sampled latent variable z to a conditional data distribution $p_{\theta_m}(M|z, \alpha_{bb}, \alpha_c)$ over the sequence of label maps $M = \{m_1, m_2, \dots, m_p\}$ with θ_m representing parameters of the decoder network. α_c and α_{bb} respectively represent the conditioning induced by feature representations of object category c and the stochastically generated bounding box representation B_{l_c}. To obtain a one-hot representation of the object label map, each part mask is positioned within a $H \times W \times \mathsf{p}$ tensor where $H \times W$ represents the 2-D spatial dimensions of the object canvas. The part's index $k, 1 \leqslant k \leqslant \mathsf{p}$, is used to determine one-hot label encoding while the spatial geometry is obtained by scaling the mask according to the part's bounding box.

2.3. Label2obj

For translating part label maps generated by LabelMap–VAE (Sec. 2.2) to the final object depiction in a category-aware manner, we design Label2obj as a modified, conditioned variant of the approach by Park et. al. [17] (see Fig. 4). The one-hot representation of object category is transformed via an embedding layer. The embedding output is reshaped, scaled and concatenated depth-wise with appropriately warped label maps. The resultant features comprise one of the inputs to the SPADE blocks within the pipeline. Our modification incorporates a category-aware discriminator [16] which complements our category-conditioned generator.

3. Experimental Setup

Dataset: For our experiments, we use the PASCAL-Part dataset [6], containing 10,103 images across 20 object categories annotated with part labels at pixel level. We select the following 10 object categories: cow, bird, person, horse, sheep, cat, dog, airplane, bicycle, motorbike. The collection is characterized by a large diversity in appearance, viewpoints and associated part counts. The objects are normalized with respect to the minimum and maximum width across all images such that all objects are centered in a $[0, 1] \times [0, 1]$ canvas. We use 75% of the images for training, 15% for validation and the remaining 10% for evaluation.

Baseline Generative Models: Since no baseline models exist currently for direct comparison, we designed and implemented the baselines ourselves – see Fig. 6 for a visual illustration of baselines and component configurations. We modified existing scene layout generation approach [14] to generate part layouts. In some cases, we modified existing scene generation approaches having layouts as the starting point [1, 19, 15] to generate objects. We also included modified variants of two existing part-based object generative models (3-D objects [20], faces [7]). To evaluate individual components from MeronymNet, we also designed hybrid baselines with MeronymNet components (BoxGCN–VAE, LabelMapVAE, Label2obj) included. We incorporated object category, part-list based conditioning in each baseline to ensure fair and consistent comparison.

Evaluation Protocol: For each model (including MeronymNet), we generate 100 objects for each category using the per-category part lists in the test set. We use Frechet Inception Distance [11] (FID) as a quantitative measure of generation quality. The smaller the FID, better the quality. For each model, we report FID for each category's generations separately and overall average FID.

4. Results

Quantitative results: As Table 1 shows, Meronymnet outperforms all other baselines. The quality of part box layouts from BoxGCN–VAE is better than those produced using LSTM-based LayoutVAE [14] (compare rows $1, 7$, rows $3, 5$ and $4, 6$, also see Fig. 6). Note that the modified version of PQNet [20] (row 10) also relies on a sequential (GRU) model. The benefit of modelling object layout dis-

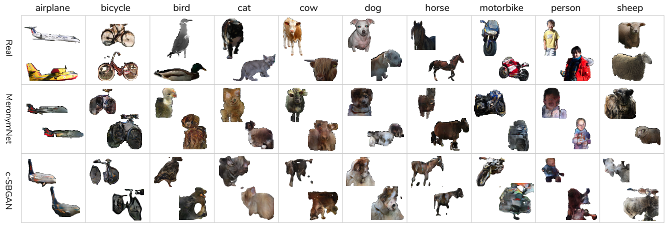

Figure 5: Sample object generations from MeronymNet (second row), c-SBGAN (third row) and objects from test set (top row).

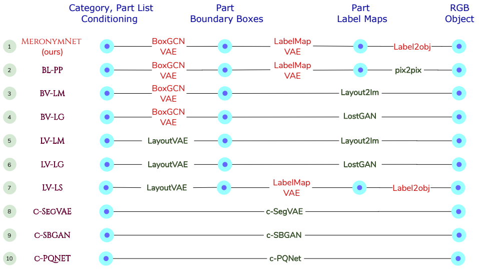

Figure 6: An illustration of baselines (rows) used for comparative evaluation against MeronymNet (top row). The blue concentric circles in columns denote inputs and outputs of components. Components reused from MeronymNet are shown in orange text and those based on our modifications to existing scene-based models are in green text.

tributions via the more natural graph-based representations (Sec. 2.1) is evident. The results also highlight the importance of our three-stage generative pipeline. In particular, MeronymNet distinctly outperforms approaches which generate objects directly from bounding box layouts [25, 19] (compare rows $1, 3, 4$). Also, the relatively higher quality of MeronymNet's part label maps ensures better performance compared to other label map translation-based approaches [1, 7, 13] (compare rows $1, 2, 8, 9$). In particular, note that our modified variants of some models [1, 7] (rows $8, 9$) employ a SPADE-based approach [17] for the final label-to-image stage.

The quality of sample generations from MeronymNet (second row in Fig. 5) is comparable to unseen PASCAL-Parts test images with same part list (top row). The same figure also shows sample generations from c-SBGAN, the next best performing baseline (last row). The visual quality of MeronymNet's generations is comparatively better, especially for categories containing many parts (e.g. person, cow, horse). Across the dataset, objects tend to have a large range in their part-level morphology (area, aspect ratio, count) which poses a challenge to image-level label map generation approaches, including c-SBGAN. In contrast, our design choices - generating all part label masks at same resolution (Fig. 3), decoupling bounding box and label map geometry – all help address the challenges better [8].

		Frechet Inception Distance (FID) ↓										
ID	**Model**	**Overall**	plane	bicycle	bird	cat	cow	dog	horse	m.bike	person	sheep
1	**MERONYMNET**	**331.6**	341.9	384.1	340.3	247.3	334.1	319.8	364.5	326.0	368.5	289.6
2	BL-PP [13]	387.2	408.1	413.0	363.5	315.9	390.7	385.7	390.0	454.5	408.8	341.6
3	BV-LM [25]	397.3	415.6	437.8	346.4	406.8	371.5	390.4	370.4	421.9	420.6	391.8
4	BV-LG [19]	368.9	421.7	406.2	335.3	346.0	338.9	336.0	337.5	440.6	393.1	334.0
5	LV-LM [14, 25]	405.6	419.2	394.5	374.5	424.0	386.4	411.8	387.6	415.0	456.0	387.3
6	LV-LG [14, 19]	420.8	426.2	438.2	389.2	428.6	390.2	418.3	402.6	468.8	438.3	406.9
7	LV-LS [14]	373.8	361.3	374.2	351.2	376.5	383.6	366.7	375.4	385.0	392.1	372.3
8	C-SEGVAE [7]	378.2	355.7	367.1	345.6	406.9	376.4	402.7	372.9	379.5	408.3	367.4
9	C-SBGAN [1]	342.4	287.7	326.3	321.0	346.5	350.3	363.9	356.8	357.6	379.4	334.8
10	C-PQNET [20]	380.6	353.3	335.1	331.6	410.0	392.7	385.7	400.2	368.7	449.3	379.4

Table 1: Category-wise and overall FID for different baselines and MeronymNet. References to scene-generation components are provided alongside each baseline's name (column 2). Refer to Fig 6 for a visual representation of baseline components.

It is somewhat tempting to assume that parts are to objects what objects are to scenes, compositionally speaking. However, the analogy does not hold well in the generative setting. This is evident from our results with various scene generation models as baseline components (Table 1). The structural and photometric constraints for objects are stricter compared to scenes. In MeronymNet, these constraints are addressed by incorporating compositional structure and semantic guidance in a considered, coarse-to-fine manner which enables better generations and performance relative to baselines.

Qualitative results: MeronymNet's generations conditoned on object category and associated part lists from test set can be viewed in full paper [2]. The results demonstrate the MeronymNet's ability in generating diverse object maps for multiple object categories. The results also convey our model's ability to accommodate a large range of parts within and across object categories.

5. Conclusion

Our novel generative model MeronymNet generates diverse looking RGB object sprites in a part-aware manner across multiple categories using a single unified architecture. The strict and implicit constraints between object parts, variety in layouts and extreme part articulations, generally make multi-category object generation a challenging problem. Through our design choices involving GCNs, CVAEs, RNNs and guidance using object attribute conditioning, we show that these issues can be successfully tackled using a single unified model. Our evaluation establishes the quantitative and qualitative superiority of MeronymNet, overall and at individual component level. The advantages of our hierarchical setup include efficient processing and scaling with inclusion of additional object categories in future. Going forward, we intend to explore modifications towards improved quality, diversity and degree of controllability. We also intend to explore the feasibility of our unified model for multi-category 3-D object generation.

References

[1] Samaneh Azadi, Michael Tschannen, Eric Tzeng, Sylvain Gelly, Trevor Darrell, and Mario Lucic. Semantic bottleneck scene generation. *ArXiv*, abs/1911.11357, 2019. 1, 3, 4

[2] Rishabh Baghel, Abhishek Trivedi, Tejas Ravichandran, and Ravi Kiran Sarvadevabhatla. Meronymnet: A hierarchical model for unified and controllable multi-category object generation. In *ACM International Conference on Multimedia (ACMMM)*, 2021. 1, 4

[3] Zachary Bessinger and Nathan Jacobs. A generative model of worldwide facial appearance. In *WACV*, pages 1569–1578. IEEE, 2019. 1

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018. 1

[5] Megan Charity, Ahmed Khalifa, and Julian Togelius. Baba is yall: Collaborative mixed-initiative level design. In *2020 IEEE Conference on Games (CoG)*, pages 542–549. IEEE, 2020. 1

[6] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1971–1978, 2014. 3

[7] Yen-Chi Cheng, Hsin-Ying Lee, Min Sun, and Ming-Hsuan Yang. Controllable image synthesis via segvae. In *European Conference on Computer Vision*, 2020. 1, 3, 4

[8] Apoorva Dornadula, Austin Narcomey, Ranjay Krishna, Michael Bernstein, and Fei-Fei Li. Visual relationships as functions: Enabling few-shot scene graph prediction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 4

[9] Jiuxiang Gu, Handong Zhao, Zhe Lin, Sheng Li, Jianfei Cai, and Mingyang Ling. Scene graph generation with external knowledge and image reconstruction. In *CVPR*, pages 1969–1978, 2019. 2

[10] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Trans. on Image Processing*, 2019. 1

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017. 3

[12] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *CVPR*, pages 7986–7994, 2018. 1

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 4

[14] Akash Abdu Jyothi, Thibaut Durand, Jiawei He, Leonid Sigal, and Greg Mori. LayoutVAE: Stochastic scene layout generation from a label set. In *ICCV*, October 2019. 3, 4

[15] Jianan Li, Jimei Yang, Aaron Hertzmann, Jianming Zhang, and Tingfa Xu. LayoutGAN: Generating graphic layouts with wireframe discriminators. *ICLR*, 2019. 3

[16] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651, 2017. 3

[17] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 1, 3, 4

[18] Daniel Ritchie, Kai Wang, and Yu-an Lin. Fast and flexible indoor scene synthesis via deep convolutional generative models. In *CVPR*, pages 6182–6190, 2019. 1

[19] Wei Sun and Tianfu Wu. Image synthesis from reconfigurable layout and style. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 3, 4

[20] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3, 4

[21] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for scene graph generation. In *ECCV*, pages 670–685, 2018. 2

[22] Guojun Yin, Bin Liu, Lu Sheng, Nenghai Yu, Xiaogang Wang, and Jing Shao. Semantics disentangling for text-to-image generation. In *CVPR*, pages 2327–2336, 2019. 1

[23] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *ACMMM*, page 516520, 2016. 2

[24] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE TPAMI*, 41(8):1947–1962, 2018. 1

[25] Bo Zhao, Lili Meng, Weidong Yin, and Leonid Sigal. Image generation from layout. In *CVPR*, 2019. 1, 4