Записки научных
семинаров ПОМИ
Том 530, 2023 г.

D. Shelepneva, K. Arkhipenko

REALISTIC ADVERSARIAL ATTACKS ON OBJECT DETECTORS USING GENERATIVE MODELS

Abstract. An important limitation of existing adversarial attacks on real-world object detectors lies in their threat model: adversarial patch-based methods often produce suspicious images while image generation approaches do not restrict the attacker's capabilities of modifying the original scene. We design a threat model where the attacker modifies individual image segments and is required to produce *realistic* images. We also develop and evaluate a white-box attack that utilizes generative adversarial nets and diffusion models as a generator of malicious images. Our attack is able to produce high-fidelity images as measured by the Fréchet inception distance (FID) and reduces the mAP of Faster R-CNN model by >0.2 on Cityscapes and COCO-Stuff datasets. A PyTorch implementation of our attack is available at `https://github.com/DariaShel/gan-attack`.

§1. Introduction

Deep learning models are being implemented in an increasing number of industry-level software systems. Computer vision is a field where deep neural nets have been showing the best results since 2012 [17], powering applications such as medical imaging [19], self-driving cars [10], and visual recommender systems [15]. However, the curse of dimensionality and overparameterization of these models make them vulnerable to *adversarial examples*, where malicious and stealthy input perturbations can reduce the model accuracy down to zero.

Test-time adversarial attacks on computer vision have been studied since 2013: the landmark paper [29] proposed imperceptible, L^p-norm bounded perturbations which manipulate model predictions after being added to the testing images. This type of adversarial perturbations remains the most studied in the literature thanks to its stealthiness, ease

Key words and phrases: adversarial examples, object detectors, generative adversarial networks, diffusion models.

of implementation in a white-box scenario [9] and transferability of adversarial examples between different models trained over the same dataset or even over different datasets [21].

While very effective against undefended models, L^p-norm bounded adversarial examples can barely survive input transformations [11] deployed as a defense mechanism in a black-box scenario or occurring naturally as a result of shooting a (potentially malicious) scene with a camera. Therefore, the application systems of our interest — *object detectors in the physical world* — require *other* types of perturbations to benchmark against.

Recent work on fooling real-world object detectors [13, 32] builds upon the *adversarial patch* method introduced in [3]. This method generates an *unbounded* perturbation and applies it to a limited portion of a testing image. The high magnitude of adversarial patches makes them robust to input transformations *but* also makes the attacked example more suspicious to both human eye and defense mechanisms: such examples can be detected by uncanny patterns not fitting well into the original images. This issue has already been addressed in [13] by employing *generative models*, namely StyleGAN2 [16] and BigGAN [2] networks, for generating more natural-looking adversarial patches.

Besides adversarial patch attacks, there are notable attack methods that utilize generative models to generate *whole* images. In the earlier work [28], the attacked images are obtained as a result of optimization over the latent space of the generative model. A more promising approach from our point of view is to *train* the model to generate malicious images instead of latent space optimization; this approach is taken in [23] for adversarial image editing. Another property of [23] that distinguishes it from [28] and [13] is *conditioning* the generation process on the input scene; both properties contribute to the improved naturalness and fidelity of the attacked images.

In order to develop a practical attack on real-world object detectors, we decided to combine the advantages of adversarial patches and conditional image generation. Our **primary contributions** are the following.

(1) We propose a *threat model* designed specifically for object detectors in the physical world. In this threat model, the attacker is allowed to affect only a limited portion of the image, similar to adversarial patch attacks. *Different* from adversarial patches, we permit modifying individual *segments* of the input image. Moreover, the attacker is forced to produce *realistic*, high-fidelity images in order to evade human eye and/or defense mechanisms. We use

the popular Fréchet inception distance (FID) [12] as a stealthiness metric for the attacked images.

(2) According to this threat model, we develop a white-box adversarial attack based on generative models and applicable to *any* differentiable object detector (e.g., Faster R-CNN [24]). The attack employs pix2pixHD [31] or PITI [30] as a generator of malicious images and thus is conditioned on *semantic label maps* of the original scenes.

§2. Related work

2.0.0.1. Generative models. Generative models learn to generate realistic images and use quality metrics consistent with human perception.

A generative adversarial network (GAN) consists of a generator G and a discriminator D. The purpose of the generator is to make the most realistic images, and the discriminator is to differentiate the generated images from the real ones. Therefore, the training of generative models is based on a minimax game [8]:

$$\min_G \max_D \mathbb{E}_x[\log D(x)] + \mathbb{E}_z[1 - \log D(G(z))] \tag{1}$$

Here z is the random noise vector, and x is the original image.

There exist many types of GANs, e.g., with an auxiliary classifier [22] or with an additional gradient penalty loss function aimed at improving the stability of training [1].

We cast our problem of scene-conditioned adversarial attacks on object detectors as an *image-to-image translation* task which was introduced in [14]. The variation of GAN proposed in this work is capable of generating images in high resolution, but with low detail. The issue was solved in [31] which was taken as the basis for our work. To achieve high resolution and detail, pix2pixHD uses two generators, three discriminators, as well as an encoder and VGG-loss. The model accepts *semantic label maps* as input and generates an image so that objects are located in the specified segments.

2.0.0.2. Adversarial attacks. While there exist a number of adversarial attacks employing generative adversarial networks and style transfer [7, 28], these works have their limitations. In [28], the proximity of the generated image to the original one in the latent space is required; this greatly limits the possibilities for attack. In addition, the image is completely replaced with the generated one, so the attack becomes more noticeable. In [7], the method transfers the style of the selected *style image* to the one being attacked. However, the success and stealthiness of an attack highly depend on the choice of the style image, and there is yet no algorithm to automatically choose one. We also believe that preserving the *content* of the attacked image portions is an excessive restriction which limits attack possibilities in many cases.

Despite the limitations of [7], it provided us with an idea of replacing only a few segments of the input image. Inspired by pix2pixHD [31], we utilize ground-truth segmentation label maps to select the segments for attack. Unlike [7], we allow *arbitrary* changes to these segments but require high fidelity of the attacked image in return. In our method, the pix2pixHD model is trained so that the generated segments remain visually realistic, but the victim model is mistaken on them. The method is described in more detail in Section 3.

2.0.0.3. Diffusion models. Recently, diffusion models have become very popular [6, 25–27]. Diffusion models can solve a wide range of tasks, including text-to-image, super-resolution, and others. Since we use image-to-image translation as a proxy problem for our attack, we took PITI [30] as an alternative to pix2pixHD. While the authors of [30] train their model in multiple steps, we only employ PITI for increasing the resolution.

§3. Method description

3.1. Attack on image segments with GAN. Our attack is based on a pre-trained model capable of generating an image using a segmentation mask [31]. We train the model with the addition of several targets to the original loss function $\mathcal{L}_0$ [31]:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{(s,x)}[\log D(s, x)] + \mathbb{E}_s[1 - \log D(s, G(s))] \tag{2}$$

$$\mathcal{L}_{VGG}(G) = \lambda \sum_i \frac{1}{M_i} ||F^{(i)}(x) - F^{(i)}(G(s))||_1 \tag{3}$$

$$\mathcal{L}_0 = \min_G \Big(\sum_{k=1,2,3} \mathcal{L}_{GAN}(G, D_k) + \mathcal{L}_{VGG}(G) \Big) \tag{4}$$

Here G and D_1, D_2, D_3 are generator and discriminators from the architecture from [31] respectively, x is the original image, and s is the semantic (segmentation) label map of this image. In equation (3), $\lambda = 10$

controls the importance of $\mathcal{L}_{VGG}$, and $F^{(i)}$ denotes the i-th layer of the VGG network with M_i elements.

$\mathcal{L}_0$ is the initial loss function for training the generative model. In order for the model to generate an attack, we added a few more expressions to this function:

$$\mathcal{L}_{overlay}(G, D) = \mathbb{E}_{(s,x)}[1 - \log D(s, overlay(T, x, G(s)))] \tag{5}$$

$$\mathcal{L}_{adv} = \frac{1}{N}\sum_{i=1}^{N} BCELoss(C_i, 0) \tag{6}$$

The *overlay* function replaces the selected T segments of the original image with generated ones. T is the set of labels in the selected segments. We want the resulting image to be considered real by the discriminator as well. In equation (6), C_i is the confidence of the victim detector in detecting the i-th object in the image. Accordingly, we want the confidence to be low, so the detector will be unable to detect objects.

The final loss function for our attack will look as follows:

$$\begin{aligned}\mathcal{L} = \min_G \Bigg(\Big(\max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{GAN}(G, D_k) \Big) + \mathcal{L}_{VGG}(G) \\ + \sum_{k=1,2,3} \mathcal{L}_{overlay}(G, D_k) + \mathcal{L}_{adv} \Bigg)\end{aligned} \tag{7}$$

The full attack procedure is presented in Algorithm 12.

3.2. Attack on image segments with diffusion. In this variation of our attack, we take PITI [30] as the basis. The authors of [30] train a diffusion model in multiple steps with the segmentation map of the image as input. After that, a model is trained that improves the resolution of the image generated by the diffusion model. In our attack, we will further train this model, which is essentially a GAN, using the loss function from equation (7).

The attack scheme is essentially no different from equation (12), since we do not change the diffusion model but only train the GAN to (maliciously) increase the resolution.

§4. Experiments

4.1. Datasets and models. The experiments were carried out on Cityscapes [5] and COCO-stuff [4] datasets. The results were averaged

Algorithm 1: Training loop for generating adversarial image segments

```
Data: $x$ – Original image,
$s$ – Image segmentation map,
$T$ – The set of labels for the segments to be replaced,
$N$ – Number of loop iterations per image,
$netG$ – Pre-trained generators,
$netD$ – Pre-trained discriminators,
$det$ – The victim detector,
$overlay(T, x, x')$ – Function that replaces the selected $T$ segments
of the original image with the generated ones,
$lvgg(x, x')$ – Function for calculating $\mathcal{L}_{VGG}$ loss (eq. 3),
$getConfidence(det(x))$ – Function that gets confidence $C$ from the
detector outputs,
$compute\_Dloss(real, fake)$ – A function that computes the
discriminator loss function,
$compute\_Gloss(fake, attacked)$ – A function for the generator loss
function,
$compute\_adv\_loss(C)$ – A function for the $\mathcal{L}_{adv}$ loss (6)
Output: $\hat{x}$ – The attacked image

for $iter = 1$ to $N$ do
    $x' \leftarrow netG(s)$;
    $real = netD(x)$;
    $fake = netD(x')$;
    $lossD = compute\_Dloss(real, fake)$;
    $\hat{x} \leftarrow overlay(T, x, x')$;
    $attacked = netD(\hat{x})$;
    $C \leftarrow getConfidence(det(\hat{x}))$;
    $lossG = compute\_Gloss(fake, attacked) + lvgg(x, x') +$
      $compute\_adv\_loss(C)$;
    # Optimization step
    $step(netD, lossD)$;
    $step(netG, lossG)$;
end
```

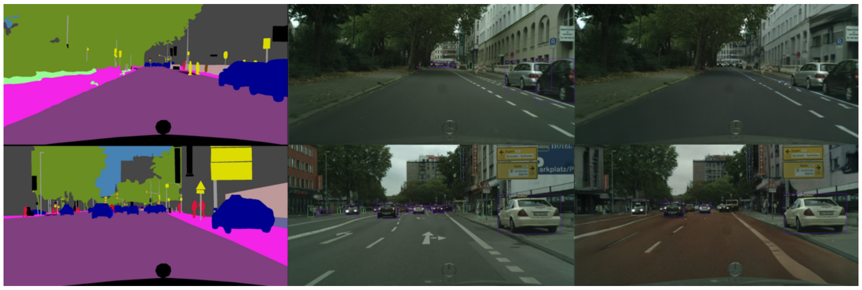

Figure 1. GAN attack. From left to right: segmentation map; the result of object detection in the original and attacked image, respectively.

and are given in Tables 1 and 2. Faster R-CNN [24] was taken as the victim detector.

4.2. Metrics. mAP was chosen as the success metric of the attack.

The stealthiness metric of our attack is the Fréchet inception distance (FID) [12]. FID compares the distribution of 2048-dimensional activations of the Inception v3 `pool3` layer for generated and real images. Two Gaussian distributions are estimated, and the FID value is calculated as the Fréchet distance between these distributions. The lower the FID value, the higher the image quality:

$$FID = |\mu - \mu'| + \mathrm{Tr}(\Sigma + \Sigma' - 2(\Sigma\Sigma')^{\frac{1}{2}}). \tag{8}$$

Here $\mathcal{N}(\mu, \Sigma)$ is the multivariate normal distribution estimated from Inception v3 features calculated on real images, and $\mathcal{N}(\mu', \Sigma')$ is the multivariate normal distribution estimated on the generated (*attacked* in our case) images.

4.3. GAN. In our experiment with pix2pixHD model as an attack generator, we used the Cityscapes dataset.

In the original image, we replace the segments representing *road*, *buildings* and *sky* with generated ones. They were chosen as they cover a sufficiently large portion of the image, which enables a more successful attack. But at the same time, these segments are essentially background, so they do not catch the eye. In addition, these segments are not objects that the detector is trained to detect, so our attack turns out to be *"fair"*.

Table 1. The mAP on attacked images is almost 2 times lower than on the original ones, which indicates a successful attack. The FID on images where only a few segments are replaced with generated ones is lower than on fully generated images. This means that partially generated images are more realistic.

mAP		FID			
original	attacked	fully generated VS original		segments generated VS original	
		w/attack	no attack	w/attack	no attack
0.6709	**0.3854**	104.879	74.155	**74.702**	53.867

After running the experiments, the following results were obtained. The detector found much fewer objects in the attacked image than in the original one (see Figure 1).

Experiments were carried out on 500 test images, and the averaged metrics are listed in Table 1. We also provide FID scores for *benign* images generated by pix2pixHD in “no attack” columns. One can see that while the attacked images are less realistic than benign ones, the difference in FID is not very big.

4.4. Diffusion. For our experiments with the diffusion model, we have used the COCO-stuff dataset.

In the COCO-stuff dataset, there are no segments that occur in all images, unlike the Cityscapes dataset. Therefore, in the original image we replace a single segment that occupies the largest area with a generated one.

As in the GAN-based attack, the detector recognized much fewer objects in the attacked image than in the original one (see Figure 2).

Experiments were conducted on 550 testing images, and averaged metrics are listed in Table 2.

§5. Conclusion

In this work, we have proposed a threat model and an adversarial attack targeted at real-world object detection systems. Our results on subsets of Cityscapes and COCO 2017 have the following implications.

(1) It *is* possible to significantly reduce the quality of object detectors by modifying individual segments of the original images with

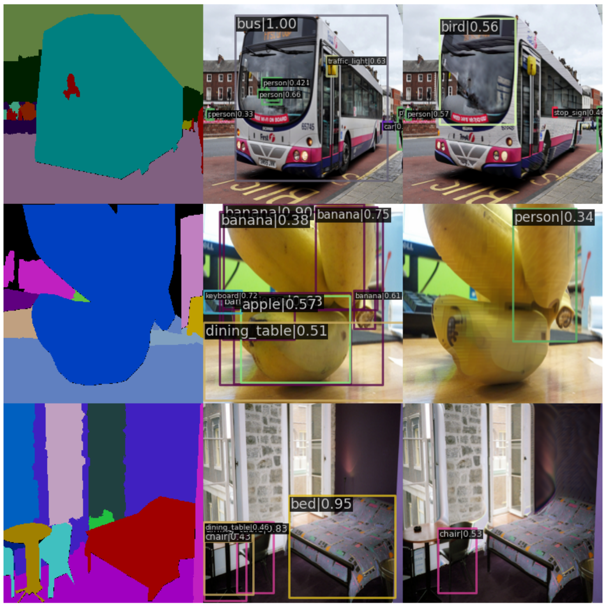

Figure 2. Example of the attack using a diffusion model. From left to right: segmentation map; the result of object detection on the original and attacked image, respectively.

Table 2. The results of mAP evaluation are similar to those obtained in the experiment with GAN, which means that the attack employing the diffusion model is as successful as with GAN. However, the FID of images generated using the diffusion model is higher, which indicates that images generated using the diffusion model are less realistic. It can be assumed that this is due to the artifacts that occur when using diffusion.

mAP		FID			
original	attacked	fully generated VS original		segments generated VS original	
		w/attack	no attack	w/attack	no attack
0.6056	**0.3621**	182.413	158.062	**139.058**	103.694

the help of GANs and diffusion models. The generated malicious images are high-fidelity and therefore our attack is harder to recognize compared to adversarial patch attacks.

(2) Fréchet inception distance alone *cannot* serve as a reliable indicator of attack presence in the testing images, and defenses against our attack require *other* criteria. Note that we are aware of the vulnerability of FID itself to attacks (see e.g. [18]); however, we confirm that one does *not* have to optimize FID in order to obtain high FID scores for the attacked images in our method.

However, our present research has several **limitations** that are left for our future work.

(1) We have not yet tested our attack for robustness to input transformations [11] such as cropping-rescaling. Since these transformations are common in the physical world, our attack needs further improvement so that it can be used as a fair robustness benchmark for real systems of interest.

(2) Our immediate goal is an extensive comparison of our attack with state-of-the-art adversarial patch-based attacks including [13]. While we can already judge the "naturalness" of perturbed images by looking at them, automated evaluation is still needed to confirm the usefulness of our approach. This comparison should also be carried out in the presence of *defense* mechanisms, e.g. [20].

(3) For practical use of our attack, we need to investigate the transferability of perturbations generated by the attack to other scenes and other models, including *black-box* ones. As real-world object detection systems are typically black boxes, a more fair benchmark (compared to white-box evaluation) would be to assess the success rate of attack transfer.

References

1. M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein GAN*, ArXiv **abs/1701.07875** (2017).
2. A. Brock, J. Donahue, and K. Simonyan, *Large scale GAN training for high fidelity natural image synthesis*, ArXiv **abs/1809.11096** (2018).
3. T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, *Adversarial patch*, ArXiv **abs/1712.09665** (2017).
4. H. Caesar, J. R. R. Uijlings, and V. Ferrari, *Coco-stuff: Thing and stuff classes in context*, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2016), 1209–1218.

5. M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, *The cityscapes dataset for semantic urban scene understanding*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016), 3213–3223.
6. P. Dhariwal and A. Nichol, *Diffusion models beat gans on image synthesis*, ArXiv **abs/2105.05233** (2021).
7. R. Duan, X. Ma, Y. Wang, J. Bailey, A. K. Qin, and Y. Yang, *Adversarial camouflage: Hiding physical-world attacks with natural styles*, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020), 997–1005.
8. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, *Generative adversarial nets*, NIPS, 2014.
9. I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and harnessing adversarial examples*, CoRR **abs/1412.6572** (2014).
10. S. M. Grigorescu, B. Trasnea, T. T. Cocias, and G. Macesanu, *A survey of deep learning techniques for autonomous driving*, Journal of Field Robotics **37** (2019), 362 – 386.
11. C. Guo, M. Rana, M. Cissé, and L. van der Maaten, *Countering adversarial images using input transformations*, ArXiv **abs/1711.00117** (2018).
12. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, *GANs trained by a two time-scale update rule converge to a local Nash equilibrium*, NIPS, 2017.
13. Y. Hu, J.-C. Chen, B.-H. Kung, K.-L. Hua, and D. S. Tan, *Naturalistic physical adversarial patch for object detectors*, 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021), 7828–7837.
14. P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-image translation with conditional adversarial networks*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016), 5967–5976.
15. W.-C. Kang, C. Fang, Z. Wang, and J. McAuley, *Visually-aware fashion recommendation and design with generative image models*, 2017 IEEE International Conference on Data Mining (ICDM) (2017), 207–216.
16. T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, *Analyzing and improving the image quality of StyleGAN*, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019), 8107–8116.

17. A. Krizhevsky, I. Sutskever, and G. Hinton, *ImageNet classification with deep convolutional neural networks*, Neural Information Processing Systems **25** (2012).
18. T. Kynkääniemi, T. Karras, M. Aittala, T. Aila, and J. Lehtinen, *The role of ImageNet classes in Fréchet inception distance*, Proc. ICLR, 2023.
19. G. J. S. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. van der Laak, B. van Ginneken, and C. I. Sánchez, *A survey on deep learning in medical image analysis*, Medical image analysis **42** (2017), 60–88.
20. J. Liu, A. Levine, C. P. Lau, R. Chellappa, and S. Feizi, *Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14973–14982.
21. Y. Liu, X. Chen, C. Liu, and D. X. Song, *Delving into transferable adversarial examples and black-box attacks*, ArXiv **abs/1611.02770** (2016).
22. A. Odena, C. Olah, and J. Shlens, *Conditional image synthesis with auxiliary classifier GANs*, International Conference on Machine Learning, 2016.
23. H. Qiu, C. Xiao, L. Yang, X. Yan, H. Lee, and B. Li, *SemanticAdv: Generating adversarial examples via attribute-conditional image editing*, ArXiv **abs/1906.07927** (2019).
24. S. Ren, K. He, R. B. Girshick, and J. Sun, *Faster R-CNN: Towards real-time object detection with region proposal networks*, IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2015), 1137–1149.
25. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, *High-resolution image synthesis with latent diffusion models*, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021), 10674–10685.
26. C. Saharia, W. Chan, H. Chang, C. A. Lee, J. Ho, T. Salimans, D. J. Fleet, and M. Norouzi, *Palette: Image-to-image diffusion models*, ACM SIGGRAPH 2022 Conference Proceedings (2021).
27. J. N. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, *Deep unsupervised learning using nonequilibrium thermodynamics*, ArXiv **abs/1503.03585** (2015).
28. Y. Song, R. Shu, N. Kushman, and S. Ermon, *Constructing unrestricted adversarial examples with generative models*, Neural Information Processing Systems, 2018.
29. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, *Intriguing properties of neural networks*, CoRR **abs/1312.6199** (2013).
30. T. Wang, T. Zhang, B. Zhang, H. Ouyang, D. Chen, Q. Chen, and F. Wen, *Pretraining is all you need for image-to-image translation.*

31. T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, *High-resolution image synthesis and semantic manipulation with conditional GANs*, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2017), 8798–8807.
32. Z. Wu, S.-N. Lim, L. S. Davis, and T. Goldstein, *Making an invisibility cloak: Real world adversarial attacks on object detectors*, European Conference on Computer Vision, 2019.

Ivannikov Institute
for System Programming of the RAS
E-mail: d-d-sh@ispras.ru
E-mail: arkhipenko@ispras.ru

Поступило 6 сентября 2023 г.