Digital twins of physical printing-imaging channel

Yury Belousov, Brian Pulfer, Roman Chaban, Joakim Tutt, Olga Taran, Taras Holotyak and Slava Voloshynovskiy
Department of Computer Science, University of Geneva, Switzerland
{yury.belousov, brian.pulfer, roman.chaban, joakim.tutt, olga.taran, taras.holotyak, svolos}@unige.ch

***Abstract*—In this paper, we address the problem of modeling a printing-imaging channel built on a machine learning approach a.k.a. *digital twin* for anti-counterfeiting applications based on copy detection patterns (CDP). The digital twin is formulated on an information-theoretic framework called *Turbo* that uses variational approximations of mutual information developed for both encoder and decoder in a two-directional information passage. The proposed model generalizes several state-of-the-art architectures such as adversarial autoencoder (AAE) [1], CycleGAN [2] and adversarial latent space auto-encoder (ALAE) [3]. This model can be applied to any type of printing and imaging and it only requires training data consisting of digital templates or artworks that are sent to a printing device and data acquired by an imaging device. Moreover, these data can be paired, unpaired or hybrid paired-unpaired which makes the proposed architecture very flexible and scalable to many practical setups. We demonstrate the impact of various architectural factors, metrics and discriminators on the overall system performance in the task of generation/prediction of printed CDP from their digital counterparts and vice versa. We also compare the proposed system with several state-of-the-art methods used for image-to-image translation applications. The code and extended results of the simulation are publicly available[1].**

***Index Terms*—Copy detection patterns, machine learning, digital twin, information theory, variational approximation.**

I. Introduction

In recent years copy detection patterns (CDP) [4], [5] attracted a lot of attention as an anti-counterfeiting technology. At the same time, a lot of research was done to investigate the different factors impacting the authentication accuracy of CDP. However, the production of datasets of real CDP is a costly and timely process. It requires the printing and acquisition of original CDP and the production and acquisition of fakes, preferably on equipment close to the industrial one. The factors of cost, time and needed domain knowledge considerably constrain the study of the anti-counterfeiting aspects of CDP.

The lack of accurate mathematical models of complex production and acquisition systems leads to a need to collect a huge amount of data for each particular case, reduces the system scalability to new products, production technologies and imaging devices, and makes the optimization process difficult, time-consuming and expensive. Moreover, the optimization of this system is complicated by a non-differentiable nature of existing models and their non-stochastic nature that does not reflect real practical situations.

S. Voloshynovskiy is a corresponding author.

This research was partially funded by the Swiss National Science Foundation SNF No. 200021_182063.

[1]https://gitlab.unige.ch/sip-group/digital-twin

The knowledge of the physical printing-imaging channel plays a very important role in anti-counterfeiting systems and is crucial for both the defender and the attacker. On the side of the defender, the knowledge of a model for this channel can (a) enable the overall optimisation of the whole authentication system by end-to-end training of encoders, decoders and decision modules, (b) simulate and predict the intra-class variabilities and (c) help generate synthetic samples of both originals and fakes that can be used to efficiently train decision module's classifiers.

The attacker can also benefit from such a model by (a) optimising the estimation of digital templates from the physical samples in the scope of copy attacks and (b) developing adversarial samples for the physical domain.

At the same time, the design of *digital twins* of printing-imaging channels is not a trivial task. To simplify it somehow, one can consider printing and imaging systems separately.

Besides some works [6], [7] addressing the physics of specific production systems, there is no generalized theory on how to model even straightforward printing systems characterized by a high level of stochasticity and nonlinearity. The printing process model of each printing technology, such as off-set, digital off-set, inkjet or flexo, representing the most significant interest for practical applications, is very complex and domain-specific. Moreover, such a model should consider not only hardware but also software particularities of drivers that significantly impact the printed outcome. Altogether, it requires a lot of domain-specific know-how and makes the model development for each printing system very time-consuming. Furthermore, the validation of the model is also expensive and might require tuning many parameters.

Not less important is the modelling of the acquisition/imaging process. Besides some remarkable exceptions [8], [9] that present the models of noise in the CCD and CMOS imaging devices and practical methodologies of their validation, the simulation of the interaction between the incident light and reflecting object surface is not a trivial task [10]. The imaging device hardware components and drivers' settings such as type of sensor, resolution of sensors, optics, ISO, shutter time, denoising, white color balancing, compression, etc., greatly impact the output image features. Finally, similarly to the models of production systems, there is no guarantee that the imaging model will be interpretable and differentiable and thus suitable for the envisioned ML tasks.

In this paper, we aim at addressing these challenges and shortcomings by following machine-learning framework and

Fig. 1. Turbo digital twin system: direct and reverse paths.

introducing a concept of digital twins of complex and unknown physical systems. More specifically, we propose a digital twin system that simulates the entire chain from the digital template t to the acquired image y that might represent both the original data x and fakes $\hat{x}$. The proposed system is based on an auto-encoder (AE) structure. To our best knowledge, there is no such framework for the addressed printing-imaging problem.

The framework of digital twins might be used as a simulator of complex physical printing-imaging systems for:

- creation of differentiable models leading to the investigation of unexplored adversarial attacks in the physical world;
- generation of synthetic samples in order to train a supervised classifier for originals and fakes even when no fakes are known in advance by using synthetic samples as fakes;
- creation of augmentations for self-supervised learning (SSL) methods;
- investigation of variability in printing-imaging systems.

Notations We use the following notations: $t \in \{0,1\}^{m \times m}$ denotes an original digital template; $x \in [0,1]^{m \times m}$ corresponds to an original printed code, while $\hat{x} \in [0,1]^{m \times m}$ is used to denote a printed fake code; $y \in [0,1]^{m \times m}$ stands for a probe that might be either original or fake. We use $E_{p(x)}[\cdot]$ to denote mathematical expectation with respect to a distribution $p(x)$, $D_{KL}(\cdot\|\cdot)$ denotes Kullback-Leibler divergence and $I(\cdot;\cdot)$ stands for mutual information. We assume that a pair of digital template and CDPy are distributed as $(y,t) \sim p_{y,t}(y,t)$.

II. Proposed Turbo digital twin system

The proposed digital twin system is based on an auto-encoder structure and is represented by general stochastic encoder $q_{\theta_t}(t|y)$ and decoder $p_{\theta_y}(y|t)$ that are deep networks parametrized by the parameters θ_t and θ_y, respectively. The block diagram of Turbo digital twin system is shown in Fig. 1.

According to the proposed framework, given a pair of observable vectors $(y,t) \sim p_{y,t}(y,t)$, where t is a digital template and y is a printed code, i.e. either original or fake, the system maximizes the mutual information between y and t for both encoder and decoder in direct and reverse paths. We are considering a variational approximation for the direct path of the proposed system based on the maximization of two bounds on mutual information for the latent space and the reconstruction space:

$$I^{\theta_t}_{\theta_t}(Y;T) = E_{p_{y,t}(y,t)}\left[\log \frac{q_{\theta_t}(t|y)}{p_{\theta_t}(t)} \frac{q_{\theta_t}(t)}{q_{\theta_t}(t)}\right] \geq \underbrace{E_{p_y(y)} E_{q_{\theta_t}(t|y)}[\log q_{\theta_t}(t|y)]}_{L_{\theta_t}(t;\theta_t)} - \underbrace{D_{KL}(p_{\theta_t}(t)\|q_{\theta_t}(t))}_{D_{t\theta_t}(\theta_t)}; \quad (1)$$

$$I^{\theta_y}_{\theta_t,\theta_y}(T;Y) = E_{p_{y,t}(y,t)}\left[\log \frac{p_{\theta_y}(y|t)}{p_{\theta_y}(y)} \frac{\hat{p}_{\theta_y}(y)}{\hat{p}_{\theta_y}(y)}\right] \geq \underbrace{E_{p_y(y)} E_{q_{\theta_t}(t|y)} \log p_{\theta_y}(y|t)}_{L_{\theta_y}(y;\hat{y})} - \underbrace{D_{KL}\left(p_{\theta_y}(y)\|\hat{p}_{\theta_y}(y)\right)}_{D_{y\hat{y}}(\hat{y})}. \quad (2)$$

Thus, the network is trained in such a way to maximise a weighted sum of (1) and (2) in order to find the best parameters θ_t and θ_y of the encoder and the decoder, respectively. This is achieved in the direct path by minimising the L^{Direct} loss, representing the left network shown in Fig. 1:

$$L^{\text{Direct}}(\theta_t,\theta_y) = L_{\theta_t}(t;\theta_t) + D_{t\theta_t}(\theta_t) + \alpha L_{\theta_y}(y;\hat{y}) + \alpha D_{y\hat{y}}(\hat{y}); \quad (3)$$

where α is a parameter controlling the trade-off between the terms (1) and (2).

The variational approximation for the reverse path is:

$$I^{\theta_y}_{\theta_y}(T;Y) \geq \underbrace{E_{p_t(t)} E_{p_{\theta_y}(y|t)} \log p_{\theta_y}(y|t)}_{L_{\theta_y}(y;\theta_y)} - \underbrace{D_{KL}\left(p_{\theta_y}(y)\|\hat{p}_{\theta_y}(y)\right)}_{D_{y\hat{y}}(\hat{y})}; \quad (4)$$

$$\mathcal{I}^{\mathbf{t}}_{t;\ y}(Y;T) \geq \underbrace{\mathsf{E}_{p_t(\mathbf{t})}\mathsf{E}_{p\ y}(\mathbf{y}|\mathbf{t})[\log q\ _t(\mathbf{t}|\mathbf{y})]}_{L_{\mathbf{t}}(\mathbf{t},\hat{\mathbf{t}})} - \underbrace{D_{\mathsf{KL}}\left(p_t(\mathbf{t})\|q\ _t(\mathbf{t})\right)}_{D_{t\hat{t}}(\hat{\mathbf{t}})}: \tag{5}$$

The reverse path loss $\overline{\mathcal{L}}^{\text{Reverse}}$, weighted by , is represented by the right network shown in Fig. 1:

$$\overline{\mathcal{L}}^{\text{Reverse}}(\ _t;\ _y) = \mathcal{L}_{\mathbf{y}}(\mathbf{y};\hat{\mathbf{y}}) + \mathcal{D}_{\mathbf{y}\hat{\mathbf{y}}}(\hat{\mathbf{y}}) + \ \mathcal{L}_{\hat{\mathbf{t}}}(\mathbf{t};\hat{\mathbf{t}}) + \ \mathcal{D}_{\mathbf{t}\hat{\mathbf{t}}}(\hat{\mathbf{t}}): \tag{6}$$

III. Architectural details

The *Turbo* system is flexible and allows different configurations. It can be used for paired data when all losses are preserved and we possess pairs of digital template $\mathbf{t}$ and CDP $\mathbf{y}$. In contrast, if such pairs are not available at the training that corresponds to the unpaired setup, the terms $\mathcal{L}_{\mathbf{t}}(\mathbf{t};\mathbf{t})$ and $\mathcal{L}_{\mathbf{y}}(\mathbf{y};\hat{\mathbf{y}})$ disappear and one gets a *Turbo unpaired* setup.

In addition, many existing models can be expressed as part of the *Turbo* framework. For example, the CycleGAN [2] model can be obtained by removing the discriminators on reconstruction $\mathcal{D}_{\mathbf{t}\hat{\mathbf{t}}}(\hat{\mathbf{t}})$ and $\mathcal{D}_{\mathbf{y}\hat{\mathbf{y}}}(\hat{\mathbf{y}})$ from *Turbo unpaired*. The pix2pix model [11] is also a part of the complete *Turbo* framework with the removed cycle losses while keeping $\mathcal{L}_{\mathbf{t}}(\mathbf{t};\mathbf{t});\mathcal{D}_{\mathbf{t}\mathbf{t}}(\mathbf{t})$ or $\mathcal{L}_{\mathbf{y}}(\mathbf{y};\hat{\mathbf{y}});\mathcal{D}_{\mathbf{y}\hat{\mathbf{y}}}(\hat{\mathbf{y}})$ depending on the direction of training. The adversarial autoencoder (AAE) [1] corresponds to the direct path with the adversarial and reconstruction losses. The CUT [12] and ALAE [3] models can also be expressed through the Turbo framework.

A. Structure of encoder and decoder

The proposed approach does not impose any restrictions on the encoder and decoder architecture, which allows a wide variety of options. In our work, we have considered several most widely used architectures for the encoders and decoders, namely:

- CNN-RESNET-CNN adapted from CycleGAN [2] and StarGAN [13] models, consisting of two convolutional layers for downsampling, nine residual blocks [14], and two transposed convolutional layers for upsampling.
- UNET [15] with skip-connections layers.

In both cases, instance normalization [16] was used to stabilize training together with Adam optimizer [17].

B. Adversarial loss and structure of discriminator

Selection of the adversarial loss, which implements $D_{\mathrm{KL}}(:||:)$ terms, for the considered models could be crucial for the success of the training [18]. In our work, we examine three of the most popular losses: LSGAN [19], HINGE [20] and WGAN [21] with gradient penalty [22].

We started with the standard PatchGAN [11] discriminator. However, we quickly discovered that in combination with a WGAN-GP loss, the results were extremely bad. We believe this follows from the fact that PatchGAN generates overlapping patches, which interfere when calculating the earth's moving distance. Therefore, we added another discriminator "ImageGAN", based on residual networks [14], for the comparison, which takes the whole picture as the input and produces a single scalar output.

IV. Training details

We used PyTorch for all experiments. One training cycle per model varies from one to four days using four RTX 2080 Ti or a single A100 80 GB card depending on the configuration.

A. Dataset

For the empirical evaluation of the proposed *Turbo* framework, we use the Indigo 1x1 base dataset [23] that consists of CDP with 1×1 pixel symbol size. This dataset contains 720 samples that we divide at 80% and 20% for the training and test sets, respectively. For the sake of experimental purity, the same non-intersecting split is used in all trials. To speed up the study, each original image of size 684×684 pixels is divided into four non-overlapping crops of size 256×256 each. Due to the paper length limit, all of the results below are obtained for the HP Indigo 7600 printer (HPI 76), but we do not observe significant differences when codes printed on another printer are used as input.

B. Setups under consideration

To our best knowledge, all previous works in CDP field use only paired data for the estimation. However, we believe that this condition might not always hold. One of the key novelties of our work is that we consider the case where an attacker has an unpaired dataset, where there is no exact match between the digital template and the respective printed code, and all data are represented as an unordered set.

However, the flexibility of the *Turbo* framework allows the use of paired losses $\mathcal{L}_{\mathbf{t}}(\mathbf{t};\mathbf{t})$ and $\mathcal{L}_{\mathbf{y}}(\mathbf{y};\hat{\mathbf{y}})$ if paired data is available. It is also possible to train only one path estimation for example from the template to the printed code or vice versa.

C. Stability of training

Adversarial training with discriminators is known to be quite unstable due to the mode collapse and vanishing gradients. Therefore, the following refinements were investigated to improve the quality of results:

- Balancing discriminator and generator iterations via the number of discriminator iterations per generator iteration n_D [22].
- However, selecting the appropriate number of iterations is not an obvious task. Therefore, in the case of constraints on the possible values of loss function, i.e., in the case of LSGAN — values are non-negative, instead of one parameter, a principled approach is preferable, where the discriminator is updated if its loss is greater than $D_{\text{threshold}}$ (discriminator poorly separates the generated samples) or the generator's loss is less than $G_{\text{threshold}}$ (the generated samples easily fools the discriminator) [24].

Updating the discriminator using the history of generated images, rather than only those generated at the last iteration [25].

Flipping labels from time to time when training the discriminator with probability p_{flip} [24] and adding some artificial noise to the discriminator's inputs [26] with probability p_{noise} and weight w_{noise}. We have experimented with ways of combining these two heuristics and noticed that together they give better results compared to using only one or none of them.

V. Computer simulation

The reported results are obtained without any post-processing and represent a direct output of deep networks. Additional post-processing might increase the accuracy of digital template estimation and generation. However, to preserve the scalability to any artwork and fair comparison, we report all results without any refinements. The UNET paired model from [23] is used as a baseline.

A. Metrics

The following metrics were used to evaluate the quality of the predictions:

Hamming distance $d_H(\mathsf{t}; \mathsf{binary}(\tilde{\mathsf{t}}))$, where $\mathsf{binary}(\cdot)$ is a binarization function.

Mean square error (MSE) distance $d_2(\mathsf{y}; \tilde{\mathsf{y}})$.

Structural similarity index (SSIM) $d_{SSIM}(\mathsf{y}; \tilde{\mathsf{y}})$ introduced in [27] to address an issue that the mean squared error is not highly indicative of perceived similarity of images.

Fréchet Inception Distance (FID): $\mathrm{FID}_{\mathrm{t} \to \tilde{\mathrm{y}}}$ and $\mathrm{FID}_{\mathrm{y} \to \tilde{\mathrm{t}}}$ proposed in [28]. Instead of a simple pixel-by-pixel comparison of images, FID estimates the mean and standard deviation of one of the deep layers in the pretrained convolutional neural network. We suppose that the usage of deep network statistics can be helpful not only as a measure of human perception of image similarity but also to assess the difficulty of distinguishing the generated images from the real ones since the network activations are similar at a metric close to zero.

B. Evaluation

First of all, we investigated the impact of the encoder-decoder architecture. The obtained results are shown in Table I. In all scenarios, the configuration with CNN-RESNET-CNN performs better than with UNET. However, in the case of paired data, the difference is less significant. The *Turbo* paired also outperforms CycleGAN with respect to most metrics and is also less sensitive to the choice of architecture.

Table II illustrates the impact of adversarial loss and discriminator type depending on the chosen *Turbo* configuration. It should be noted that the configuration with WGAN-GP [22] does not converge when used together with PatchGAN, but shows one of the best results with ImageGAN.

The best results among all investigated configurations are summarized in Table III. It is obvious that the models without pairwise information perform worse. The *Turbo* configurations outperform also contrastive system based on the CUT model, and the *Turbo paired* outperforms the baseline in almost all metrics.

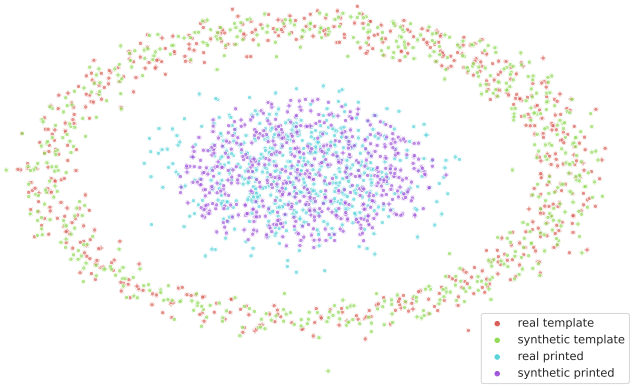

Fig. 2. Umap [29] visualisation for artificially generated templates and printed jointly with digital templates and physically printed codes.

C. Visualization

To further assess the quality of the proposed models, an UMAP [29] visualisation was performed. The projection of artificially generated templates and printed codes from the test part jointly with corresponding original digital templates and physically printed codes is shown in Fig. 2. As expected, matching samples are close to each other, and there are different clusters for printed and template codes.

D. Visualization of synthetic samples

To illustrate the quality of the synthetic samples produced by various systems studied in this paper, we pick up a random sample and show both synthetic digital templates estimated from physical CDP and vice versa in Table IV. Models that use paired examples show better generation performance, but models trained entirely in unpaired mode also perform decently. Visually, the synthetic samples look almost indistinguishable from their real counterparts.

VI. Conclusions

In this paper, we present the *Turbo digital twin* framework for the simulation of the physical printing-imaging channel. We believe that such a differential model allows to consider the adversarial fakes for the physical world and also opens new perspectives for the optimization of authentication systems.

For future work, we will consider the usage of the generated examples to build a classifier based on the augmented synthetic samples of both original CDP and fakes. Additionally, issues of stochasticity and usage in hybrid settings, where only part of the data is paired, remain open for future research.

References

[1] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

TABLE I
TURBO PERFORMANCE WITH REGARD TO THE ENCODER-DECODER BACKBONE

Model	backbone	$\text{FID}_{y \to \tilde{t}}$	Hamming distance	$\text{FID}_{t \to \tilde{y}}$	MSE	SSIM
CycleGAN [2]	UNET	9.599	0.195	12.399	0.064	0.675
	CNN-RESNET-CNN	3.865	0.155	**4.451**	0.049	0.732
Turbo paired	UNET	4.272	0.107	8.942	0.043	0.765
	CNN-RESNET-CNN	**3.164**	**0.086**	6.605	**0.040**	**0.779**

TABLE II
TURBO PERFORMANCE WITH REGARD TO THE GAN LOSS AND DISCRIMINATOR TYPE

Model	GAN Loss	Discriminator type	$\text{heuristics}_{\text{IV-C}}$	$\text{FID}_{y \to \tilde{t}}$	Hamming distance	$\text{FID}_{t \to \tilde{y}}$	MSE	SSIM
CycleGAN [2]	LSGAN [19]	PatchGAN	✗	66.968	0.214	14.611	0.061	0.689
			✓	46.215	0.205	9.828	0.064	0.683
		ImageGAN	✗	48.801	0.198	12.481	0.065	0.668
			✓	65.458	0.203	5.713	0.062	0.694
	HINGE [20]	PatchGAN	✗	4.074	0.194	**4.451**	0.065	0.673
			✓	4.741	0.184	9.340	0.062	0.687
		ImageGAN	✗	124.996	0.243	20.254	0.071	0.650
			✓	28.206	0.192	5.978	0.066	0.674
	WGAN-GP [22]	PatchGAN	✗	236.835	0.225	68.579	0.079	0.628
			✓	248.583	0.231	80.362	0.078	0.628
		ImageGAN	✗	3.865	0.155	8.398	0.057	0.714
			✓	4.130	0.163	17.415	0.049	0.732
Turbo unpaired	LSGAN [19]	PatchGAN	✗	33.303	0.201	12.300	0.063	0.674
			✓	51.815	0.208	14.399	0.062	0.681
		ImageGAN	✗	136.316	0.211	20.880	0.070	0.667
			✓	26.151	0.195	12.048	0.065	0.679
	HINGE [20]	PatchGAN	✗	4.479	0.195	12.651	0.063	0.674
			✓	3.571	0.186	16.121	0.064	0.678
		ImageGAN	✗	322.374	0.432	54.078	0.189	0.300
			✓	25.717	0.177	16.836	0.059	0.683
	WGAN-GP [22]	ImageGAN	✗	3.601	0.155	5.912	0.065	0.678
			✓	4.333	0.167	15.031	0.064	0.685
Turbo paired	WGAN-GP [22]	ImageGAN	✗	**3.164**	**0.086**	6.605	0.043	0.772
			✓	4.312	0.092	9.712	**0.040**	**0.779**

TABLE III
PERFORMANCE WITH REGARD TO THE MODEL

Model	$\text{FID}_{y \to \tilde{t}}$	Hamming distance	$\text{FID}_{t \to \tilde{y}}$	MSE	SSIM
CUT [12]	3.8644	0.1990	5.2941	0.0610	0.6979
CycleGAN [2]	3.8653	0.1549	**4.4507**	0.0490	0.7315
Turbo unpaired	3.5713	0.1550	5.9117	0.0589	0.6849
Turbo paired	**3.1640**	**0.0855**	6.6049	0.0400	**0.7787**
UNET paired [23]	6.2113	0.1002	28.1099	**0.0363**	0.7775