Parallel Learning: Overview and Perspective for Computational Learning Across Syn2Real and Sim2Real

Qinghai Miao, *Senior Member, IEEE*, Yisheng Lv, *Senior Member, IEEE*, Min Huang, *Member, IEEE*, Xiao Wang, *Member, IEEE*, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*—The virtual-to-real paradigm, i.e., training models on virtual data and then applying them to solve real-world problems, has attracted more and more attention from various domains by successfully alleviating the data shortage problem in machine learning. To summarize the advances in recent years, this survey comprehensively reviews the literature, from the viewport of parallel intelligence. First, an extended parallel learning framework is proposed to cover main domains including computer vision, natural language processing, robotics, and autonomous driving. Second, a multi-dimensional taxonomy is designed to organize the literature in a hierarchical structure. Third, the related virtual-to-real works are analyzed and compared according to the three principles of parallel learning known as description, prediction, and prescription, which cover the methods for constructing virtual worlds, generating labeled data, domain transferring, model training and testing, as well as optimizing the strategies to guide the task-oriented data generator for better learning performance. Key issues remained in virtual-to-real are discussed. Furthermore, the future research directions from the viewpoint of parallel learning are suggested.**

***Index Terms*—Machine learning, parallel learning, parallel systems, sim-to-real, syn-to-real, virtual-to-real.**

I. Introduction

BIG data, algorithms, and computing capability, are well known as three driving forces that push deep learning to the current prosperity. Algorithms, as well as a variety of neural models, have kept refreshing the state-of-art (SOTA) during the last decade, boosted by powerful computing hardware. ResNet (residual neural network), YOLO (you only look once), Mask R-CNN (region convolutional neural network), transformer, and GAN (generative adversarial network) are such outstanding models/methods.

Besides many achievements from both academia and industry, deep learning is obstructed in some areas because of data scarcity. Factors that affect the data for deep learning mainly come from three aspects. First, for specific tasks like visual perception in autonomous driving, data imbalance is a typical problem. While most of cases can be collected through daily driving across the city, corner cases including vital accidents occur at extremely low frequency, which is known as long tail distribution. In such a situation, data acquisition methods cannot make sure the training data appropriately represent the data distribution supposed to be covered by deep learning models. Second, data labeling for supervised methods is labor-intensive and time-consuming, and the quality of labeled data is not always guaranteed due to human mistakes. Third, data privacy is an essential factor that cannot be neglected. For example, biometric data that are related to facial identification and health examinations are not publicly available for deep learning in order to protect them from malicious use.

To address these data problems, people have proposed methods either augmenting existing data or transferring models trained on plenty of data to domains with few data available. Particularly, virtual data generated by synthetization or simulation have attracted increasing attention in recent years. The main feature is using image processing tools, computer graphics tools, and simulation toolkit to automatically generate labeled data for deep models to get better performance and generalization in real applications. Taking visual tasks as an example, works on sub-tasks like classification, recognition, detection, segmentation, estimation, reported improvements on performance with the help of virtual data. People from domains like computer vision, autonomous driving, robotics control, as well as natural language processing have taken advantage of virtual data to address data scarcity problems in their applications. These methods are referred as the virtual-to-real paradigm, or V2R in short.

During the last five years, numerous methods have been proposed to generate virtual data for machine learning. As a

Manuscript received October 11, 2022; revised November 23, 2022; accepted December 20, 2022. This work was partially supported by the National Key Research and Development Program of China (2020YFB 2104001), the National Natural Science Foundation of China (62271485, 61903363, U1811463), and Open Project of the State Key Laboratory for Management and Control of Complex Systems (20220117). Recommended by Associate Editor Choon Ki Ahn. *(Corresponding author: Yisheng Lv.)*

Citation: Q. H. Miao, Y. S. Lv, M. Huang, X. Wang, and F.-Y. Wang, "Parallel learning: Overview and perspective for computational learning across Syn2Real and Sim2Real," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 3, pp. 603–631, Mar. 2023.

Q. H. Miao, Y. S. Lv, and M. Huang are with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: miaoqh@ucas.ac.cn; yisheng.lv@ia.ac.cn; huangm@ucas.ac.cn).

X. Wang is with the School of Artificial Intelligence, Anhui University, Hefei 266114, China, and also with Qingdao Academy of Intelligent Industries, Qingdao 230031, China (e-mail: xiao.wang@ahu.edu.cn).

Y. S. Lv and F.-Y. Wang are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yisheng.lv@ia.ac.cn; feiyue.wang@ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2023.123375

fast-developing topic, reviews on recent progress are essential for further development. There have been several related surveys. Nikolenko [1] conducted a review on synthetic data for deep learning, covering data synthetization within and outside computer vision tasks, synthetic-to-real domain adaptation problems and privacy-related applications. Shorten and Khoshgoftaar [2] and Tsirikoglou *et al.* [3] emphasized on image augmentation and image synthesis respectively. Zhao *et al.* [4] and Muratore *et al.* [5] focused on domain randomization in sim-to-real dedicated to robotic control. Although these surveys reviewed the virtual-to-real methods from different aspects, it is still necessary to review new advancements from a comprehensive viewport. First, most the existing surveys mainly focused on specific sub-domain, like computer vision and robotics control, etc. Lacking of high level virtual-to-real theory makes it difficult for readers to compare methods and borrow better ideas from different domains. It is highly necessary to summarize the common issues and techniques of virtual-to-real methods across sub-domains in machine learning. Second, the existing taxonomy is too simple to help readers from different domains to find the most related works. A comprehensive taxonomy with hierarchical dimensions is more appreciated. In addition, since these surveys published, much of the work has made incremental advances to state-of-the-art, so establishing a baseline for future work remains important.

To overcome these constraints, in this paper, we provide a survey on virtual-to-real methods at the level of machine learning, under the framework of parallel intelligence [6], [7]. We try our best to cover new advancements in recent years, in multiple machine learning sub-domains including computer vision (CV), natural language processing (NLP), robotics, and autonomous driving (AD), as shown in Fig. 1. We propose a multi-dimensional taxonomy according to the three key components of parallel learning, i.e., description, prediction, and prescription.

This paper is organized as follows. The related background is introduced in Section II. A new taxonomy with multiple dimensions is presented in Section III. Methodologies are reviewed in Section IV succeeded by discussion in Section V. The paper is summarized with the remaining problems and future directions in Section VI.

This survey provides the following contributions:

1) An extended Parallel Learning framework covering main machine learning tasks including computer vision, natural language processing, robotics and autonomous driving.

2) A systematical survey of the existing methods via virtual-to-real paradigm from the viewpoints of parallel learning.

3) A multi-dimensional and multi-level taxonomy of virtual-to-real methods.

4) A discussion about the current situation, and the main challenges and opportunities for future work.

II. Background

In this section, we give a background of virtual-to-real in machine learning. We start with parallel intelligence [6], which is a theory on interactions between the virtual and real worlds to form a closed-loop artificial intelligence (AI) system. We then introduce related techniques for virtual data generation, including domain augmentation, domain randomization, and domain adaptation. We also have a look at the challenges in syn-to-real or sim-to-real applications.

A. Parallel Intelligence

Though the virtual data used in machine learning can be traced back to a very early age, it was regarded as a trick to alleviate data shortage problems. There was not a theory to guide the use of real and virtual data until 2004 when Fei-Yue Wang proposed the theory of parallel intelligence (PI), with three principles known as artificial systems, computing experiments and parallel execution (ACP) [6]−[10]. The parallel intelligence theory has been successfully applied to various domains like [11]−[14], etc. Further in 2017, Li *et al.* [15] proposed the parallel learning (PL) framework that searches for optimized policy with virtual-real interactions. The PL framework comprises three components known as description, prediction, and prescription, as shown in Fig. 2.

Given a machine learning task in real world, the description in PL stands for the process to construct corresponding artificial system from which virtual data are derived with labels. A typical way is taking off-the-shell computer graphics tools to reconstruct three-dimensional virtual scenes, aiming to generate virtual data that approaches the real data distribution. In the case of reinforcement learning (RL), system identification is required to reconstruct the real world with high fidelity not only on appearance but also on dynamic parameters.

The prediction in PL is to train models with the generated virtual data aiming for better generalization, or train intelligent agents within the virtual environment aiming for better policy. Succeeding tests on the real data or in the real world are fulfilled to evaluate the learning performance. The training and testing form the pipeline of computational experiments.

Due to the domain gap between the virtual and real world, multiple trials are required to repeat the phases of description and prediction, which results in computational overhead. To deal with this issue, the prescription in PL is essential for better reconstruction of the virtual environment, by feeding back the errors from Prediction to the description. This closed-loop connecting both the real and virtual system iteratively improves the efficiency of the learning process.

B. Digital Twin

The concept of the digital twin was introduced by Grieves [16] around 2003 for product lifecycle management, while the term digital twin was coined by National Aeronautics and Space Administration (NASA), USA in 2010 [17] and has since become a hot topic in recent years, driven by both industry and academia. The principle of the digital twin is to create a digital model (usually three-dimensional) of a physical entity (usually a product) and to evaluate, optimize and manage the physical entity through the interaction between reality and reality. Although parallel intelligence and the digital twin share some common features such as virtual-real interaction, they differ in several ways. Influenced by cybernetics, parallel intelligence covers a broader field of study, including not

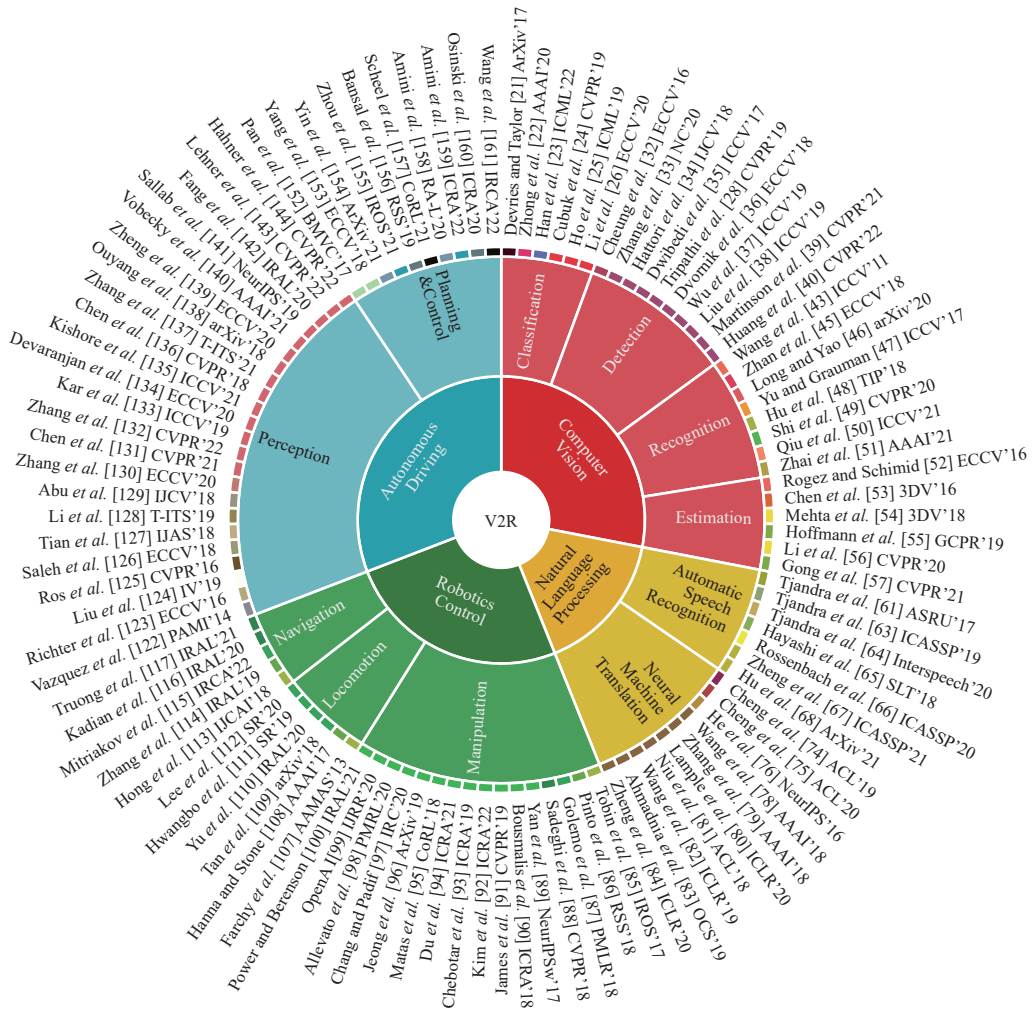


Fig. 1. Overall picture of the survey.

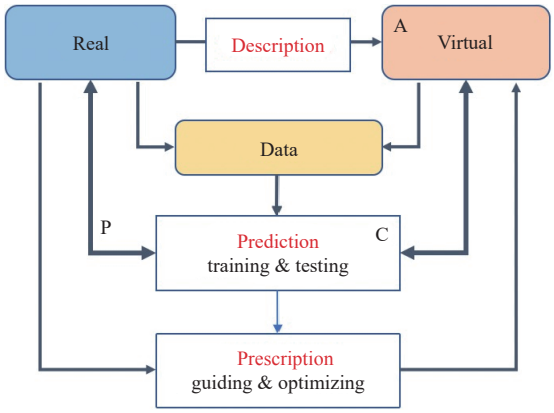


Fig. 2. A brief illustration of parallel learning under the ACP framework.

only physical products but also social systems. Inspired by the Merton's theory, parallel intelligence places more emphasis on the three principles of description, prediction, and especially prescription. For the virtual-to-reality paradigm discussed in this paper, AI applications using digital twins, e.g., [18] and [19], focus on the construction of accurate 3-dimensional digital models through evaluation, prediction, and simulation to improve physical entities. Thus, the principles of AI applications through digital twins have been incorporated into an extended parallel intelligence framework. The related literature can be classified into the taxonomy introduced in this paper.

C. Virtual-to-Real in Machine Learning

From Deep Blue to AlphaGo, data play an essential role in the breakthrough of artificial intelligence. The AlphaGo paradigm indicates that, we can find a proper way to generate data if the real labeled data are insufficient. Games like Chess and Go are ideal cases where the rules can be used to run rollouts for the purpose of data collecting. In practical applications of machine learning, there are two types of methods working similarly to AlphaGo paradigm, i.e., syn-to-real and sim-to-real, or virtual-to-real as the unified name.

1) Syn-to-Real: Data synthesis is the most popular method to generate virtual data, especially images for deep visual tasks. There is a variety of ways to generate synthetic images. According to sources from which synthetic images are derived, we can divide synthetic methods into two types: 3D−2D and 2D−2D. The 3D−2D is firstly constructing virtual 3D scenes with objects and then rendering images with configurations of camera pose, lighting, and post processing. Both foreground and background of generated images are

fully configurable. Unlike 3D−2D, the 2D−2D method tries to insert new objects into background images, followed by an additional fusing process to reduce artefacts. Some popular methods include data augmentation (DA) that modifies the initial image by translating/rotating the objects, image composition that inserts new objects, and image translation that converts the images to target styles.

2) Sim-to-Real: Intelligent agents usually learn to navigate or manipulate in a given environment. As running experiments in the real world is both dangerous and expensive, people turn to take advantage of simulations. Intelligent agents interact with the virtual environment reconstructed in a simulator, supported by including numerical simulation, rendering simulation, and mechanical simulation. The goal is transferring the policy learned in virtual environment to real world with satisfied performance. Domain randomization (DR) is the most popular method in Sim2Real application, including visual domain randomization and dynamic randomization.

D. Challenges

1) Challenges of Description (Data Generation)

a) Virtual world reconstruction: The first step for a virtual-to-real application is to build a system to generate virtual data. Building such a system, whether virtual 3D scenes or directly composing on 2D images, is nontrivial and sometimes requires intensive human interventions.

b) Domain gap: It is well known that two datasets sampled under different conditions have domain gaps in distribution. A learning model trained and tested well on one dataset usually cannot generalize well on the other one. Such a domain gap is also inevitable between virtual and real datasets.

2) Challenges of Prediction

To verify the effeteness of virtual data, multiple rounds of training and testing are required to get satisfying performance. However, training deep learning models on big data is time consuming. A large number of generated virtual data make the situation even worse.

3) Challenges of Prescription

A large number of virtual data can be generated at a low cost by image augmentation or domain randomization. However, experiments have indicated that randomly generated virtual data may be harmful to the learning process. Methods to ensure data quality are essential in virtual-to-real applications. To this end, there is much work to be done, albeit some recent works have realized this and tried to improve the quality of virtual data.

III. Taxonomy of Virtual-to-Real Parallel Learning Methods

To give readers a comprehensive overview of the virtual-to-real methodology, we present a hierarchical taxonomy from the viewpoint of parallel intelligence.

A. Classification by Application Domains and Tasks

Researchers on machine learning may come from different domains with different academic backgrounds. It is better to classify the literature according to research domains at the top level. This paper covers four popular domains known as computer vision (CV), natural language processing (NLP), robotics control and autonomous driving (AD), as illustrated in Fig. 1.

The first category is computer vision (CV), which achieved great improvements during the last decade. We further classify the related works according to visual tasks including classification, detection, recognition, and estimation.

Natural language processing (NLP) has become a hot domain with the introduction of transformer in recent years. Sequential data like text can be regarded as another type of perception of real world, as essential as images. The training of NLP models also requires a huge number of labeled data, where the virtual-to-real paradigm can take its role.

Robotics is a comprehensive domain that heavily depends on simulation environment due to the high cost of real robotic experiments. The main challenge is finding optimized policy in a simulated environment and then transferring it to the real world.

Autonomous driving (AD) is another domain covering a large number of virtual-to-real applications. Though sharing some common perception and control tasks in CV and robotics, AD has more challenges including multiple sensors fusion, planning under complex environments, and high safety requirements.

B. Classification According to Parallel Learning

At the second level, we analyze each specific virtual-to-real case according to the three aspects of parallel learning, i.e., description, prediction, and prescription.

1) Description: For the description, we aim to summarize the methods to construct a virtual system from which we can generate virtual data with labels. We design an array whose elements represent the main features of a virtual-to-real task including dimensional mapping type, software to generate virtual data or environment, name of real dataset, name and size of generated virtual dataset, domain adaptation methods, etc.

a) Real-virtual mapping: We regard building a virtual system as a mapping from the real world to virtual world, as shown in Fig. 3. Both real data and generated virtual data may have a shape of one (1D), two (2D), three (3D) or four (4D) dimensions. Specifically, text and audio are 1D data with sequential representations, image is in the popular 2D shape, video and point cloud are typical 3D data, while simulations in 3D space are 4D data.

A dimensional mapping from real to virtual reflects how we build a virtual system. For example, image augmentation, image composition and style transfer are all 2D−2D mappings that take real images as input and give modified images as output. The 2D−2D mapping usually edits the foreground while keeping the background of the input image unchanged so that the domain gap is small.

3D−2D mapping is another popular style. To this end, a virtual 3D scene is firstly constructed as a source with an off-the-shell software then rendered 2D images as output. This method does not depend on real images and can generate a set of images by projecting from a camera with various configurations.

3D−3D mapping is usually seen in domain of autonomous

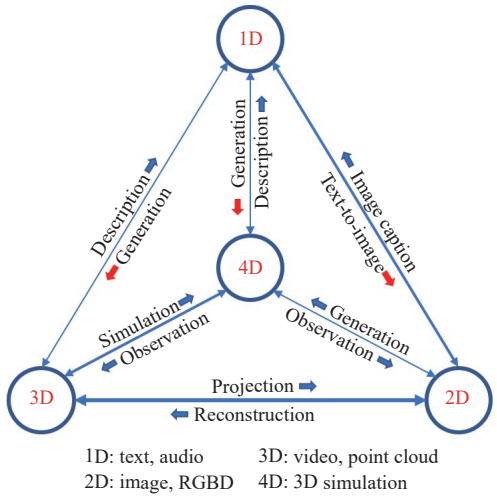


Fig. 3. Dimensional data type mapping of virtual data generation.

driving where laser radar (LiDAR) acts as an essential sensor. Virtual 3D cloud points can be generated from 3D virtual scenes.

3D−4D mapping means conducting dynamic simulations in a 3D virtual environment where an agent, e.g., a robot or an autonomous vehicle, can learn policies.

We refer to sequential data, including text and audio, as 1D data. Though in the minority, mappings from 1D source to high dimensional target are emerging. Examples of 1D−1D mapping include augmenting text in NLP, or synthetizing sound in speech recognition. 1D−2D or 1D−3D mapping, which means generating images or 3D scenes from a sentence, has become a new frontier in MultiModal machine learning, like DALL-E [20] and its succeed versions.

b) Domain adaption methods: Domain adaption to minimize the gap between the virtual and real data distributions is essential to the success of virtual-to-real transfer. There are various domain adaptation methods, including data augmentation, realistic rendering, style translation, image composition, domain randomization, dynamic randomization, feature alignment, pretraining and finetuning, etc.

2) Prediction: For prediction, we first introduce the type of learning models for the specific task. virtual-to-real applications cover most machine learning models, e.g., Faster R-CNN for object detection tasks, mask R-CNN for segmentation tasks, or quantum CNN (QCNN) of reinforcement learning for robots. We also select representative evaluation results to show their performance improvements after training with virtual data.

Training-testing strategy: To utilize virtual data, there are different choices according to the availability of real data. First, we can train a model purely on virtual dataset and apply it to real world in zero-shot manner, in case of no real data available. Second, we can mix real and virtual data by a certain ratio to train the models and then test on real data. Third, we can train a model purely on virtual data then finetune it on a small part of real data, or optimize a policy with a few real rollouts.

3) Prescription: For prescription, we emphasize on the formats to improve the quality and efficiency of virtual data generation. Typical formats include open-loop, knowledge-guided, adversarial, and feed-back.

a) Open-loop: The open-loop means a stepwise pipeline of training and testing, a common procedure for most of machine learning applications. For virtual-to-real, this means training on virtual data, usually synthetic images, then testing on real data. Or, training an agent, e.g., a robot or an autonomous vehicle, in simulation and then applying it to a real environment. There is no feedback from testing to training step.

b) Guided: There is no guarantee that the randomly generated virtual data help in boosting learning performance. Cues (knowledge) drawn from additional experiments or human heuristic experiences can be used as guidance to improve the data quality with higher efficiency.

c) Feed-back: While the open-loop and guided formats have been approved to be effective in some works, they are limited by laborious human interventions that also inevitably introduce bias in virtual data distribution. Feed-back, on the contrary, is regarded a better choice that combines virtual data generator, domain adaptor and the learning model into one closed loop wiping out human interventions. One typical strategy is to feed back the test errors of learning model on validation dataset to the data generator as guiding signals. Two forms of the closed loops, for different learning tasks, are shown in Fig. 4.

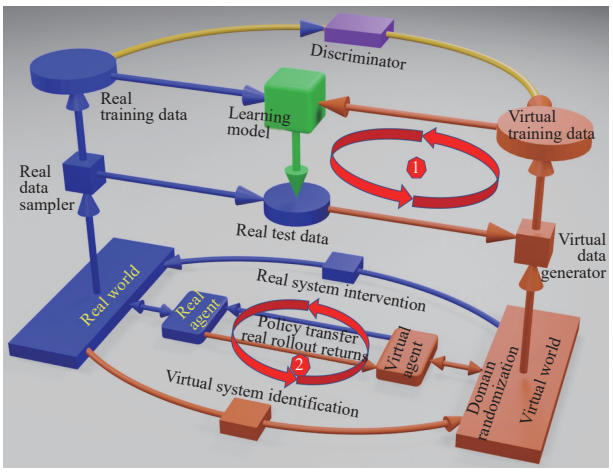


Fig. 4. Extended parallel learning framework. The cycles 1 and 2 indicate prescriptive learning process.

d) Adversarial: Another commonly used strategy is to organize the data generator and learning model into an adversarial framework by introducing one or multiple discriminators, as shown in upper side of Fig. 4. This method takes advantages of GANs and works well when a part of real data are available.

IV. Virtual-to-Real Learning Methods Overview

In this section, we give an overview of virtual-to-real methods according to the taxonomy described in Section III. For each domain of computer vision, natural language processing, robotics and autonomous driving, typical solutions are presented with their features and highlights in tables, where readers can quickly get the differences and similarities of these

TABLE I
COMPARISON OF SELECTED WORKS ON VIRTUAL-TO-REAL IN CLASSIFICATION

Method	Description		Prediction				Prescription	Highlights
	Real data	DA	Model	T-time (h)	Error (%)	Improve (%)	Loop	
Devries *et al*. [21] ArXiv'17	CIFAR-10 CIFAR-100	Random Cutout	WRN-28-10	– –	5.54 23.94	1.43 2.12	Open	Fixed-size zeromask to random location
Zhong *et al.* [22] AAAI'20	CIFAR-10 CIFAR-100	Random Erasing	WRN-28-10	– –	3.08 17.73	0.72 0.76	Open	Randomly erase rectangle region with random values
Han *et al.* [23] ICML'22	CIFAR-10 CIFAR-100	Cut aug Concat	WRN-28-10	– –	2.6 17.17	0.7 –	Open	Half cut image augment each concatenate two
Cubuk *et al.* [24] CVPR'19	CIFAR-10 CIFAR-100	RL search	WRN-28-10	5000 –	2.68 17.09	0.6 1.5	Feedback accuracy	Search for best policy in DA space inspired by NAS
Ho *et al.* [25] ICML'19	CIFAR-10 CIFAR-100	PBA search	WRN-28-10	5 –	2.58 16.73	0.1 0.32	Feedback accuracy	DA schedule defining best policy for each epoch
Li *et al.* [26] ECCV'20	CIFAR-10 CIFAR-100	Differentiable search	WRN-28-10	0.1 –	2.7 17.5	−0.1 −0.4	feedback accuracy	Differentiable DA gradient-based optimization

methods. Please note that the items in the tables are partially selected from the original papers. For complete information, please refer to the original papers for more details. For a better understanding, some figures are taken from the literature and grouped for comparison.

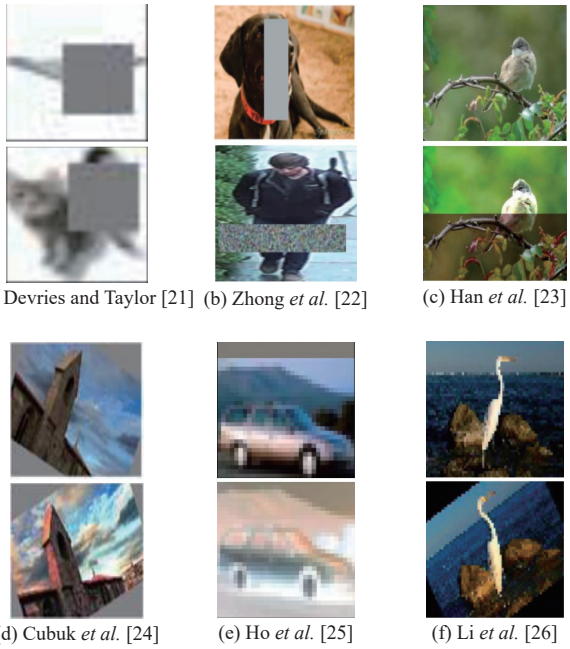
A. Virtual-to-Real in Computer Vision

Though computer vision is the most prosperous AI domain in the last decades, deep learning models are suffering from shortage of high-quality data for training. To alleviate this problem, people have applied the virtual-to-real paradigm in all visual tasks including classification, detection, segmentation, estimation, etc. In this part, we focus V2R solutions on classification, detection, recognition, and estimation.

1) Classification: Classification is one of the most fundamental tasks in machine learning. It is also one of the earliest areas using synthetic images. Specifically, we summarize the related literature from the viewport of parallel learning with prescription, prediction and prescription shown in Table I. Selected synthetic samples refer to Fig. 5.

Data augmentation (DA) is a popular virtual-to-real method that exerts operations on the original images to generate new label-preserving samples. These image operations, like translation, rotation, scaling, cutout, erasing, cut-and-combine, color-changing, and so on, are simple yet effective. In this part, we emphasize on two types of DA, i.e., the random DA and the automatic DA. For better comparison of different DAs, we select methods that share a common classifier WRN-28-10 and test on common dataset CIFAR-10 (100), as listed in Table I.

The random DA methods usually take an open-loop pipeline that applies simple operations on input images and generates augmented images to train classifiers. Devries and Taylor [21] proposed to randomly cutout patch from the original image, Zhong *et al.* [22] proposed to randomly erase area from the original image, while Han *et al.* [23] proposed a slightly complex operation to equally cut the image into two parts, change color of one part, and then combine the two parts into one image. Improvements on the evaluation metric of error rate of each method are list in the Table I, the later the better with same size of augmentation.



(a) Devries and Taylor [21] (b) Zhong *et al.* [22] (c) Han *et al.* [23]

(d) Cubuk *et al.* [24] (e) Ho *et al.* [25] (f) Li *et al.* [26]

Fig. 5. Selected examples of virtual-to-real methods for classification. (a) random cutout; (b) random erasing; (c) cut-aug-concat; (d) RL search; (e) PBA search; and (f) differentiable search.

While random augmentations take effects, they have shortcomings that generate out-of-domain samples which result in low efficiency for training and low performance in testing. To overcome this problem, Fawzi *et al.* [27] proposed an automatic and adaptive method to choose effective transformations among huge number of possible choices. The virtual data generation is modeled as an optimization process trying to find a slight transformation that results in maximal classification loss on each transformed sample. The authors implemented a trust-region optimization strategy and integrated it into the stochastic gradient descent algorithm to train deep neural networks. This is one of the earliest works of full parallel learning with prescription for high effective data generation. Tripathi *et al.* [28] proposed a task-oriented synthesizer

which is a trainable network that can be optimized to produce informative training samples based on output of the classifier. To ensure the synthesizer generating realistic data, the networks of both synthesizer and classifier are trained in an adversarial manner with a discriminator trained on real-world images. The proposed approach achieved baseline (classifier trained on MNIST) accuracy with less than half size of the original training data. Tran *et al.* [29] proposed a Bayesian formulation that treats new annotated training points as missing variables which can be generated according to the real data distribution. This method adaptively teaches a data generator as the training of classifier progresses by iteratively generating new training samples. The authors introduced a generalized Monte Carlo expectation maximization (GMCEM) for learning and gave an implementation by extending the generative adversarial network (GAN).

In addition to these adversarial methods, some works treat the DA as optimization problem solved by searching algorithms. Inspired by neural architecture search (NAS), Cubuk *et al.* [24] proposed AutoAugment. Instead of searching for optimized neural architecture, AutoAugment aims to automatically search for improved data augmentation policy for a target dataset with a fixed neural architecture. The search space is designed with 16 image operations including translation, rotation, shear, etc. The reinforcement learning is used to search best policies to augment the images and train a neural network whose validation accuracy is sent back as rewards to update the controller. AutoAugment achieved SOTA classification accuracy on several datasets, including ImageNet, but at the cost of hundreds of GPU hours. To reduce the computational cost, Ho *et al.* [25] proposed to apply the population based algorithm (PBA) to optimize the searching procedure in parallel, getting a 1000 × speedup compared to AutoAugment while keeping on-par accuracy. Further, Li *et al.* [26] proposed differentiable automatic data augmentation (DADA), a high efficiency pipeline by modeling the DA policy selection as a differentiable optimization problem solved with an unbiased gradient estimator, whose training time was reduced to only 0.1 h on CIFAR-10 with almost equal accuracy of the two former solutions. Comparing key factors of random DA with automatic DA as listed in Table I, we can conclude that automatic DA with prescriptions outperforms random DA in both accuracy and efficiency. A common feature of automatic methods is the closed-loop that adaptively improves the virtual data quality by introducing prescriptive signals like classification errors, discriminator outputs, or losses defined on distribution discrepancy.

Over-sampling (OS) is another commonly used method in addition to data augmentation, especially in the imbalanced dataset containing very few samples of minority classes. To balance the number of samples in different classes, Yan *et al.* [30] presented their work on synthetization of minority class samples. To overcome distribution mismatching between real and synthetic data, the authors proposed to take use of global geometric information based on optimal transport, so that the generated data follow a similar distribution to that of minority class samples. Loss value is used as feedback to guide efficient sampling, which forms a closed loop in the style as shown in Fig. 4. He *et al.* [31] proposed a novel oversampling approach by leveraging the GAN to model the data generating mechanism with non-linear latent representations. A Bayesian regularization guides the GAN to extract salient features controlled by a predefined structure in a human-in-the-loop manner.

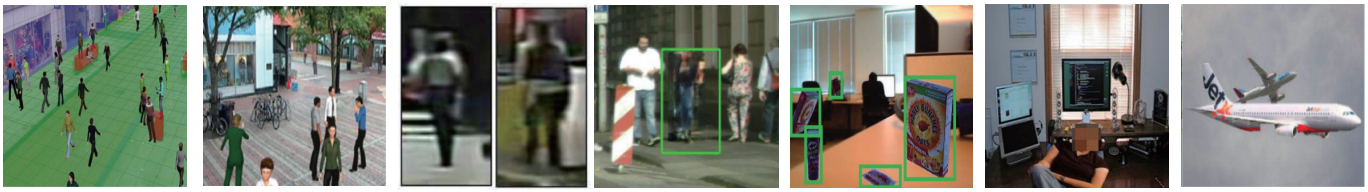
2) Detection: Detection is a basic visual task of identifying and correctly labeling objects at the pixel level. In this part, we look into the virtual-to-real methods from the viewport of parallel learning. We list related works in Table II for comparison, with selected examples shown in Fig. 6.

A group of methods use 3D models to avoid manually labeling the training data. 3D models were collected and then imported into 3D software like 3DS Max and Blender3D. Within 3D software, parameters including 3D model pose, camera pose, lighting, texture, and background, were randomly or intendedly configured to increase variations. From these 3D scenes, virtual 2D images with labels can be rendered out easily. This procedure provides authors convenience to generate virtual images on demand. For example, Peng *et al.* [41] proposed to train a few-shot deep CNN detector by augmenting the real data with synthetic images generated from 3D CAD models. They also introduced low-level cues, like color and shape, to further study the feature invariance of the representation. Cheung *et al.* [32] presented LCrowdV, a framework based on the unreal engine to generate videos of crowd with labels. The crowd movements were generated through simulation by altering number of pedestrians, density, behavior, flow, as well as lighting, viewpoint, etc. Detectors of both HOG (histogram of oriented gradient)-SVM (support vector machine) and Faster R-CNN were trained on the virtual data, combined with small part of real data. To deal with the zero-shot person detection, Hattori *et al.* [42] tried to generate training data through a 3D virtual environment, for a scene-specific scenario with a fixed camera in 3DS Max. In order to put virtual pedestrians in the reasonable regions, the authors manually labeled walls and obstacles in the scene. Images with pedestrians were generated through the camera with consideration of distortions to mimic the real surveillance cameras. Then an HOG + SVM pedestrian detector is trained on the virtual data and tested on real dataset. Their results show that learning model trained purely on virtual data outperforms models trained on real data or virtual-real mixed data. Later in [34], Hattori *et al.* generated a large virtual dataset by controlling variations in human appearance including gender, height, pose, and orientation based on only 139 different human models. In addition to the traditional classifier based detector, the authors proposed to use deep convolutional neural network, named as ScenePoseNet, for multiple tasks including detection, pose estimation and segmentation.

Instead of using 3D models, Dwibedi *et al.* [35] and Dvornik *et al.* [36] generated virtual data by 2D image editing and composition. The first step is collecting images from real datasets for both foreground and background (context). A mask of foreground was predicted and placed on background images to help paste instances into scenes with minimized pixel discrepancies at the boundaries. Dvornik *et al.* [36] pro-

TABLE II
COMPARISON OF TWO CLOSE RELATED WORKS ON VIRTUAL-TO-REAL IN OBJECT DETECTION

Method	Description		Prediction			Prescription	Highlights
	Synthesizer	Real data	Model	mAP (%)	Improve (%)	Loop	
Cheung *et al.* [32] ECCV'16	3D−2D Unreal 10 K	TownCenter	Faster R-CNN	72	7.3	Open guided by Menge simulation	Procedurally generate crowd video with label
Zhang *et al.* [33] NC'20	3D−2D 3DS Max -	TownCenter	Faster R-CNN	79.2	34	Open guided by environment change	Long-term learning in changing environment
Hattori *et al.* [34] IJCV'18	3D−2D 3DS Max 2.5 M	TownCenter	ScenePose Net 640 models	90	22	Open guided by scene geometry	Scene-specific pedestrian detector pure synthetic data
Dwibedi *et al.* [35] ICCV'17	2D−2D blending, DA 6728	GMU Kitchen	Faster R-CNN V+R	88.8	2.5	Open	Extract and paste patch level realism learn to ignore artefact
Tripathi *et al.* [28] CVPR'19	2D−2D blending -	GMU Kitchen	Faster R-CNN	89.8	3.5	Adversarial	Task-aware learning synthezier detector, discriminator
Dvornik *et al.* [36] ECCV'18	2D−2D Context-DA	VOC'12	ResNet50	65.9	1.3	Open guided by context	Learn what object and where to place in images
Wu *et al.* [37] ICCV'19	2D−2D CC-GAN	CUHK-Squares MIT-Traffic	Faster R-CNN PRC	FPPI 0.19	0.05	Adversarial	Generator and classifier compete two discriminators
Liu *et al.* [38] ICCV'19	2D−2D DetectorGAN	Chest X-ray Cityscapes	RetinaNet	0.124 0.613	0.112 0.011	Adversarial detector loss	GAN closed by discriminator for X-ray images
Martinson *et al.* [39] CVPR'21	3D−2D Blender GAN	xView	RetinaNet	0.24	0.02	Adversarial	Render 3D models to image composition for satellite images
Huang *et al.* [40] CVPR'22	nD−nD feature space	VOC'07'12 COCO	Faster R-CNN	65.5 19.8	0.6 0.8	Adversarial	Feature synthesizer from visual to semantic space



(a) Hattori *et al.* [34] (b) Zhang *et al.* [33] (c) Wu *et al.* [37] (d) Liu *et al.* [38] (e) Dwibedi *et al.* [35] (f) Tripathi *et al.* [28] (g) Dvornik *et al.* [36]

Fig. 6. Examples of synthetic images for object detection.

posed to learn a context model firstly using box annotation and applied it to determine what object to be inserted and where to be placed in the original image automatically. Data augmentation was used to ensure a diverse viewpoint or scale coverage. Martinson *et al.* [39] introduced a 3D−2D−2D method that firstly rendered the foreground from 3D models and then did image composition with a scene from satellite images as background. A further step was taken to input the composed image into a trained CycleGAN for improvement.

The aforementioned methods took a sequential pipeline: once a virtual data set was generated, the following steps are training a detector on the virtual data and testing the trained detector on real data set. On the contrary, Liu *et al.* [38] and Tripathi *et al.* [28] introduced a learnable adversarial framework that integrated data synthesizer, detector, and discriminators. With detector loss as feedback, the synthesizer is guided to generate hard examples that improve the detector in an adversarial style. The discriminator helps to improve realism and make the generated images conform to the real image distribution. The three components, synthesizer, detector and discriminator, are trained similarly to typical GANs. Real images are necessary to train the discriminators. The GAN framework enforces the synthesizer to generate images more effectively to improve detection performance. Here effectively means generating harder examples and neglecting normal examples without contribution to improve detector performance. The experiments in [28] showed that the detector got better accuracy with only half number of images compared to previous work.

In order to generate more realistic virtual images, Wu *et al.* [37] also proposed an adversarial framework in which a post-refinement classifier (PRC) is introduced following the Faster R-CNN detector. The PRC and the synthesizer work together to compete with a class-conditional discriminator and a class-specific discriminator. These four networks are trained in a manner that facilitates both pedestrian synthesis and detection in semi-supervised setting. Experiments validated the proposed model by significantly improving the detector performance with state-of-the-art results on multiple real dataset. Zhang *et al.* [33] construct a prescriptive learning connecting

virtual and real world dealing with long term learning problem that environment changing cannot be neglected. The virtual world is a 3D scene in 3DS Max with configurable parameters including lighting. The difference between the virtual and real background, for example changes in day and night, was fed back to update virtual scene in order to approach the real world.

The image composition procedure inserting the foreground into background will inevitably introduce boundary artefacts, which may be recognized as a feature by the detector. That is, the virtual data introduced artificial features that does not belong to real data, which have been verified to be harmful to the performance of detector. Pixel blending along foreground boundary is a common solution to reduce artefacts. Tripathi *et al.* [28] proposed another strategy by generating additional hallucinated artefacts in the background images, making the detector invariant to artefacts in the synthesized images.

Instead of augmenting images, a new way is to synthesize at feature level. Huang *et al.* [40] proposed a robust region feature synthesizer with two components, i.e., an intra-class semantic diverging component to obtain diverse visual features, and an inter-class structure preserving component to avoid the synthesized features too scattered. On PASCAL VOC and COCO dataset, this approach achieved the state-of-the-art performance for zero-shot object detection.

3) Recognition: For virtual-to-real in visual task of recognition, we include text recognition and face recognition respectively as follows.

a) Text recognition: In this part, we summarize works on text/word recognition in natural images, examples can be found in Fig. 7.

(a) Wang *et al.* [43] (b) Gupta *et al.* [44] (c) Zhan *et al.* [45] (d) Long and Yao [46]

Fig. 7. Examples of synthetic text in natural scenes.

We start with an early work in 2011, when Wang *et al.* [43] proposed a pipeline based on SVM training on SYNTH, a synthesized dataset that containing 1000 images per character of 40 font types. Gaussian noise and random affine deformation were also applied to add divergency. In recent years, deep neural networks have been used as task models instead of SVMs. The key problem to generate virtual text in natural background is to determine the proper place and pose of the text. Semantic priors, from both object segmentation and 3D scene structure, may help to generate virtual dataset. Gupta *et al.* [44] designed an engine that adds text to images with natural backgrounds. Their sample texts, collected through Google search, were placed into the image according to estimated depth and geometry, then blended into the background using Poisson image editing. 8000 high realistic images were synthetized automatically as a dataset called SynthText. Zhang *et al.* [45] proposed to improve text placement according to semantic coherence and visual saliency, which were provided as prior from semantic segmentation. In addition, the color and brightness of embedded texts were determined by learning from real scene in an adaptive manner. Experiments on five public datasets show its superior performance. Differently from aforementioned methods, Long and Yao [46] introduced an image synthesis method named as UnrealText. The process begins with a 3D scene in which the region is proposed to place the sampled text. A camera moving around in the 3D scene produced a virtual dataset with 600 K text images. Using the 3D graphics engine unreal, the scene and the text can be rendered as a whole not only for realistic appearance but also with annotations.

b) Face recognition: Face recognition is another visual task intensively taking advantages of synthetic images. We list related works in Table III for comparison, and show typical samples in Fig. 8 for brief overview.

(a) Yu and Grauman [47] (b) Hu *et al.* [48] (c) Shi *et al.* [49] (d) Qiu *et al.* [50]

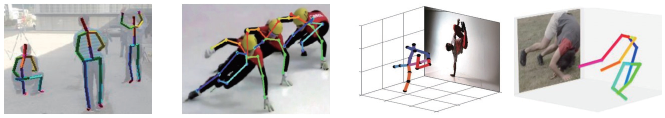
Fig. 8. Examples of synthetic faces.

A partial work of [47] is to densify supervision for face recognition with synthetic images. The main idea is generating identities, i.e., similar faces but with slight difference, of real sample by modifying attributes through the Attribute2-Image engine. Hu *et al.* [48] proposed to generate new face images by composing different face parts of one or multiple persons based on a given real dataset. The experiments showed that a model trained on 10 K images with the proposed method had similar performance as it was trained on 500 K images.

Shi *et al.* [49] proposed URFace with the ability to learn universal representation. URFace firstly augmented the training data by introducing variations of blur, occlusion and head pose. A confidence-aware identification loss was designed to encourage the model to learn from hard examples, through sub-embeddings spilt from feature vectors. Qiu *et al.* [50] proposed SynFace with identity mixup (IM) and domain mixup (DM) as two features. IM is based on DiscoFaceGAN with purpose to enlarge intra-class variations, while DM utilizes a large number of virtual data with a small number of real data, aiming to reduce domain gap between synthetic and real face data. On the observation that virtual data and real data have a natural discriminability in fixed form, denoted as modality, Zhai *et al.* [51] proposed three methods to remove the modality for better performance when utilizing virtual data.

4) Estimation: Estimation from 2D images is a broad topic covering various specific tasks, e.g., pose estimation and depth estimation. In this part, we mainly focus on human/animal pose estimation from monucular camera, related works are list in Table IV, samples are shown in Fig. 9.

Rogez and Schmid [52] proposed an image synthesis engine that took in both images with 2D pose annotations and 3D



(a) Mehta *et al.* [54] (b) Hoffmann *et al.* [55] (c) Li *et al.* [56] (d) Gong *et al.* [57]

Fig. 9. Examples of pose estimation using synthetic data.

motion capture data. The core to this method is that, given a 3D pose, a set of image patches whose 2D pose locally matches the projected 3D pose for all joints are collected and then combined as a new image by stitching local patches constrained by kinematical rules. The output images, with 3D pose annotations, are used to train CNN for 3D pose estimation. Chen *et al.* [53] also took use of MoCap data and 2D annotated pose images, but instead of compositing 3D annotated images directly, the authors firstly built a statistical human model based on the available data. Then a large scale virtual images dataset with pose annotations was generated by sampling from the statistical model, which was rendered into background with clothes. A further domain adaptation was introduced during training CNN models by extracting common features of both virtual and real data. Focusing on multi-person estimation with occlusions, Mehta *et al.* [54] created MuCo-3DHP, a synthetic data set by compositing multiple 2D person images with 3D pose annotations from multi-view motion captures. Deep models trained on this dataset achieved state-of-the-art performance when tested on the 3D annotated multi-person test dataset MuPoTs-3D.

All aforementioned works generated a number of virtual data with annotations and trained their deep models with performance improvement. However, people may have questions. The first, which kind of virtual data is better, purely synthetic or composed? And second, does every generated image contribute to the improvement of leaning model equally? Or in other words, is there any redundancy in the virtual dataset? To answer these questions, Hoffmann *et al.* [55] designed experiments for pose estimation. For the first question, the authors created two datasets, one for purely synthetic humans and the other one with real data augmented by synthetic humans. Experiments showed that models trained on mixed data with domain stylization get best generalization performance. For the second question, the authors found that, through experiments, some data contribute more than others in different training stages. Based on this observation, an adversarial training method with a teacher and a student was designed to optimize the sampling process when generating virtual humans. From viewpoint of parallel learning, this method introduces prescriptive learning with feedback from estimator to virtual data generator, a closed loop for more efficient learning.

Taking a non-stationary view toward training data, Li *et al.* [56] proposed an augmentation method by generating human data evolutionally. A human skeleton was represented as a hierarchical tree structure that each joint is a node with mutable parameters. An evolutional process including crossover, mutation and natural selection was operated to augment the Human3.6M dataset with a binary fitness function. A dedicated 3D pose estimator network called TAG-Net was trained on evolved dataset and achieved not only SOTA accuracy on public benchmark but also generalized better on unseen dataset. Results also showed that this evolutional method can reduce data bias efficiently.

Gong *et al.* [57] presented PoseAug, a framework that automatically learns to generate virtual training data taking pose estimator errors as guiding signals. The human body was represented by parameters adjusted through differentiable operations. The data generator took in estimation errors as feedback to generate rare poses based on this differentiable pipeline. In this way, the data generator and estimator were jointly trained in a closed loop, which becomes a typical closed-loop prescriptive learning. Discriminators were necessary to ensure that the generated human poses fall in a plausible range by measuring local joint-angle defined in kinematic chain space (KCS).

5) Summary: CV is the most mature area of V2R, spanning traditional machine learning and deep learning models, and the methods involved are relatively diverse, with deeper perspectives and more adequate experiments. DA based on image processing has become a standard operation in deep learning. 3D−2D methods were more common in the early days, but with the development of generative models in deep learning, especially the widespread use of adversarial generative networks, 2D−2D high quality image generation has also taken an important place. In aspect of prescription, early open-loop DA methods have demonstrated the effectiveness of data augmentation, while the rapidly developing differentiable closed-loop feedback, prior knowledge-based bootstrapping, and adversarial generative networks have gradually shown advantages in recent years.

B. Virtual-to-Real in Natural Language Processing

Unlike CV, natural language processing (NLP) deals with 1D data such as audio, speech, and text, which has a discrete and structural nature compared with images. Like CV, the success of NLP tasks depends on deep neural models and a large number of labeled data. NLP tasks are also hindered by data scarcity in practice, where synthetic data are essential in complementary to the real data. In this section, we firstly look into virtual-to-real methods by summarizing popular data augmentation techniques in NLP, then introduce selected applications.

1) Overview of Data Augmentation Methods in NLP: A number of data augmentation methods in NLP have emerged in recent years and there are different viewpoints to categorize them. Chen *et al.* [58] summarized augmentation methods as four types, i.e., token-level, sentence level, adversarial and hidden-space augmentations. Feng *et al.* [59] classified representative augmentation techniques into three groups, i.e., rule-based techniques, example interpolation techniques and model-based techniques. While Li *et al.* [60] divided augmentation methods into three categories including noising-based methods, paraphrasing-based methods and sampling-based methods.

These techniques have been used to alleviate low resources problems in NLP tasks ranging from audio recognition, text classification, to machine translation, abstractive summarization, question and answering, dialogue, etc. It is impossible to

TABLE III
COMPARISON OF FACE RECOGNITION USING SYNTHETIC DATA

Method	Task	Description			Prediction		Prescription	Highlights
		Dim	Virtual	Real	Model	Improve (%)	Loop	
Wang *et al.* [43] ICCV'11	Text	2D−2D	SYNTH 1.4 M	ICDAR'11 SVT	SVM	accuracy 62 (5) 57 (1)	Open	End-to-End scene text recognition
Zhan *et al.* [45] ECCV'18	Text	2D−2D	- 5 M	ICDAR'13 SVT	CRNN	CRW 87.1 (55.9) 96.7 (34.6)	Open guided by semantic	Visual attention semantic coherence adaptive text appearance
Long and Yao [46] arXiv'20	Text	3D−2D Unreal	UnrealText 600 K	ICDAR'15 SVT	ASTER	accuracy 39.1 (0.7) 40.3 (0.5)	Open guided by scene information	Place text using 3D scene information
Yu and Grauman [47] ICCV'17	Face	2D−2D	LFW-Synth 5000	LFW-10 PFSmile	RankSVM DeepSTN	75 (8) 84.36 (3.5)	Open guided by semantic	Densify supervision slightly modify attributes Attribute2Image
Hu *et al.* [48] TIP'18	Face	2D−2D	synthesis 32X	LFW	CNN-L	95.77 (4)	Open	Composing different face parts into one image
Shi *et al.* [49] CVPR'20	Face	2D−2D	URFace 4.8 M	LFW TinyFace	ResNet 100-layer	99.78 (0.03) 63.89 (17.14)	Adversarial	Learning a universal face representation variation augmentation
Qiu *et al.* [50] ICCV'21	Face	2D−2D	SynFace 0.5 M	LFW	LResNet50E -IR	91.97 (4.37)	Adversarial	Synthetic face by identity mixup and domain mixup
Zhai *et al.* [51] AAAI'21	Face	2D−2D	DMFR MS-Celeb-1 M 2.2 M	CP-IJB-C CQ-IJB-C	ResNet 100-layer	81.92 (−2.4) 84.77 (8.62)	Open	Meta learning removing modalities of synthetic data

TABLE IV
COMPARISON OF POSE ESTIMATION USING SYNTHETIC DATA

Method	Description			Prediction		Prescription	Highlights
	Dim	Virtual	Real	Model	Improve (%)	Loop	
Rogez and Schmid [52] ECCV'16	2/3D−2/3D	MoCap 10K+	Human3.6M	Adapted AlexNet	Error 88.1 (20.2)	Open	Combine new image by stiching local patches with kinematical constraint
Chen *et al.* [53] 3DV'16	2/3D−2/3D	MoCap 5M+ DA	Human3.6M	Modified AlexNet VGG	Error 0.044 (0.01) 0.04 (0.016)	Open	Genrate annotations by sampling from statistical human model
Mehta *et al.* [54] 3DV'18	2/3D−2/3D	MoCap MuCo-3DHP	Human3.6M MuPoTS-3D	ResNet-50 2D&3D Pose	MPJPE 69.6 (10) 132.5 (13.5)	Open guided by location-maps	Single-shot multi-person with occlusions from multi-view MoCap
Hoffmann *et al.* [55] GCPR'19	2/3D−2/3D	MoCap SMPL+H Blender	MPII multi-person	OpenPose	mAP 78.4 (0.7)	Adversarial	Adversarial training teacher v.s. student thorough ablation
Li *et al.* [56] CVPR'20	2D−2/3D	Evolved Human3.6M	Human3.6M 3DHP	TAG-Net	MPJPE 63.5 (7.5) 99.7 (13.5)	Feed-back evolution	Evolution strategy with binary fitness function
Gong *et al.* [57] CVPR'21	2D−2/3D	PoseAug	Human3.6M 3DHP	SemGCN 4 models	MPJPE 50.2 (0.7) 73.0 (16.4)	Feed-back adversarial error	Differentiable generator with constraints from discriminators

cover the tremendous literature in one single paper, so that we select two tasks, automatic speech recognition (ASR) and neural machine translation (NMT), that typically represent the virtual-to-real NLP paradigm from viewpoint of parallel learning.

2) Selected Virtual-to-Real Applications in NLP:

a) Automatic speech recognition (ASR): We list related works in Table V and start from the pioneering work of Tjandra *et al.* [61] who put forward a sequence-to-sequence deep learning framework that integrates both listening and speaking, inspired by the closed-loop speech chain mechanism [62]. Specifically, an automatic speech recognition (ASR) model responses for listening, i.e., transcribes the unlabeled speech features, and a text-to-speech synthesis (TTS) model takes charge of speaking, i.e., reconstructs the original speech waveform based on the text output from ASR. The ASR also tries to reconstruct the original text transcription based on synthesized speech from TTS, which iteratively forms a closed-loop that simultaneously improves both speech perception and production, as shown in Fig. 10. This work provides a way to uti-

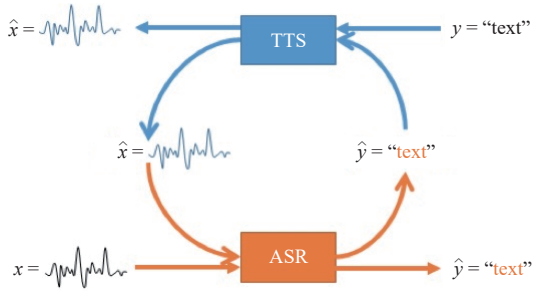


Fig. 10. Principle of closed loop by integration of ASR and TTS (Tjandra *et al.* [61]).

lize unlabeled data in speech recognition task. Further in [63], [64], Tjandra *et al.* introduced a straight-through estimator in ASR so that reconstruction loss can be back-propagated the same way as in TTS. This solved the non-differentiable problem of discrete tokens output by ASR and resulted in a full end-to-end framework.

Instead of converting between source and target, Hayashi *et al.* [65] proposed to use hidden states by training a text-to-encoder, in scenarios of ASR given a large number of unpaired texts. Using hidden states, instead of acoustic features, has advantages of fast attention learning and avoiding speaker dependencies.

Other works tried to improve the closed-loop with different concerns. Rossenbach *et al*. [66] presented a work extending the state-of-the-art attention-based ASR with a TTS trained only on the ASR outputs. Zheng *et al*. [67] proposed to provide synthetic audio for out-of-vocabulary (OOV) words by a TTS. In such a way, the authors boosted the recognition accuracy of a recurrent neural network transducer (RNN-T) on OOV words by using the extra audio-text pairs, while maintaining the performance on the non-OOV words. Hu *et al*. [68] paid attention to the gap between synthetic and real data distributions. The authors proposed a rejection sampling algorithm with batch normalization for both real and synthetic samples, which improves the ASR performance compared with simply using synthetic data. Hori *et al*. [69] introduced a cycleconsistency loss based on the Text-To-Encoder reconstruction error, instead of the raw speech. Wang *et al.* [70] improved training on synthetic data by promoting consistent predictions in response to real and synthesized speech. With this method, models trained on 460 h of LibriSpeech augmented with 500 h of transcripts (without audio) perform on par with a system trained on 960 h of transcribed audio. This suggests that reliance on transcribed audio can be cut nearly in half when sufficient text is available. Chen *et al.* [71] proposed to improve acoustic diversity of TTS outputs by combining the GAN and multi-style training (MTR). The authors also presented a contrastive language model-based data selection technique to improve the efficiency of learning from unspoken text. Du and Yu [72] proposed to train a TTS system with speaker representations from a variational auto-encoder (VAE). Such a speaker augmentation method enables TTS to generate unseen new speakers via sampling from the trained latent distribution. Fazel *et al.* [73] proposed SynthASR with a multi-stage training strategy to avoid catastrophic forgetting by mixing of weighted techniques, including multi-style training, data augmentation, encoder freezing, and parameter regularization. SynthASR showed good transferability to new applications.

b) Neural machine translation (NMT): Neural machine translation (NMT) has dominated in machine translation in recent years, but it still struggles in low-resource or out-of-domain scenarios. There are several good methods to address this problem, e.g., DADA [74] which takes an attack-defend adversarial method to improve robustness of NMT models, and AdvAug [75] which trains NMT models using virtual sentence's embeddings in seq2seq learning, etc. Here we intend to focus on a group of Paraphrasing-based methods as follows.

Translation has a feature of duality that translates from target to source language is an inverse process of translating from source to target. To take use of this feature, He *et al.* [76] published an impressive work of dual learning for machine translation in 2016. Dual learning regards translation as a game between two agents. One agent responds for primal task to translate from source to target, and the other agent responds for a dual task to translate from target to source. The primal and dual tasks form a closed loop that each side updates iteratively based on the reconstruction error feedback from the other side. In such a way, the two agents teach each other through reinforcement learning using policy gradient method. Dual learning can be regarded as a special parallel learning case involving mutual prescriptive learning process. We add a figure according to the original paper as shown in Fig. 11.

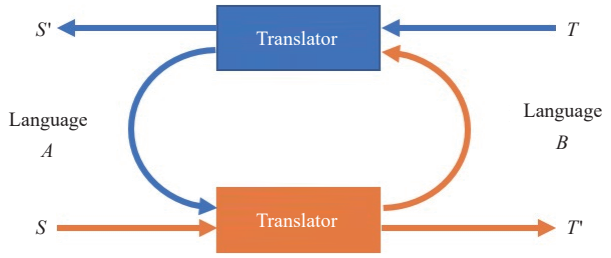


Fig. 11. Principle of closed loop by integration of bi-directional translation.

In a following work [77], the authors extracted the dual learning as general paradigm that is suitable for paired tasks like speech recognition v.s. text-to-speech, and image classification v.s. image generation, in addition to language translation. To improve the training efficiency of RL based dual learning translation [76], Wang *et al.* [78] proposed to connect the probability of a given target-side monolingual sentence to the conditional probability of translating from a source sentence to the target one, by leveraging the dual learning translation model to sample several most likely source-side sentences, thus avoid enumerating all possible candidate sentences of source language. This can be viewed as a transfer learning that transfers the knowledge in the dual model (target-to-source) to boost the training of the primal model (source-to-target).

Zhang *et al.* [79] also presented methods by jointly training source-to-target and target-to-source NMT models with a joint EM optimization. In a similar way, Lample *et al.* [80] also

TABLE V
COMPARISON OF SELECTED WORKS ON VIRTUAL-TO-REAL IN SPEECH RECOGNITION

Method	Description		Prediction		Prescription	Highlights
	Virtual	Real	Model	Improve (%)	Loop	
Tjandra *et al.* [61] ASRU'17	TTS Tacotron revised	BTEC Google TTS	Sequence-to-sequence attention	CER 5.4 (4.6)	Closed ASR+TTS	Speech chain deep learning speaking while listening
Tjandra *et al.* [63] ICASSP'19	TTS Tacotron revised	Wall Street Journal	Sequence-to-sequence attention	CER 5.7 (0.73)	Closed ASR+TTS	Differentiable ASR of previous work
Tjandra *et al.* [64] Interspeech'20	TTS based on Transformer	ZeroSpeech 2019 Track	Transformer based VQ-VA	ABX error rate 21.6 (2.2)	Closed ASR+TTS	ASR TTS Transformer zero text
Hayashi *et al.* [65] SLT'18	TTE Tacotron2	LibriSpeech	Sequence-to-sequence language model	CER 10 (0.4) WER 22 (0.9)	Open	End-to-end ASR back-translation-style hidden states
Rossenbach *et al.* [66] ICASSP'20	TTS SpecAugment perturbation	LibriSpeech	LSTM+MLP attention	WER 7.9 (1.4)	Open	Synthetic audio by TTS improve SOA ASR
Zheng *et al.* [67] ICASSP'21	TTS OOV mixed v+r	Train19 DEV EVAL	RNN-T LSTM+MLP	NWER 1.28 (1.54)	Open	Out-of-vocabulary word recognition
Hu *et al.* [68] ArXiv'21	Synt++ TTS SpecAugment	LibriSpeech-100h LibriSpeech-960h	ESPNet	WER 4.0 (3.7) WER 2.4 (0.5)	Closed rejection sampling	Synthesizer using rejection sampling double BN

proposed a dual framework but mapping both source and target domains into a latent space where back-translations are applied from both sides to get reconstruction error. Niu *et al.* [81] proposed to combine both the primal and dual model into a single model for bi-directional translations with much higher efficiency than two uni-directional models. Wang *et al.* [82] extended the dual learning to multi-agents framework that translates between multiple languages instead of one pair of source and target language. Ahmadnia and Dorr [83] presented a round-trip training approach that shares a similar idea with dual learning. Zheng *et al.* [84] presented mirror-generative NMT (MGNMT) which is also a single unified model like [81] but additionally introduced two language models to collaborate with the unified model during decoding.

3) Summary: In the field of NLP, the complementarity between tasks provides for the construction of closed-loop, bi-directional guidance. Dual learning for NMT tasks is the representative approach. This idea has been further extended and is now applied to multimodal scenarios, e.g., image caption and image generation are cross-modal complementary tasks. In recent years, pre-trained big models in the field of NLP have developed rapidly, using unsupervised learning to extract knowledge from massive number of unlabeled data for several tasks such as recognition and generation. It is foreseeable that pre-trained big models will play an important role in V2R applications.

C. Virtual-to-Real in Robotics

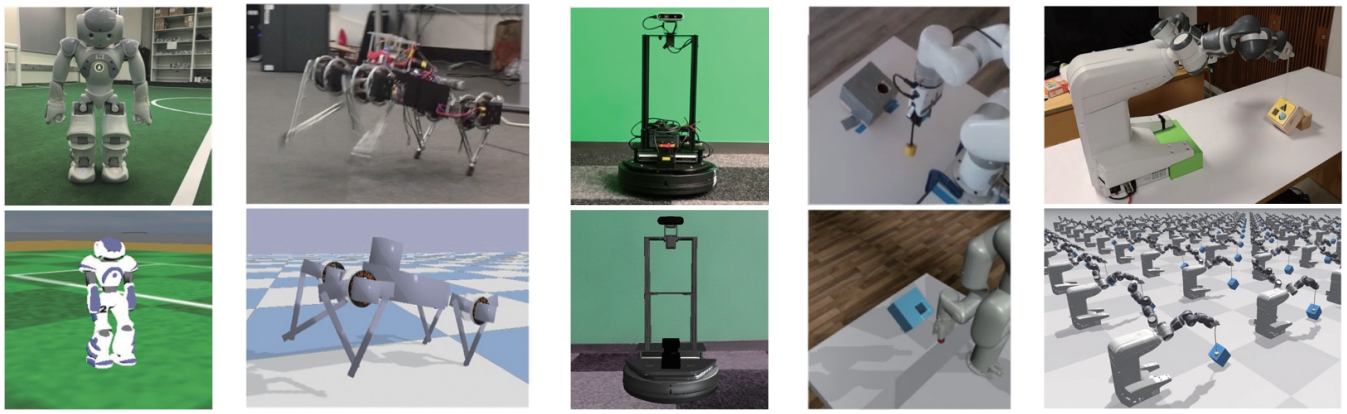
Learning a policy for robotic control is another area that deeply involves virtual-real interactions, for the same reason to get more data efficiently in a safe manner. Differently from syn-to-real, the paradigm used in robotic control is referred as sim-to-real, where dynamic simulations instead of static synthetic images are defined as source domain. The dynamic control is usually modeled as a partially observable Markov decision process (POMDP) that is solved by methods including imitation learning and reinforcement learning. Same as syn-to-real, closing the domain gap between the simulation and real world is a key research topic yet remains as an open problem. Nevertheless, many progresses in sim-to-real have been reported.

In this section, we summarize sim-to-real works on robotic control reported in recent years from viewpoint of parallel learning. According to the type of control tasks, we organize the related literature in two categories, i.e., manipulation and navigation (including locomotion), with selected examples shown in Fig. 12.

1) Manipulation: In this part, we introduce sim-to-real methods in zero-shot scenarios, succeeded by one-shot and few-shot settings. Selected works are listed in Table VI for comparison.

With zero-shot settings, Tobin *et al.* [85] proposed to generate simulated images by randomizing rendering configurations. For a task of grasping tiny sphere, the proposed method successfully transfers a deep neural network (DNN) trained only with simulated RGB images to real world robotic control. For vision-based robotic tasks, the controller makes decisions depending on results of visual process. To take advantages of this two-step pipeline, Yan *et al.* [89] proposed a framework which comprises a vision module succeeded by a control module. The vision module performs object segmentation, and the control module, which is a closed-loop DNN controller, takes in binary segmentation mask to train the control policy by imitation learning. This decoupled manner equals mapping both virtual and real images into a common space, bridging the gap between virtual and real environment. The method achieved a 90% success rate in grasping a tiny sphere and the controller can generalize to unseen scenarios with moving targets.

Sadeghi *et al.* [88] focused on learning viewport invariant



(a) NAO [108] (b) Minitaur [109] (c) LoCoBot [116] (d) UFactory [94] (e) ABB Yumi [93]

Fig. 12. Selected examples of robotic Sim2Real for locomotion (a) and (b), navigation (c), and manipulation (d) and (e). The upper are real robots and the lower are the corresponding virtual worlds.

TABLE VI
COMPARISON OF SELECTED WORKS ON VIRTUAL-TO-REAL IN ROBOTIC MANIPULATION

Method	Task	Description		Prediction		Prescription	Highlights
		Virtual	Real	Percept	Controller	Loop	
Tobin *et al.* [85] IROS'17	Loca	MuJoCo DR rendering	Fetch robot	RGB VGG-16	SL SGD	Open	Transferring from low-fidelity simulated camera images
Pinto *et al.* [86] RSS'18	Pick push	MuJoCo DR	Fetch robot	RGBD-state CNN-MLP	RL AAC	Open	Asymmetric actor critic training
Golemo *et al.* [87] PMLR'18	Push strike reach	MuJoCo	Poppy Ergo Jr	State	LSTM RL PPO	Closed	Real data grounded RNN predict the discrepancies
Sadeghi *et al.* [88] CVPR'18	Reach	Bullet DR finetune	Kuka IIWA	RGB CNN	LSTM RL SL CEM	Open	View invariant visual servoing by recurrent control
Yan *et al.* [89] NeurIPSw'17	Grasp	Gazebo DR	Baxter	RGB-state eye-in-hand CNN	IL DAGGER	Open guided by expert	Decomposed vision control segmentation as interface real background virtual object
Bousmalis *et al.* [90] ICRA'18	Grasp	Bullet Blender DR-DA-Mix	Kuka IIWA	RGB VGG	GraspGAN	Adversarial	Vision based grasping by GAN
James *et al.* [91] CVPR'19	Grasp	Bullet RCAN	Kuka IIWA	RGB cGAN	RL QT-Opt	Adversarial	Real-to-virtual randomized-to-canonical adaptation
Kim *et al.* [92] ICRA'22	Grasp	IPC-GraspSim DR	ABB YuMi gripper jaw	State	Contact dynamic model	Open guided by IPC	Accurate collision deformation model reduce domain gap
Chebotar *et al.* [93] ICRA'19	Swing-peg-in-hole	Nvidia Flex DR SimOpt	ABB Yumi Franka Panda	Position DART	RL PPO	Closed roll-out	Parallel policy learning real world roll-out with guided DR
Du *et al.* [94] ICRA'21	Peg-in -Hole etc.	Deepmind Control Suite DR	UFactory xArm 7	RGB	RL SAC	Closed few-shot	Auto-Tuned simulator using SPM
Matas *et al.* [95] CoRL'18	Fold cloth	Pybullet DR	Kinova Mico	RGB-state fixed camera	RL DDPG	Open	Deformable object manipulation
Jeong *et al.* [96] ArXiv'19	Cube stacking	MuJoCo DR	Sawyer robotic	RGB-state fixed camera	RL MPO	Open	Self-supervision DA and time-contrastive
Chang and Padif [97] IRC'20	DRC Plug	Gazebo SI	Kinova Jaco	RGB-D	inverse kinetics Open RAVE	Closed	Vision based sim2real2sim dynamic cable
Allevato *et al.* [98] PMRL'20	Bounce ball	PyBullet SI	Kinova Jaco Robotiq85	State parameters	SL	Closed	Iterative residual tuning
OpenAI [99] IJRR'20	Cube rotate	MuJoCo Unity3D DR	Shadow Dexterous	RGB CNN	RL PPO	Open	Trained entirely in parallel simulation
Power and Berenson [100] IRAL'21	Rope manip	Gazebo SI-DR	KUKA iiwa	RGB-D Kinect state	MPC MPPI	Closed	Learned visual similarity predictive control deformable object

visual servo. They presented a framework which also disentangled perception from a memory-based recurrent controller learned from synthetic images. The model adapted from virtual side enables a real robotic arm to interact with unseen objects from novel viewpoints. Pinto *et al.* [86] introduced the asymmetric actor critic algorithm in which the critic is trained on full states while the actor (or policy) is trained on images, to speed up learning process. Combined with domain randomization, this method achieved sim-to-real transfer in zero-shot scenarios. In [90], Bousmalis *et al.* proposed GraspGAN, a method by extending domain adaptation and randomization to train a grasping robot based on raw monocular RGB images. Using unlabeled real data, this method is comparable with those using millions of labeled real-world samples.

Generalizing to new environments and crossing various robot configurations are important goals for sim-to-real applications. Scherzinger *et al.* [101] proposed to learn contact skills from human demonstrations through simulation, using an LSTM (long short-term memory) network. This framework comprised a real robot and its virtual twin to solve inverse kinematics problem. The generated sequences of forces and torques in task space can be generalized to new tasks across different joint configurations of the robot. OpenAI [102] proposed a zero-shot automatic domain randomization (ADR) method that transfers from simulation to unprecedented complex real world, by increasing difficulty of distribution over randomized environments. The ADR method solved a Rubik's cube with a humanoid robot hand. The randomized-to-canonical adaptation networks (RCANs) proposed by James *et al.* [91] is also a zero-shot method to cross the visual-reality gap by translating randomized rendered images, as well as real images, into their equivalent canonical versions. Using RCAN and QT-opt, they improved the performance from 36% to 70% for grasping unseen objects.

While most work focused on manipulating rigid objects, Matas *et al.* [95] introduced deformable object manipulation from sim-to-real, which overcame the problem of large configuration space of deformable objects. Trained fully in simulation with randomization, the agent was successfully deployed in the real world in a zero-shot manner. Kim *et al.* [92] proposed IPC-GraspSim, that introduced an accurate contact algorithm IPC (incremental potential contact) into the simulation for sim-to-real transfer. They testified through experiments that more accurate contact simulation with proper deformation parameters helps to reduce reality gap for grasping objects, improving F1 score by 0.09 over Isaac Gym.

One-shot or few-shot strategy brings further benefits when real environments are available during training stage. Golemo *et al.* [87] proposed neural-augmented simulation (NAS), a method that augments the simulator by training an RNN based on the differences between simulated and real robot trajectories. This can be viewed as a system identification (SI) method that closed the domain gap for a better sim-to-real transfer. Chebotar *et al.* [93] proposed a few-shot method to match the simulation with real world by adapting the simulation parameters through real world rollouts during training. The training process is in a distributed manner with randomized parameters. Allevato *et al.* [98], [103] presented TuneNet, an efficient system identification method that tunes the simulator parameters to match real world using iterative residual tuning (IRT). The system was trained via supervised learning over an auto-generated simulated dataset, with minimal real-world observations. Experiments showed that TuneNet helped a robot in real world to perform a dynamic manipulation with a new object, after one-shot observation.

Chang and Padif [97] proposed simulation-to-real-to-simulation (Sim2Real2Sim), a new strategy to bridge reality gap. For a manipulation task, this strategy starts training with simulation using rough virtual environment with estimated models. Then, sim-to-real transfer is taken to compare the performance between virtual and real robots. Finally, models in simulation is updated based on the differences between simulated and real worlds. Such a closed-loop pipeline realized the prescriptive learning in parallel learning. Heiden *et al.* [104] also made efforts to identify parameters of highly nonlinear and underactuated systems. The authors proposed to approximate a posterior distribution over simulation parameters given real sensor measurements based on Bayesian inference. Physical experiments demonstrated that this technique could identify symmetries between the parameters and provide highly accurate predictions.

Domain randomization depends on prior knowledge and engineering efforts to decide how much to randomize the parameters for robust sim-to-real transfer. Breyer *et al.* [105] introduced an adaptive learning mechanism that learns gradually from simple to complex tasks, with prioritized sampling scheme. Du *et al.* [94] proposed search param model (SPM), a method to automatically tune simulator parameters to match the real world using only raw RGB images from real world. Given a sequence of observations, SPM predicts whether the parameters are higher or lower than the true values. Experiments on robotic control in real world demonstrated improvement over simple domain randomization. Shashua *et al.* [106] also demonstrated a strategy to learn from both simulation and real environments simultaneously, by maintaining a replay buffer for each environment with which the agent interacts. Power and Berenson [100] proposed learned visual similarity predictive control (LVSPC), a data-efficient online learning method to control systems with complex dynamics and high-dimensional state spaces from images. Experiments of both rigid and deformable objects showed comparable performance to state-of-the-art reinforcement learning methods but with much fewer data.

2) Navigation: In mobile tasks like navigation and locomotion, robots face more complex environments, which makes sim-to-real transfer more challenging. For better comparisons, we select typical works and list key features in Table VII.

For bipedal locomotion, efforts to cross the domain gap between virtual and real domains can be traced back to Farchy *et al.* [107] who introduced grounded simulation learning (GSL). GSL tries to improve robot learning in virtual world by transferring physical state from the real system to update the virtual system such that the updated virtual system becomes closer to the real system. The method to make the simulator approach the real world is referred to as grounding. GSL starts with an imperfect simulator in which a policy is

TABLE VII
COMPARISON OF SELECTED WORKS ON VIRTUAL-TO-REAL IN ROBOTIC NAVIGATION

Method	Task	Description		Prediction		Prescription	Highlights
		Virtual	Real	Percept	Controller	Loop	
Farchy *et al.* [107] AAMAS'13	Bipedal locomotion	SimSpark	NAO	State	Walk engine M5P tree regression	Closed SI	Grounded simulation learning openparams human in loop
Hanna and Stone [108] AAAI'17	Bipedal locomotion	SimSpark Gazebo	NAO	State	Walk engine RL MLP	Closed	Grounded action transformation OpenParams
Tan *et al.* [109] arXiv'18	Quadruped locomotion	PyBullet DR	Minitaur	State	RL PPO	Closed SI	Controller latency space
Yu *et al.* [110] IRAL'20	Quadruped locomotion	PyBullet	Minitaur	State	RL ARS SO	Closed roll-outs	Meta learning latent space strategy optimization
Hwangbo *et al.* [111] SR'19	Quadruped locomotion	RaiSim	ANYmal	State	RL TRPO	Closed SI	Parameter identification dynamic actuator policy by simulation
Lee *et al.* [112] SR'20	Quadruped locomotion	RaiSim DR Zero-Shot	ANYmal	State	RL TRPO	Open guied by learning	Privileged teacher proprioceptive student terrain curriculum
Hong *et al.* [113] IJCAI'18	Navigation	Unity3D Zero-Shot	real robot	RGB DeepLab	RL A3C	Open	Decoupled visual-control via segmentation
Zhang *et al.* [114] IRAL'19	Navigation	Gazebo Carla Zero-Shot	Turtlebot3 Bulldog	RGB CycleGAN	RL A3C	Closed roll-outs	Translate real to sim task agnostic
Mitriakov *et al.* [115] IRCA'22	Navigation	Gazebo DA Zero-Shot	Absolem Jaguar	State	RL PPO	Open	KL divergence multiple tasks multiple robots
Kadian *et al.* [116] IRAL'20	PointNav	Habitat PyRobot	LoCoBot	RGB-D LIDAR ResNet	RL DD-PPO	Open	SRCC metric to quantify real predictivity
Truong *et al.* [117] IRAL'21	PointNav	Habitat PyRobot DA	LoCoBot	RGB-D CycleGAN	RL DD-PPO	Closed rollouts	Bi-directional domain adaptation

learned then tested on real robots. Discrepancy between simulated and real performances is used to improve the initial simulator to be grounded as reality. This method helps a real humanoid robot, Nao, to increase walking speed by 25% with only four iterations starting from hand-coded walk parameters. From the viewpoint of parallel learning, GSL can be regarded as off-line prescriptive learning with iterative virtual-real interaction. But a constraint of GSL is the requirement of a human expert in loop. Later work of Hanna and Stone [108] proposed grounded action transformation, an optimization method based on the covariance matrix adaption evolutionary strategy (CMA-ES) algorithm, to alter the simulator for better matching to real world. Results on real robot outperform the SOTA hand-coded method to improve walk velocity by 43%.

For training agile locomotion of quadrupedal robots, Tan *et al.* [109] proposed to narrow the virtual-real gap by improving the simulator through system identification (SI) to get an accurate actuator model with latency. Randomization and perturbation also played essential roles in learning a robust controller. After learning in simulation, a quadruped robot Minitaur can successfully trot and gallop in real world with higher speed and lower power cost compared to a handcrafted robot. Yu *et al.* [110] proposed a meta-learning algorithm, Meta strategy optimization. By utilizing latent variables with a few (75) of trials in real word, the learned policies can quickly adapt to new target environment. Evaluation on a real quadruped robot demonstrates successful adaptation to various scenarios and outperforms two baseline methods. Hwangbo *et al.* [111] introduced a method to learn policies with an in-house rigid-body simulator and transfer to a sophisticated medium-dog-sized quadrupedal robot, ANYmal, who achieved skills recovering from falling which goes beyond that learned with prior methods. Lee *et al.* [112] presented an impressive work for learning a robust quadrupedal locomotion controller for legged robots in challenging natural environment like mud and snow, or thick vegetation and gushing water over ground. In addition to domain randomization of physical parameters, their training strategy includes a teacher policy and a student policy through simulation. The learning of the teacher policy has access to privileged information, e.g., ground-truth of the terrain and the robot's contact with it. The learning of student policy is based on a temporal convolutional network such that it produces actuation based on an extended history of proprioceptive states. Both learning process involves an automated curriculum that synthesizes terrains adaptively according to the robot's performance. This approach demonstrates zero-shot generalization from simulation to a variety of natural environments.

For tasks of navigation, Hong *et al.* [118] proposed a modular architecture separating the learning model into a perception and a control policy module, using semantic segmentation as mutual interface. Zero-shot real world experiments showed that the proposed method outperformed several baselines with high success rate. In addition, using semantic segmentation brings advantages of better efficiency and less noise. Instead of tuning the simulator to match reality, or mapping the virtual and real into a common segmentation space, Zhang *et al.* [114] proposed to close the reality gap in an

(a) Adding cars in 3D scene [125] (b) Inserting cars in image [131] (c) Inserting pedestrian in image [140] (d) Detection and segmentation [137] (e) Synthetic LiDAR point clouds of snow fall [144]

Fig. 13. Selected examples of virtual-to-real for autonomous driving.

opposite direction by translating the real-world images backwards to the virtual domain. Both indoor and outdoor experiments, with real robot and vehicle, validated this method as a flexible and efficient solution for visual control. Mitriakov *et al.* [115] studied a reinforcement learning based staircase negotiation method through simulation and transferred to reality in a zero-shot setting. By comparing different robots and task variants via Kullback-Leibler (KL) policy divergence, the proposed method has high scalability and portability.

Aiming for a PointGoal navigation task, Kadian *et al.* [116] built a virtual environment by 3D-scanning a physical lab and run parallel tests with 9 different models. The authors proposed sim-vs-real correlation coefficient (SRCC) as a new metric to quantify sim-to-real predictivity. Experiments using LoCoBot showed that they can improve sim2real predictivity by tuning simulation parameters to increase SRCC. Furthermore, Truong *et al.* [117] proposed bi-directional domain adaptation (BDA) to bridge sim-real gap in both directions, i.e., real2sim to bridge the visual domain gap and sim2real to bridge the dynamics domain gap. A policy based on BDA took only 5 k real-world (state, action, next-state) samples and performed on par with a policy fine-tuned with 600 k samples, showing much improvement on data efficiency, using 117X less real data.

3) Other Tasks: Beyond robotic navigation and locomotion, there are many sim-to-real applications related to mobile robots like [119] and [120], to mention a few. Sadeghi and Levine [119] focused on controlling the collision-free indoor flight. They proposed CAD2RL, a method that transfers a control policy to real world after training entirely on 3D CAD models from which RGB images are rendered as input to CNN for visual processing. With randomized rendering settings in simulation during training, experiments by flying a real quadrotor across indoor environments showed the learned policy could generalize to real world. Du *et al.* [120] presented a method to control an underwater soft robot, Starfish, with a differentiable simulator coupled with an analytical hydrodynamic model. This method initializes the simulator with data from real robot, then alternates between simulation and real rollouts. Specifically, the simulation step uses gradients from a differentiable simulator to run system identification and trajectory optimization, and the experiment step executes the optimized trajectory on the robot to collect new data to be fed into simulation. Experiments with Starfish showed that using gradients from the differentiable simulator not only narrows down the reality gap but also improves the performance of the open-loop controller in real world. This is also a typical prescriptive learning method in the same manner as the method introduced in [121].

4) Summary: Reinforcement learning is the predominant approach in robotics, and simulation and domain randomization (DR) are the typical combination indispensable to assist the training of autonomous driving vehicles. Zero-shot transfer is the ideal goal to pursue, but considering the high cost and low safety of try-error, few-shot learning or approaches based on system identification (SI) are more popular with better performance.

D. Virtual-to-Real in Autonomous Driving

In recent years, autonomous driving (AD) and advanced driver assistance system (ADAS) have become hot research topics, where deep learning models play essential roles throughout the whole process including perception, planning and control. However, due to safety and cost concerns, it is difficult to collect enough labeled data covering all scenarios by driving real cars along roads across the city. To solve this problem, researchers turned to simulation for help and achieved great progress on perception, planning and control. Selected examples refer to Fig. 13.

1) Perception: Given the facts that both perception in AD (P-AD) and computer vision (CV) share some common tasks like detection, segmentation, estimation, etc., they differ from each other in three aspects. First, the objects to detect are different. CV tries to detect various objects we concern in our daily life, while P-AD focuses on traffic related objects like pedestrian, vehicle, signs, lanes, etc. Second, the backgrounds are different. CV usually works in a specific view, while P-AD has to detect objects with varying backgrounds as the vehicle driving along the road. Third, P-AD involves multi-modal sensors like LiDAR, Radar, and event camera, in addition to normal camera. In this part, we mainly focus on virtual-to-real methods (Table VIII) on detection, segmentation, and multi-tasks within one framework in AD.

Works using 3D game engine to generate synthetic traffic data can be traced back to early 2010s, when Marin *et al.* [145] presented an approach based on half-life 2. Synthetic datasets through cameras on a car driving around the virtual city. An HOG/linear-SVM model is trained on the virtual data then tested on the real Daimler dataset, showing improvement with evaluation metric of mean average precision (mAP). This is a straight forward pipeline from virtual to real without feedback. Later, Vazquez *et al.* [122] extended the work by introducing a domain adaptation module named V-AYLA, which achieves the same accuracy as training on many human-labeled data and testing with real-world images.

The virtual KITTI by Gaidon *et al.* [146] is one of the pioneering virtual datasets generated from a dynamic and realistic virtual world using the Unity3D game engine built through

TABLE VIII
COMPARISON OF SELECTED WORKS ON VIRTUAL-TO-REAL PERCEPTION FOR AUTONOMOUS DRIVING

Method	Task	Description			Prediction		Prescription	Highlights
		Dim	Virtual	Real	Model	Improve (%)	Loop	
Vazquez *et al.* [122] PAMI'14	Det	3D−2D	Half-Life2 DA V-AYLA	INRIA Daimler	HOG/LBP linear-SVM	MR 26.13 (2.14)	Open	Human data from 3D game with domain adaptation
Richter *et al.* [123] ECCV'16	Seg	3D−2D	GTA-V v+r	CamVid KITTI	Dilated Convolutions	mIoU 68.9 (3.6) 61.6 (2.4)	Open	Creating semantic label using game engine
Liu *et al.* [124] IV'19	Det	3D−2D	GTA-V 200K	KITTI Cityscapes rendering	YOLOv3	86.7 (5.5)	Open	Auto generation rainy snowy harsh weather
Ros *et al.* [125] CVPR'16	Seg	3D−2D	Unity3D SYNTHIA	Camvid KITTI	T-Net	accuracy 90.7 (8.8) 80.8 (0.3)	Open	SYNTHIA dataset for urban scenarios
Saleh *et al.* [126] ECCV'18	Seg	3D−2D	Unity3D VEIS v+r	CityScapes	DeepLab Mask R-CNN	mIoU 42.5 (9.2)	Guided by prior	Treat foreground and background differently
Tian *et al.* [127] IJAS'18	Det	3D−2D	Unity3D CityEngine mix V+R	KITTI VOC COCO	DPM Faster R-CNN	79.8 (1.3)	Open	Parallel vision automaticlly annotations
Li *et al.* [128] T-ITS'19	Det seg	3D−2D	Unity3D ParallelEye v+r	KITTI CityScapes	Faster R-CNN FCN	89.6 (2.7) 56.6 (2.3)	Open	Large-scale artificial scenes for virtual data
Abu *et al.* [129] IJCV'18	Det seg	2D−2D	Blender KITTI-360 rendering	CityScapes	MNC	51.3 (8) 49.7 (6.4)	Open guided by human	Augment real image with 3D models a few manual
Zhang *et al.* [130] ECCV'20	Loc det seg	2D−2D	PlaceNet inpainting	Cityscape KITTI	YOLOv3 Mask R-CNN	30.2 (3.7) 30.2 (1.6)	Guided learned prior	Image inpainting learn to place data augmentation
Chen *et al.* [131] CVPR'21	Seg	2D−2D	GeoSim Multi-Sensor 3D Reconstruction	UrbanData	PSPNet DeepLabv3	95.3 (1.8) 94.2 (0.2)	Open guided by geometry	Augment real images with 3D objects reconstruction
Zhang *et al.* [132] CVPR'22	Det track	2D−3D -2D	SIMBAR relighting	vKITTI	CenterTrack	93.3 (0.9)	Guided learned geometry	Geometry awared image-based scene relighting
Kar *et al.* [133] ICCV'19	Det	3D−2D	Unreal Meta-Sim translation	KITTI	Mask R-CNN	66.5 (2.8)	Closed feedback	Learns to modify scene graphs prob. scene grammar
Devaranjan *et al.* [134] ECCV'20	Det	3D−2D	Unreal Meta-Sim2 translation	KITTI	Mask R-CNN	67 (0.7) FID 99.7 (6.9)	Closed feedback	Learn scene structure in addition to parameters
Kishore *et al.* [135] ICCV'21	Det	3D−2D	Unreal mix V+R blur	KITTI	DNN ResNet Faster R-CNN	68.2 (0.3)	Closed feedback	Corner case imitation training cyclic optimization
Chen *et al.* [136] CVPR'18	Det	2D−2D	SIM 10K DA	Cityscapes	Faster R-CNN	27.6 (8.8)	Adversarial	Image & instance level alignment adversarial training
Zhang *et al.* [137] T-ITS'21	Det seg	2D−2D	SYNTHIA vKITTI feature align	CityScapes CamVid	Mask R-CNN	34.1 (3.8) 25.2 (2.7)	Guided learned prior	Global and local V2R interaction aligned discrepancy
Ouyang *et al.* [138] arXiv'18	Det seg	2D−2D	PS-GAN mix V+R	Cityscapes	Faster R-CNN	46.4 (2.3)	Adversarial	Insert pedestrian into real image minimize artefacts
Zheng *et al.* [139] ECCV'20	Loc seg det	2D−2D	ForkGAN	BBD100K	Deeplab-v3	14.4 (7.1)	Adversarial	Day & night decouple invariant and specific content
Vobecky *et al.* [140] AAAI'21	Det	2D−2D	DummyNet GAN	Citypersons Caltech NightOwls	Faster R-CNN CSP	MR 10.25 (0.74)	Closed feedback	Controlled person place in real image using GAN
Sallab *et al.* [141] NeurIPS'19	Det	3D−3D PC	CARLA CycleGAN NST	KITTI	Oriented YOLO	71.5 (5.6)	Closed feedback	LiDAR augmentation CycleGANs task oriented
Fang *et al.* [142] IRAL'20	Det seg	3D−3D PC	3D scanner PointNet++ CAD model	KITTI	SECOND MV3D	71.5 (5.7) 45.5 (4.6)	Closed learned prior	Real background CAD objects outperform CARLA
Lehner *et al.* [143] CVPR'22	Det	3D−3D PC	3D−VField CrashD deform	KITTI Waymo	PointPillar etc.	77.13 (0.02)	Adversarial	Out of domain damaged/rare cars adversarial loss
Hahner *et al.* [144] CVPR'22	Det	3D−3D PC	physically based simulation	STF	PointPillar etc.	29.79 (1.15)	Open guided by knowledge	Physically based snow fall ground wetness

a real-to-virtual cloning method. The dataset came with ground truth for training models on multiple tasks including detection, segmentation, depth and optical flow estimation. Similarly, SYNTHIA proposed by Ros *et al.* [125] is a synthetic dataset of diverse urban images with automatically generated annotations based on the Unity3D and verified on segmentation tasks. Considering the domain shift problem, Saleh *et al.* [126] proposed treating foreground and background separately in Unity3D, based on the observation that they are not affected in the same way. The core feature is keeping shapes of objects look natural but not devoting too much efforts to make their textures photorealistic. Tian *et al.* [127] proposed to train a deep neural detector, Faster R-CNN, with virtual images generated by a tool-kit including Unity3D, CityEngine and OpenStreetMap. Totally 15 931 images of car, bus and truck were generated with automatic annotations. Experiments showed that a mixed training setting with both real and virtual data outperforms training with pure real data of KITTI. Li *et al.* [147] proposed ParallelEye based on [127], an automatically generated virtual image dataset with precise annotations for multi-tasks. Experiments showed that, by combining ParallelEye with real-world datasets during training, the performance of models can be significantly improved.

Richter *et al.* [123] introduced a typical virtual-to-real approach, wherein 2D images were collected by driving through a virtual city in the high-realism video game GTA-V. Totally 25 K photorealistic images with pixel-level annotations were generated to train a dilated convolutional network for semantic segmentation. The authors reported that task networks trained on generated virtual data with 1/3 real CamVid data outperformed models trained completely on real data. Aiming to handle corner cases in object detection for AD, Liu *et al.* [124] introduce an automated pipeline to generate synthetic images under harsh weather conditions through GTA-V. A large dataset with 200 K virtual images helps the YOLO v3 outperform its counterparts training and testing on both rainy and snowy validation images from real dataset Cityscapes and KITTI.

All these aforementioned works are based on open pipeline that the settings to generate virtual images depend on human priors. On the contrary, Kishore *et al.* [135] proposed an imitation training pipeline to guide synthetic data generation. The pipeline has a circular loop that comprises three phases, i.e., finding failure cases using the existing system first, synthesizing data to imitate failure cases, and training the detector with new synthetic data. This is a typical prescriptive learning that iteratively updates the synthesizer through the closed-loop with feedback from detector, until the evaluation metric converges. Experiments on real dataset Waymo and KITTI show its advantages over an SOTA detector. Kar *et al.* [133] argues that domain gap results from not only appearance but also scene content, e.g., layout and types of objects. To address this issue, the authors introduced Meta-Sim, a framework aiming to generate virtual data with minimized domain gap both in appearance and scene contents based on the Unreal engine. Firstly, an initial 3D scene graph, sampled from probabilistic scene grammar, is setup with objects whose parameters (e.g., positions, pose, color) are learnable. Secondly, Meta-Sim presented a training strategy with two loops. The inner loop aims to minimize the maximum mean discrepancy (MMD) in InceptionV3 feature space between generated virtual images and real images, by backpropagating through finite difference. The outer loop takes in task performance on real test data by defining a loss function and optimizing it by a REINFORCE score estimator. From the viewpoint of parallel learning, this is a work with prescriptive iterations to optimize virtual data generation that aims to improve performance of task model in a closed loop. However, the learnable parameters in Meta-Sim are limited to object attributes like positions and poses, while the scene structure parameters, like number of vehicles, are fixed during training. To address this issue, Devaranjan *et al.* [134] proposed Meta-Sim2, which emphasized on learning the scene structures. The authors regarded the construction of scene graph as sequentially sampling from probabilistic scene grammar, and trained the proposed model using reinforcement learning with a feature space divergence between virtual and real data. Experiments with a task model of Mask R-CNN tested on KITTI dataset showed that the Meta-Sim framework can generate both quantitatively and qualitatively better samples verified by improvement of the task model.

Instead of using 3D game engines, a group of methods directly augment images in 2D space in adversarial manner. Abualhaija *et al.* [129] tried to use background from easily available real images while placing virtual objects, manually or automatically, into real background through Blender for training segmentation models. Zhang *et al.* [130] proposed PlaceNet to place foreground objects into images without human interactions, for detection and instance segmentation. Chen *et al.* [131] proposed GeoSim, a geometry-aware image processing approach that inserts dynamic objects into existing images at plausible locations with novel poses, by exploiting the 3D HD maps and LiDAR data. A segmentation model trained with 9879 examples obtained by GeoSim based on 2 K labeled data got performance gains of 3.4% for cars.

Chen *et al.* [136] proposed a framework to address domain gaps on both image level and instance level according to H-divergence theory. By learning a domain classifier and a augmented Faster R-CNN detector simultaneously in an adversarial manner, experiments of car detection on the Cityscapes validation set showed that the detector gets a significant 7.7% improvement with evaluation metric of average precision (AP). For instance segmentation task, Zhang *et al.* [137] proposed a virtual-real interaction method that uses discrepancy between the two domains. Similarly to [136], the authors designed two components to align discrepancies from both global-level and local-level, as well as a consistency alignment component to encourage the consistency between them.

Ouyang *et al.* [138] proposed pedestrian-synthesis-GAN (PS-GAN), a synthesizer based on GAN with multiple discriminators. Instead of building 3D virtual scenes, PS-GAN tries to insert pedestrians into the background of a real image with minimized artefacts. The authors reported a Faster R-CNN detector trained with mixed real and virtual data got best performance. Zheng *et al.* [139] presented ForkGAN to boost performance for multiple visual tasks including localization, object detection and semantic segmentation in autonomous driving, with night-to-day image translation. Vobecky *et al.* [140] proposed a controlled pedestrian augmentation frame-

TABLE IX
COMPARISON OF SELECTED VIRTUAL-TO-REAL WORKS ON CONTROL OF AUTONOMOUS DRIVING WITH REAL DATA

Method	Description		Prediction			Prescription	Highlights
	Virtual	Real	Sensor	Controller	Metrics	Loop	
Pan *et al.* [152] BMVC'17	TORCS	45 K real data	SegNet	RL A3C	Acc 15.07	Closed	Realistic image translation by aligned segmentation
Yang *et al.* [153] ECCV'18	TORCS CARLA	Comma.ai Udacity	RGB images	PilotNet joint training adversarial	MAE 11−41	Closed	Transform multiple real domains to single virtual domain
Yin *et al.* [154] ArXiv'21	Data-defined simulator A^* annotation	INTERACTION	Image XCeption	MLP	Success rate 20−40	Closed	Imitation learning real traffic data as weak simulator
Zhou *et al.* [155] IROS'21	Apollo Dreamland	Real data	Image MobileNetV2 attention	LSTM MLP	Pass rate 25	Closed differentiable feedback	Mid-to-mid feedback synthesizer data augmentation
Bansal *et al.* [156] RSS'19	Perturbation expert behavior	Real data 26 M samples 60 days	FeatureNet mid-level	IL AgentRNN	L2 distance	Closed	Imitation learning synthetic perturbation multi losses
Scheel *et al.* [157] CoRL'21	Differentiable simulator	Lyft motion prediction ford fusion	Mid-level	IL GNN	L2 distance off-road collisions	Closed	Imitation learning policy gradient differentiable sim
Amini *et al.* [158] RA-L'20	VISTA	Toyota Prius V	Camera IMU GPS	RL	Lane follow near-crash recovery	Closed	Imitation learning synthetic perturbation multi losses
Amini *et al.* [159] ICRA'22	VISTA2	Lexus RX 450H	RGB camera LiDAR event camera	RL	Deviation crash rate	Closed	Multi-modality policy learning corner cases
Osinski *et al.* [160] ICRA'20	CARLA	Full-size real ar	RGB v, a	RL PPO	Deviation	Closed	End-to-end RL contol parallel training
Wang *et al.* [161] IRCA'22	Multi-agent data-driven simulation	Lexus RX 450H	RGB CNN	RL LSTM PPO	Intervention deviation	Closed	Zero-shot policy transfer interaction

work for automotive scenario, called DummyNet, which is also a GAN architecture to improve data diversity. The DummyNet takes appearance, pose in key point and target background as input, and outputs an image with composited people smoothly aligned with background. Specifically, a pretrained variational autoencoder (VAE) takes charge of appearance and the OpenPose is used to extract key point of a given pose. The authors augmented CityPersons and Caltech dataset using DummyNet, and conducted experiments with the CSP detector. An experiment with Faster-RCNN detector trained on augmented CityPersons then tested on NightOwls dataset showed DummyNet's ability to generate various data covering day-to-night changes. Other works including [148]–[150] also utilized adversarial training with generative network to synthesize realistic data for pedestrian detection. They all took effects on closing the domain gap between real and virtual data.

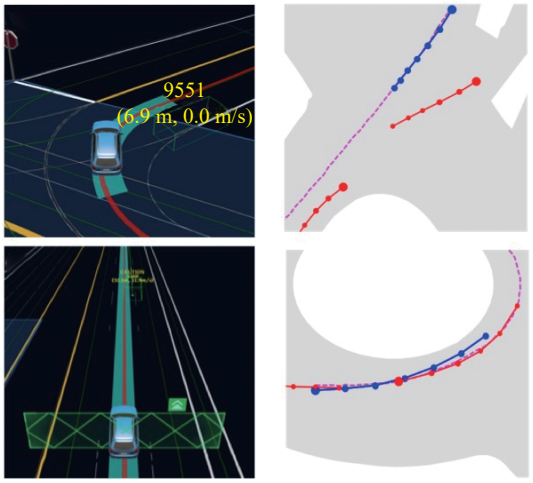
In addition to images, point clouds from LiDAR are another type of data widely used in AD. However, point clouds are very hard to annotate. To address this problem, Sallab *et al.* [141] presented their work learning on virtual LiDAR point clouds with annotation generated from CARLA. The framework takes CycleGANs and style transfer techniques to adapt domain from virtual to the real KITTI dataset. The virtual data generation is guided by evaluation of detection task, which forms a prescriptive loop. Fang *et al.* [142] proposed a LiDAR simulator that augments real point clouds with synthetic obstacles (e.g., vehicles, pedestrians, and other movable objects). By placing the synthetic obstacles properly into the background, detectors trained purely on simulated LiDAR point clouds outperform those trained with real data. Manivasagam *et al.* [151] presented LiDARsim, a method to generate LiDAR point clouds. LiDARsim first built a virtual world by configuring static maps and dynamic objects, then utilized ray-casting over the 3D scene to produce realistic LiDAR point clouds. The generated data were used to train models in case of real data with long-tail problem. To generate corner cases under harsh weather, Hahner *et al.* [144] proposed to generate the effect of snowfall on real clearweather LiDAR point clouds based on physically simulation. Lehner *et al.* [143] tried to generalize detectors to detect out-of-domain 3D objects 3D-VField, an augmentation method that plausibly deforms objects via vector fields learned in an adversarial fashion without adding new objects.

2) Planning and Control: Autonomous vehicles need to plan the route based on perceptions and then control the steering system to drive smoothly and safely on the road. Learning a robust control policy in dynamic environment is much more challenging than training perception models based on datasets. Considering safety issues, some works tried to train driving policy based on datasets collected from real traffic scenarios. However, collecting real data on road is not only expensive but also time consuming. Training with virtual environment has shown advantages. Related works on planning and control with real data, or real vars, are listed in Table IX.

Pan *et al.* [152] proposed to train driving policy in virtual environment with realistic frames then apply it to the real world. The key is a two-phase translation network that firstly converts non-realistic virtual image into a middle representation (e.g., segmentation) then converts into realistic images while keeping the scene structure constant. Policies learned by

reinforcement learning using translated images outperform the policies trained with original virtual images. Instead of converting virtual image to realistic ones, Yang *et al.* [153] presented DU-drive, an unsupervised framework for real-to-virtual domain unification to reduce domain gap. Through DU-drive, real images are converted into simplified virtual domain, where policies are trained to predict driving commands more easily. This method, avoiding intermediate representation and utilizing unlimited virtual images, showed superior performance on real driving datasets. Yuan *et al.* [162] proposed VRDriving, which is also an end-to-end framework based on adversarial learning. Similarly to DU-drive, an encoder is introduced to transform the real image into the virtual domain at feature map level. This encoder, the estimation model, and a discriminator are trained together with cost-sensitive loss function to get SOTA estimation accuracy.

The following three works are all based on imitation learning but differ in how data are used. Imitation learning is a data driven method that mimics expert's driving behavior and takes advantages of supervised learning. it can be trained directly on the collected real traffic data avoiding the simulator designing issue compared with reinforcement learning based methods. It can integrate perception and control into an end-to-end trainable model. However, IL has covariate shift problem that results in accumulated errors when applying imitation learning to real world. To address this problem, Bansal *et al.* [156] proposed ChauffeurNet to improve performance of behavior cloning based on synthetic data by perturbating demonstrated driving behavior from experts, other than rolling out policies through simulation. In addition, the imitation loss is augmented with losses that penalize undesirable actions like driving off-road or collisions, leading to robust policy. ChauffeurNet was tested under complex situations in simulation, and a real car at the test facility. Yin *et al.* [154] proposed to learn a behavioral cloning policy with interactions to real data in an iterative manner. Specifically, a weak simulator is created with surrounding vehicles following the trajectory provided by the real dataset, and an A^* planner is introduced to provide expert-like demonstration. This method alleviates both the difficulties to run real world rollouts and the tedious labeling work of human experts. While Zhou *et al.* [155] proposed a feedback synthesizer to address the distributional shift issue, generating and perturbing on-policy data in a mid-to-mid framework, illustrated in Fig. 14. The synthesizer works in a prescriptive manner that it is guided to produce unseen environments to improve overall performance. In addition to imitating, this method utilized losses that penalize undesirable behaviors by introducing a differentiable vehicle rasterizer that directly converts the waypoints output into images, avoiding heavy-weight ConvLSTM networks. The authors also exploited attention mechanism to reason critical objects and a post-processing planner to improve driving comfort. By the metric of pass rate, the proposed method achieved a 70% that outperforms pure IL's 15% with a large leap. Scheel *et al.* [157] presented a differentiable simulator, based on top of perception outputs and high-fidelity HD maps, so that an offline policy gradient method can be used in imitation learning of driving policy. Using mid-level representations, this method



(a) Zhou *et al.* [155] (b) Yin *et al.* [154]

Fig. 14. Selected examples of mid-mid control in autonomous driving using virtual data.

augmented existing demonstrations by synthesizing new driving experiences. Experiments showed that policies trained on generated data generalized well to real world cars without the need to collect additional on-policy data or introduce state perturbations during training.

There are works taking advantages of virtual LiDAR data, e.g., Sallab *et al.* [163] proposed to produce realistic LiDAR from simulation (sim2real) and generate high-resolution, realistic LiDAR from lower resolution one (real2real) by employing CycleGANs. Saputra *et al.* [164] proposed detecting casualty involving human body lying on ground using point cloud.

3) Decision and Control With Real Car on Road: While there are risks to test autonomous vehicles on real roads, researchers have been trying to take advantages of both simulation and real rollouts to learn better driving policies more efficiently.

For end-to-end reinforcement learning, Amini *et al.* [158] proposed to render virtual scenes based on human collected real trajectories. Through a simulator, VISTA, a single real trajectory was synthesized into a space of new possible trajectories. Virtual agents are trained by driving new trajectories along the road with consistent appearance and semantics. Experiments showed good generalization to unseen real-world roads with a full-size autonomous car, with better performance in metrics of lane following and near-crash recovery. Later in [159], the authors enhanced the VISTA to its 2.0 version as a multi-modality simulator, supporting sensors of RGB camera, LiDAR and Event camera.

Osinski *et al.* [160] presented reinforcement learning for driving policy of a full-size real car through simulation. The driving controller was trained with synthetic images and their semantic segmentation, but the segmentation network was trained with labeled real-world data.

Focusing on learning driving policies in challenging scenarios involving multi-agents interactions, Wang *et al.* [161] proposed using in-painted ado vehicles for learning robust driving policies in simulation. Experiments showed that the trained policies can be transferred to a full-scale autonomous vehicle directly in a Sero-Shot setting, with better perfor-

mance under test scenarios of car following and over-taking.

Full-size cars in real-world are not always available for machine learning due to cost and safety considerations, there are sim-to-real works utilizing small-scaled cars with simulations. For RL in shared space with potential collisions, Mitchell *et al.* [165] proposed a sim2real approach that uses real-world online policy adaptation. This mixed-reality setup, where a real car with other virtual vehicles and static obstacles, enables to learn by virtual collisions between the real car and other virtual objects. Stocco *et al.* [166] conducted an empirical study on sim-to-real adaptation with a small-scaled physical donkey car and its virtual counterpart in a simulation environment. The transferability of behavior and failure exposure between the virtual and real-world environments were investigated in vast set of experimental settings.

4) Summary: An autonomous driving (AD) system includes perception, planning and control. AD deeply intersects with both computer vision and robotics domains. To avoid the high cost and danger associated with learning on real city roads, people have to find safe and efficient methods. Imitation learning based on expert experience greatly reduces the danger, but suffers from covariate shift problems. Therefore, reinforcement learning relying on 3D dynamic simulation (4D) in a virtual environment has become the mainstream approach. The generation and fusion of multi-sensor virtual data including LiADR data is also an important challenge. Nevertheless, some work has been done to perform validation tests based on real vehicles on restricted real roads. Overall, V2R has made an indispensable contribution to the move towards L5 autonomous driving.

V. Analysis and Discussion

Virtual-to-real has become a popular paradigm across machine learning during last five years. We counted the number of papers whose title contains key words "augment", "synthetic" and "sim2real" from related top annual conferences including ICML, AAAI, CVPR, NeurIPS, ACL, EMNLP, ICASSP, IROS, and IRCA. As shown in Fig. 15, the number of virtual-to-real works of CV, NLP, Robotics and AD keeps increasing during the last 5 years. Although virtual-to-real has kept attracting attention, it is still under developing. In this part, we analyze the current issues and discuss the problems that remain to be studied in the future, from the viewport of parallel learning.

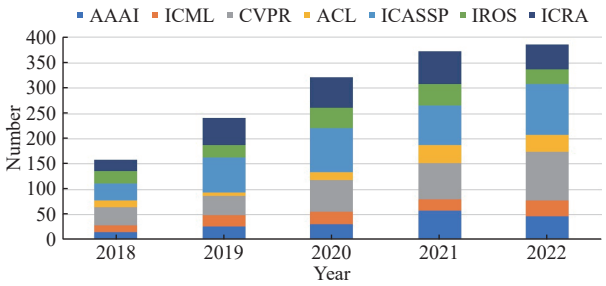


Fig. 15. The number of virtual-to-real works keeps increasing during the recent five years.

A. Current Situation

1) Discussion of Virtual-to-Real Evaluation Metric: In Section IV, we reviewed more than one hundred of virtual-to-real works in machine learning. Readers should have no doubts that the V2R paradigm helps to improve learning performance. But when we try to compare V2R methods in a specific domain, we are facing difficulties to make fair comparisons because of the lacking of benchmarks and common evaluation metrics. Taking CV for example, all V2R methods try to generate images that help to improve the performance learning models, but few of them share common settings including tasks, learning models, usage of validation and testing data, baselines, and evaluation metrics. The situations are even worse for robotics and AD.

To address this issue, we propose to evaluate a virtual-to-real method from three levels, as shown in Table X. The first level includes metrics to measure distribution similarity between the virtual and real dataset, known as the frechet inception distance (FID) and the kernel inception distance (KID). These metrics are general and task-agnostic. The second level includes metrics for evaluation of learning performance. These metrics are widely adopted in machine learning and depend on specific learning tasks, e.g., mAP in detection and mIoU in segmentation. The third level aims for evaluation of efficiency, by which we want to know the cost in training time and dataset size to obtain performance improvements.

TABLE X
Evaluation Metrics

Levels		Metrics
Data distribution		FID, KID
Learning performance	CV	(m)AP, mIoU, MR, MPPI, etc.
	NLP	Accuracy, CER, F1, etc.
	Robot	Error, distance, etc.
	AD	Distance, deviation, pass rate collision, intervention, recovery
Data quality & efficiency		Ratio of data size (or training time) to performance improvement

2) Discussion on Description of Real World: Description in parallel learning means creating the virtual opponent of the real world. Description covers a variety of techniques including text/voice generation, image composition, 3D reconstruction and dynamic simulation, etc. We try to summarize some common questions and analyze the possible reasons.

a) 2D−2D v.s. 3D−2D: A large percent of machine learning tasks take 2D data (images or frames in a video) as input, including all vision tasks and most robotic and autonomous driving tasks. There are two typical methods to generate virtual images in practical virtual-to-real applications, i.e., image composition based on real images and image projection from 3D virtual environments. What are the pros and cons of them? Which one prefers the other? We summarize their pros and cons in Table XI and discuss as follows.

3D−2D methods construct a virtual world firstly then generate images by projecting through a virtual camera. 3D−2D format mainly has two advantages. First, it is flexible for the

TABLE XI
COMPARISON OF DIMENSIONAL DATA GENERATION

Method	Pros	Cons	Examples
3D−2D	Flexible scene configurable zero-shot	High cost low realism introducing noisy	Meta-Sim [133], ParallelEye [147]
2D−2D	Easy processing high realism minimal shift	Less flexible difficult placement real data	DADA [26], ForkGAN [139], DummyNet [140]
1D−2D	High quality	Not mature for V2R	DALL-E2 [20], Imagen

TABLE XII
COMPARISON OF PRESCRIPTION METHODS

Method	Pros	Cons	Examples
Open-loop	Easy to use low cost zero-shot	Low quality bias artefacts harmful	[21]−[23], [35], [65]−[67], [85], [86], [88], [95], [122], [124], [127], [145]
Adversarial	Highly realistic GANs	Extra training real data	[28], [37], [38], [90], [91], [136], [143]
Guided by learned knowledge	High quality	Task specific extra learning	[34], [36]
Guided by expert knowledge	Priors experience	Task specific bias	[32], [33], [89], [92]
Feed-back	High quality optimized	Task specific real data roll-outs	[24]−[26], [61], [68], [87], [91], [94], [98], [133]−[135], [140]

users to setup varieties of 3D scenes and control the rendering parameters like camera pose, texture of objects and backgrounds, intensity of illumination, etc. Second, it is a choice when real data is not available, i.e., zero-shot scenarios where 3D scenes can be reconstructed based on prior knowledge. However, the 3D−2D format is a double-blade sword. First, though there is out-of-shelf software to aid the 3D construction process, it is still labor intensive and time consuming to setup 3D scenes and configure proper rendering parameters. Second, sometimes the human priors may introduce noise (bias) that is discrepant from the distribution of real data.

2D−2D methods generate virtual images based on real images, through augmentation, composition, stylization, rendering, etc. This kind of methods avoid tedious works of constructing and configuring 3D environments, so that the whole learning process can be done end-to-end. In addition, 2D−2D methods only modify objects of interesting in the image while keeping the background unchanged, which maintains the data distribution similar to the original dataset with minimal shift. On the other hand, 2D−2D methods have less flexibility because the view port is fixed when taking the picture in the real world. Furthermore, the prelimited condition is that there must be plenty of unlabeled real data.

Historically, 3D−2D methods were proposed earlier than 2D−2D methods, but the latter has dominated in recent years because of higher efficiency and plenty of image processing techniques. Nevertheless, 3D−2D methods have much space to be improved, especially after the introduction of learnable 3D scene construction and differentiable rendering engine.

b) 1D−2D generation: Recently, the large-scale generative models have attracted attention in the area of image generation. For example, DALL-E2 [20] has shown promise to generate/edit realistic images from a description in natural language, which makes 1D−2D data generation feasible. Methods like diffusion model behind the impressive images have potential power to help in virtual-to-real applications.

c) 4D dynamic simulation: For robotics and autonomous driving, dynamic simulator is essential to the success of virtual-to-real. To this end, the differentiable physics engines have been integrated into the pipeline to realize the end-to-end learning process. However, there is a long way to go before the physics engines simulate the natural world perfectly, especially rare phenomenon.

d) Photo realistic v.s. coarse realistic: Ideally, we want the virtual and real data to be twins of each other. Some works laid a lot of efforts to generate photo realistic images or data with same distribution as real data, at the cost of laborious tuning and sophisticated techniques. With the advancement of computer graphics, we have better tools to generate much more realistic data with less efforts. However, the domain gap between virtual and real datasets remains as an open problem.

While some people try to approach the real world with efforts, others argue that making the virtual data photo realistic is unnecessary. Instead, experiments had shown that deep models trained on randomly composed non-realistic images have better generalization compared with models trained on real dataset. Specifically, in robotics and AD planning and control, mid-level representations are more popular formats than high-realistic images. Currently, there is no common agreement on how realistic the virtual should be to get better performance for downstream tasks.

3) Discussion on Prescription for Better Virtual World: The concept of prescription in parallel learning suggests to update the virtual world to generate better dataset to boost learning performance. Instead of simple data augmentation or domain randomization, prescription tries to improve both data generator and task model (predictor) simultaneously. According to the way to improve the data generator, we categorize the prescription methods into four types, including open-loop, adversarial, guided (by learnt or expert knowledge), and feed-back, with their pros and cons summarized in Table XII.

a) Open-loop: The open-loop pipeline, i.e., training on virtual and then testing on real data, is popular in most domains. The word "open-loop" indicates the generation of virtual data is task agnostic, whether by synthesizing or simulation. This means human priors or domain specific knowledge are necessary to ensure the generated virtual data are consistent with the distribution of real data. Random domain augmentation

TABLE XIII
DEVELOPING TRENDS

Points	Trends
Description	1D−2D/3D generation based on diffusion model corner case generation and scenario engineering
Prescription	Knowledge guided optimization parallel driven by both knowledge and data
Pipeline	Differentiable rendering/physics engines hybrid guided/feed-back/adversarial methods

and domain randomization usually introduce noise into visual data that harms the performance of predictor. A large number of such kind of virtual data result in low efficiency because a big ratio of dataset would not contribute to the optimization of learning process. Nevertheless, open-loop pipeline is preferred in some applications due to easy implementation. More importantly, it is the only choice in the case of zero-shot scenario where real data or real environments are not accessible.

b) Adversarial: Naive open-loop pipeline cannot generate high quality efficiently. People have tried to address this problem and found that it can achieve better performance by constraining the virtual data generator. Adversarial training and GANs are popular frameworks in which one or multiple discriminators are used to reject data out of distribution of real data. This type requires that there are at least a small number of real data to train the discriminators.

c) Guided: Most successful machine learning models are deep neural networks driven by labeled big data. While in case of data scarcity, we use knowledge to compensate the data shortage. When we have experts, whose experience can be utilized to generate data for learning, we denote the prescription as guided by expert's knowledge. When there are no experts in a specific domain, we manage to learn the knowledge from a small number of unlabeled real data. With the learnt knowledge, we generate a large number of labeled data to train a better model.

d) Feed-back: In recent years, researchers have kept pursuing better data generation methods. One of promising methods is regarding the virtual data generation as an optimization problem by introducing feedback from the downstream task. However, due to the complexity of tasks, forming a closed loop that iteratively converges to an optimized solution is more difficult than open-loop methods. Searching methods are well known to solve optimal problem but suffer from low efficiency. For some tasks, reinforcement learning is a better choice by defining the feedback as rewords. To further improve the efficiency, people have tried to make the whole pipeline differentiable so that gradient descent methods can be used with massive processors. Thanks to the fast development of differentiable rendering engines and differentiable physics engines, the closed-loop pipeline can be updated automatically in which the data generator and predictor are trained simultaneously. Closed-loop pipeline iteratively minimizes the domain gap in an implicit manner, and effectively boosts the performance of learning models.

B. Problems and Trends

The virtual-to-real paradigm has become an essential way to boost learning performance in data scarcity scenarios. However, it is still under developing. Many people in the machine learning community keep trying to find better methods to solve the problems both in theory and in practice.

1) Problems: We emphasize on two typical problems.

a) Domain gap: The first problem is the domain gap (reality gap) between the real and virtual data, which is well known as an open problem. Though a variety of domain adaptation methods have been proposed to minimize the gap for better transfer from virtual to real world, it is still a hot research topic that is far from solved in recent years.

b) Benchmark: The second problem is the requirement of benchmarks. In addition to the performance of learning task, the quality of virtual data is also an essential factor for virtual-to-real applications. However, the lacking of benchmarks makes it difficult to fairly compare the existing methods and comprehensively evaluate new methods. We have proposed a three-level evaluation metric, and we also suggest to setup benchmarks in each machine learning subdomain accordingly.

2) Trends: Based on the review and analysis, we conclude that the following trends listed in Table XIII deserve more attention in the following years. The first promising direction is the efforts to use diffusion models to make the virtual world closer to the real world. The diffusion model has shown its power in text-to-image, text-to-video, and text-to-mesh, yet not used in virtual-to-real. The second promising direction is the introduction of differentiable rendering and physics engines that enable end-to-end pipeline for simultaneously learning data generator and machine learning models. The third direction is to use knowledge, learned from data or from human experts, to guide the virtual data generation. The parallel learning framework offers a good interface to fuse knowledge into the pipeline of machine learning, which is regarded as the feature of the 3rd generation of artificial intelligence.

VI. CONCLUSION

High quality data are essential to the performance of machine learning algorithms. However, the requirement of qualified data is not always satisfied in practice. To address this problem, the virtual-to-real becomes a promising paradigm that tries to generate virtual training data whenever real data are unavailable. As a fast-developing research area, it is essential to give an overview of the advances in recent years and clarify the problems that remain to be solved, as well opportunities in the future.

In this paper, we firstly extended the parallel learning. In the extended framework of parallel learning, we presented an overview of virtual-to-real methods mainly covering four domains including computer vision, natural language processing, robotic control, and autonomous driving. A taxonomy was proposed to organize the related literature systematically based on parallel learning. Specifically, for each of included method, we gave a brief analysis according to the three principles, i.e., description, prediction and prescription. Features of similar methods were listed as adjacent tuples in one table for comparison. We also summarized common features across

domains with emphasis on prescription which iteratively improves the generation of virtual data in a task-oriented closed loop.

For future development of virtual-to-real, we foresee increases in three aspects. The first is about description. Large scale pre-trained models will take their role in virtual data generation, especially in CV and NLP. Knowledges entailed in the pre-trained models will improve the quality of generated data without human interventions. The second is about prediction. Massively distributed training and testing based on virtual environments are efficient ways to accelerate the learning process, especially in the cases of learning policy for robots and autonomous vehicles. The last is on prescription. Generating virtual data in a task-oriented closed loop pipeline has shown advantages over open-loop formats. With the development of differentiable rendering engine and physics engines, end-to-end prescriptive learning is promising in more applications. In addition, the parallel learning framework offers an interface to realize machine learning driven by both data and knowledge.

REFERENCES

[1] S. I. Nikolenko, *Synthetic Data for Deep Learning*. Cham, Germany: Springer, 2021.

[2] C. Shorten and T. M. Khoshgoftaar, " A survey on image data augmentation for deep learning," *J. Big Data* , vol. 6, no. 1, p. 60, Jul. 2019.

[3] A. Tsirikoglou, G. Eilertsen, and J. Unger, " A survey of image synthesis methods for visual machine learning," *Comput. Graphics Forum*, vol. 39, no. 6, pp. 426–451, Sept. 2020.

[4] W. S. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: A survey," in *Proc. IEEE Symp. Series on Computational Intelligence*, Canberra, Australia, 2020, pp. 737-744.

[5] F. Muratore, F. Ramos, G. Turk, W. H. Yu, M. Gienger, and J. Peters, "Robot learning from randomized simulations: A review," *Front. Robot. AI*, vol. 9, p. 799893, Apr. 2021.

[6] F.-Y. Wang, " Artificial societies, computational experiments, and parallel systems: A discussion on computational theory of complex social-economic systems," *Complex Syst. Complexity Sci.*, vol. 1, no. 4, pp. 25–35, Oct. 2004.

[7] F.-Y. Wang, "Parallel system methods for management and control of complex systems," *Control Decis.*, vol. 19, no. 5, pp. 485–489, May 2004.

[8] F.-Y. Wang, "Computational theory and method on complex system," *China Basic Sci.*, vol. 6, no. 5, pp. 3–10, May 2004.

[9] F.-Y. Wang, X. Wang, L. X. Li, and L. Li, " Steps toward parallel intelligence," *IEEE/CAA J. Autom. Sinica* , vol. 3, no. 4, pp. 345–348, Oct. 2016.

[10] F.-Y. Wang, "Toward a paradigm shift in social computing: The ACP approach," *IEEE Intell. Syst.* , vol. 22, no. 5, pp. 65–67, Sept.−Oct. 2007.

[11] Y. S. Lv, Y. Y. Chen, L. Li, and F.-Y. Wang, "Generative adversarial networks for parallel transportation systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 3, pp. 4–10, Jun. 2018.

[12] Y. S. Lv, Y. Y. Chen, J. C. Jin, Z. J. Li, P. J. Ye, and F. H. Zhu, "Parallel transportation: Virtual-real interaction for intelligent traffic management and control," *Chin. J. Intell. Sci. Technol.*, vol. 1, no. 1, pp. 21–33, Mar. 2019.

[13] L. Li, X. Wang, K. F. Wang, Y. L. Lin, J. M. Xin, L. Chen, L. H. Xu, B. Tian, Y. F. Ai, J. Wang, D. P. Cao, Y. H. Liu, C. H. Wang, N. N. Zheng, and F.-Y. Wang, " Parallel testing of vehicle intelligence via virtual-real interaction," *Sci. Robot.*, vol. 4, no. 28, p. eaaw4106, Mar. 2019.

[14] Y. Y. Chen, Y. S. Lv, and F.-Y. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1624–1630, Apr. 2020.

[15] L. Li, Y. L. Lin, N. N. Zheng, and F.-Y. Wang, " Parallel learning: A perspective and a framework," *IEEE/CAA J. Autom. Sinica* , vol. 4, no. 3, pp. 389–395, Jan. 2017.

[16] M. Grieves, " PLM—Beyond lean manufacturing," *Manuf. Eng.*, vol. 130, no. 3, p. 23, Mar. 2003.

[17] M. Shafto, M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne, and L. Wang, " Modeling, simulation, information technology and processing roadmap," in *Proc. Nat. Aeronautics and Space Administration*, 2010.

[18] F. Piltan and J. M. Kim, " Bearing anomaly recognition using an intelligent digital twin integrated with machine learning," *Appl. Sci.*, vol. 11, no. 10, p. 4602, May 2021.

[19] Y. Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in digital twin networks," *IEEE Trans. Industr. Inform.*, vol. 17, no. 7, pp. 4968–4977, Jul. 2021.

[20] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with CLIP latents," arXiv preprint arXiv: 2204.06125, 2022.

[21] T. DeVries and G. W. Taylor, " Improved regularization of convolutional neural networks with cutout," arXiv preprint arXiv: 1708.04552, 2017.

[22] Z. Zhong, L. Zheng, G. L. Kang, S. Z. Li, and Y. Yang, "Random erasing data augmentation," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 13001–13008, Apr. 2020.

[23] J. L. Han, P. F. Fang, W. H. Li, J. Hong, M. A. Armin, I. Reid, L. Petersson, and H. D. Li, " You only cut once: Boosting data augmentation with a single cut," in *Proc. 39th Int. Conf. Machine Learning*, Baltimore, USA, 2022, pp. 8196–8212.

[24] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, USA, 2019, pp. 113–123.

[25] D. Ho, E. Liang, I. Stoica, P. Abbeel, and X. Chen, "Population based augmentation: Efficient learning of augmentation policy schedules," in *Proc. 36th Int. Conf. Machine Learning*, Long Beach, USA, 2019, pp. 2731–2741.

[26] Y. G. Li, G. S. Hu, Y. T. Wang, T. Hospedales, N. M. Robertson, and Y. X. Yang, " Differentiable automatic data augmentation," in *Proc. 16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 580–595.

[27] A. Fawzi, H. Samulowitz, D. Turaga, and P. Frossard, "Adaptive data augmentation for image classification," in *Proc. IEEE Int. Conf. Image Processing*, Phoenix, USA, 2016, pp. 3688–3692.

[28] S. Tripathi, S. Chandra, A. Agrawal, A. Tyagi, J. M. Rehg, and V. Chari, "Learning to generate synthetic data via compositing," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, USA, 2019, pp. 461–470.

[29] T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid, "A Bayesian data augmentation approach for learning deep models," in *Proc. 31st Int. Conf. Neural Information Processing Systems*, Long Beach, USA, 2017, pp. 2794–2803.

[30] Y. G. Yan, M. K. Tan, Y. W. Xu, J. Z. Cao, M. Ng, H. Q. Min, and Q. Y. Wu, " Oversampling for imbalanced data via optimal transport," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 5605–5612, Jul. 2019.

[31] Y. He, F. D. Lin, X. Yuan, and N. F. Tzeng, "Interpretable minority synthesis for imbalanced classification," in *Proc. 30th Int. Joint Conf. Artificial Intelligence*, Montreal, Canada, 2021, pp. 2542–2548.

[32] E. Cheung, T. K. Wong, A. Bera, X. G. Wang, and D. Manocha, "LCrowdV: Generating labeled videos for simulation-based crowd behavior learning," in *Proc. European Conf. Computer Vision*, Amsterdam, the Netherlands, 2016, pp. 709–727.

[33] W. W. Zhang, K. F. Wang, Y. T. Liu, Y. Lu, and F.-Y. Wang, "A parallel vision approach to scene-specific pedestrian detection," *Neurocomputing*, vol. 394, pp. 114–126, Jun. 2020.

[34] H. Hattori, N. Lee, V. N. Boddeti, F. Beainy, K. M. Kitani, and T. Kanade, " Synthesizing a scene-specific pedestrian detector and pose estimator for static video surveillance," *Int. J. Comput. Vis.* , vol. 126, no. 9, pp. 1027–1044, Sept. 2018.

[35] D. Dwibedi, I. Misra, and M. Hebert, " Cut, paste and learn:

Surprisingly easy synthesis for instance detection," in *Proc. IEEE Int. Conf. Computer Vision*, Venice, Italy, 2017, pp. 1310–1319.

[36] N. Dvornik, J. Mairal, and C. Schmid, "Modeling visual context is key to augmenting object detection datasets," in *Proc. 15th European Conf. Computer Vision*, Munich, Germany, 2018, pp. 375–391.

[37] S. Wu, S. H. Lin, W. H. Wu, M. Azzam, and H. S. Wong, "Semi-supervised pedestrian instance synthesis and detection with mutual reinforcement," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Seoul, Korea (South), 2019, pp. 5056–5065.

[38] L. L. Liu, M. Muelly, J. Deng, T. Pfister, and L. J. Li, "Generative modeling for small-data object detection," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Seoul, Korea, 2019, pp. 6072–6080.

[39] E. Martinson, B. Furlong, and A. Gillies, " Training rare object detection in satellite imagery with synthetic GAN images," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops*, Nashville, USA, 2021, pp. 2763–2770.

[40] P. L. Huang, J. W. Han, D. Cheng, and D. W. Zhang, "Robust region feature synthesizer for zero-shot object detection," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, New Orleans, USA, 2022, pp. 7612–7621.

[41] X. C. Peng, B. C. Sun, K. Ali, and K. Saenko, "Learning deep object detectors from 3D models," in *Proc. IEEE Int. Conf. Computer Vision*, Santiago, Chile, 2015, pp. 1278–1286.

[42] H. Hattori, V. N. Boddeti, K. Kitani, and T. Kanade, "Learning scene-specific pedestrian detectors without real data," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Boston, USA, 2015, pp. 3819–3827.

[43] K. Wang, B. Babenko, and S. Belongie, " End-to-end scene text recognition," in *Proc. Int. Conf. Computer Vision*, Barcelona, Spain, 2011, pp. 1457–1464.

[44] A. Gupta, A. Vedaldi, and A. Zisserman, " Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, USA, 2016, pp. 2315–2324.

[45] F. N. Zhan, S. J. Lu, and C. H. Xue, "Verisimilar image synthesis for accurate detection and recognition of texts in scenes," in *Proc. 15th European Conf. Computer Vision*, Munich, Germany, 2018, pp. 257–273.

[46] S. B. Long and C. Yao, "UnrealText: Synthesizing realistic scene text images from the unreal world," arXiv preprint arXiv: 2003.10608, 2020.

[47] A. Yu and K. Grauman, "Semantic jitter: Dense supervision for visual comparisons via synthetic images," in *Proc. IEEE Int. Conf. Computer Vision*, Venice, Italy, 2017, pp. 5571–5580.

[48] G. S. Hu, X. J. Peng, Y. X. Yang, T. M. Hospedales, and J. Verbeek, "Frankenstein: Learning deep face representations using small data," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 293–303, Jan. 2018.

[49] Y. C. Shi, X. Yu, K. Sohn, M. Chandraker, and A. K. Jain, "Towards universal representation learning for deep face recognition," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* , Seattle, USA, 2020, pp. 6816–6825.

[50] H. B. Qiu, B. S. Yu, D. H. Gong, Z. F. Li, W. Liu, and D. C. Tao, "SynFace: Face recognition with synthetic data," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Montreal, Canada, 2021, pp. 10860–10870.

[51] Z. H. Zhai, P. J. Yang, X. F. Zhang, M. J. Huang, H. J. Cheng, X. J. Yan, C. M. Wang, and S. L. Pu, "Demodalizing face recognition with synthetic samples," *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 4, pp. 3278–3286, May 2021.

[52] G. Rogez and C. Schmid, "MoCap-guided data augmentation for 3D pose estimation in the wild," in *Proc. 30th Int. Conf. Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 3116–3124.

[53] W. Z. Chen, H. Wang, Y. Y. Li, H. Su, Z. H. Wang, C. H. Tu, D. Lischinski, D. Cohen-Or, and B. Q. Chen, " Synthesizing training images for boosting human 3D pose estimation," in *Proc. 4th Int. Conf. 3D Vision*, Stanford, USA, 2016, pp. 479–488.

[54] D. Mehta, O. Sotnychenko, F. Mueller, W. P. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-shot multi-person 3D pose estimation from monocular RGB," in *Proc. Int. Conf. 3D Vision*, Verona, Italy, 2018, pp. 120–130.

[55] D. T. Hoffmann, D. Tzionas, M. J. Black, and S. Y. Tang, "Learning to train with synthetic humans," in *Proc. 41st German Conf. Pattern Recognition*, Dortmund, Germany, 2019, pp. 609–623.

[56] S. C. Li, L. Ke, K. Pratama, Y. W. Tai, C. K. Tang, and K. T. Cheng, "Cascaded deep monocular 3D human pose estimation with evolutionary training data," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Seattle, USA, 2020, pp. 6172–6182.

[57] K. H. Gong, J. F. Zhang, and J. S. Feng, "PoseAug: A differentiable pose augmentation framework for 3D human pose estimation," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Nashville, USA, 2021, pp. 8571–8580.

[58] J. A. Chen, D. Tam, C. Raffel, M. Bansal, and D. Y. Yang, "An empirical survey of data augmentation for limited data learning in NLP," arXiv preprint arXiv: 2106.07499, 2021.

[59] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A survey of data augmentation approaches for NLP," in *Proc. Findings of the Association for Computational Linguistics*, 2021, pp. 968–988.

[60] B. H. Li, Y. T. Hou, and W. X. Che, "Data augmentation approaches in natural language processing: A survey," *AI Open*, vol. 3, pp. 71–90, Nov. 2022.

[61] A. Tjandra, S. Sakti, and S. Nakamura, " Listening while speaking: Speech chain by deep learning," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, Okinawa, Japan, 2017, pp. 301–308.

[62] P. B. Denes and E. N. Pinson, *The Speech Chain: The Physics and Biology of Spoken Language*. Waveland Press, Inc., 2nd edition, Jul. 2015.

[63] A. Tjandra, S. Sakti, and S. Nakamura, "End-to-end feedback loss in speech chain framework via straight-through estimator," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Brighton, UK, 2019, pp. 6281–6285.

[64] A. Tjandra, S. Sakti, and S. Nakamura, " Transformer VQ-VAE for unsupervised unit discovery and speech synthesis: ZeroSpeech 2020 Challenge," in *Proc. Interspeech*, 2020, pp. 4851–4855.

[65] T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, and K. Takeda, "Back-translation-style data augmentation for end-to-end ASR," in *Proc. IEEE Spoken Language Technology Workshop*, Athens, Greece, 2018, pp. 426–433.

[66] N. Rossenbach, A. Zeyer, R. Schlüter, and H. Ney, "Generating synthetic audio data for attention-based speech recognition systems," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Barcelona, Spain, 2020, pp. 7069–7073.

[67] X. R. Zheng, Y. L. Liu, D. Gunceler, and D. Willett, "Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Toronto, Canada, 2021, pp. 5674–5678.

[68] T. Y. Hu, M. Armandpour, A. Shrivastava, J. H. R. Chang, H. Koppula, and O. Tuzel, "SYNT++: Utilizing imperfect synthetic data to improve speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Singapore, Singapore, 2022, pp. 7682–7686.

[69] T. Hori, R. Astudillo, T. Hayashi, Y. Zhang, S. Watanabe, and J. Le Roux, "Cycle-consistency training for end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Brighton, UK, 2019, pp. 6271–6275.

[70] G. Wang, A. Rosenberg, Z. H. Chen, Y. Zhang, B. Ramabhadran, Y. H. Wu, and P. Moreno, " Improving speech recognition using consistent predictions on synthesized speech," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Barcelona, Spain, 2020, pp. 7029–7033.

[71] Z. H. Chen, A. Rosenberg, Y. Zhang, G. Wang, B. Ramabhadran, and P. J. Moreno, "Improving speech recognition using GAN-based speech synthesis and contrastive unspoken text selection," in *Proc. Interspeech*, 2020, pp. 556–560.

[72] C. P. Du and K. Yu, "Speaker augmentation for low resource speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Barcelona, Spain, 2020, pp. 7719–7723.

[73] A. Fazel, W. Yang, Y. L. Liu, R. Barra-Chicote, Y. X. Meng, R. Maas, and J. Droppo, " SynthASR: Unlocking synthetic data for speech recognition," in *Proc. Interspeech*, 2021, pp. 896–900.

[74] Y. Cheng, L. Jiang, and W. Macherey, " Robust neural machine translation with doubly adversarial inputs," in *Proc. 57th Annu.*

Meeting of the Association for Computational Linguistics, Florence, Italy, 2019, pp. 4324–4333.

[75] Y. Cheng, L. Jiang, W. Macherey, and J. Eisenstein, "AdvAug: Robust adversarial augmentation for neural machine translation," in *Proc. 58th Annu. Meeting of the Association for Computational Linguistics*, 2020, pp. 5961–5970.

[76] D. He, Y. C. Xia, T. Qin, L. W. Wang, N. H. Yu, T. Y. Liu, and W. Y. Ma, "Dual learning for machine translation," in *Proc. 30th Int. Conf. Neural Information Processing Systems*, Barcelona, Spain, 2016, pp. 820–828.

[77] Y. C. Xia, T. Qin, W. Chen, J. Bian, N. H. Yu, and T. Y. Liu, "Dual supervised learning," in *Proc. 34th Int. Conf. Machine Learning*, Sydney, Australia, 2017, pp. 3789–3798.

[78] Y. J. Wang, Y. C. Xia, L. Zhao, J. Bian, T. Qin, G. Q. Liu, and T. Y. Liu, " Dual transfer learning for neural machine translation with marginal distribution regularization," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018.

[79] Z. R. Zhang, S. J. Liu, M. Li, M. Zhou, and E. H. Chen, "Joint training for neural machine translation models with monolingual data," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.

[80] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," in *Proc. 6th Int. Conf. Learning Representations*, Vancouver, Canada, 2018, pp. 14.

[81] X. Niu, M. Denkowski, and M. Carpuat, " Bi-directional neural machine translation with synthetic parallel data," in *Proc. 2nd Workshop Neural Machine Translation and Generation*, Melbourne, Australia, 2018, pp. 84–91.

[82] Y. R. Wang, Y. C. Xia, T. Y. He, F. Tian, T. Qin, C. X. Zhai, and T. Y. Liu, "Multi-agent dual learning," in *Proc. 7th Int. Conf. Learning Representations*, New Orleans, USA, 2019.

[83] B. Ahmadnia and B. J. Dorr, "Augmenting neural machine translation through round-trip training approach," *Open Comput. Sci.*, vol. 9, no. 1, pp. 268–278, Oct. 2019.

[84] Z. X. Zheng, H. Zhou, S. J. Huang, L. Li, X. Y. Dai, and J. J. Chen, "Mirror-generative neural machine translation," in *Proc. 8th Int. Conf. Learning Representations*, Addis Ababa, Ethiopia, 2020, p. 16.

[85] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vancouver, Canada, 2017, pp. 23–30.

[86] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," in *Proc. Robotics: Science and Systems XIV*, Pittsburgh, USA, 2018.

[87] F. Golemo, A. A. Taïga, A. C. Courville, and P. Y. Oudeyer, "Sim-to-real transfer with neural-augmented robot simulation," in *Proc. 2nd Annu. Conf. Robot Learning*, Zürich, Switzerland, 2018, pp. 817–828.

[88] F. Sadeghi, A. Toshev, E. Jang, and S. Levine, "Sim2Real viewpoint invariant visual servoing by recurrent control," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, USA, 2018, pp. 4691–4699.

[89] M. Y. Yan, I. Frosio, S. Tyree, and J. Kautz, "Sim-to-real transfer of accurate grasping with eye-in-hand observations and continuous control," arXiv preprint arXiv: 1712.03303, 2017.

[90] K. Bousmalis, A. Irpan, P. Wohlhart, Y. F. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, " Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *Proc. IEEE Int. Conf. Robotics and Autom.*, Brisbane, Australia, 2018, pp. 4243–4250.

[91] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Long Beach, USA, 2019, pp. 12619-12629.

[92] C. M. Kim, M. Danielczuk, I. Huang, and K. Goldberg, "IPC-GraspSim: Reducing the Sim2Real gap for parallel-jaw grasping with the incremental potential contact model," in *Proc. Int. Conf. Robotics and Autom.*, Philadelphia, USA, 2022, pp. 6180–6187.

[93] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *Proc. Int. Conf. Robotics and Autom.*, Montreal, Canada, 2019, pp. 8973–8979.

[94] Y. Q. Du, O. Watkins, T. Darrell, P. Abbeel, and D. Pathak, "Auto-tuned sim-to-real transfer," in *Proc. IEEE Int. Conf. Robotics and Autom.*, Xi'an, China, 2021, pp. 1290–1296.

[95] J. Matas, S. James, and A. J. Davison, " Sim-to-real reinforcement learning for deformable object manipulation," in *Proc. 2nd Annu. Conf. Robot Learning*, Zürich, Switzerland, 2018, pp. 734–743.

[96] R. Jeong, Y. Aytar, D. Khosid, Y. X. Zhou, J. Kay, T. Lampe, K. Bousmalis, and F. Nori, " Self-supervised sim-to-real adaptation for visual robotic manipulation," in *Proc. IEEE Int. Conf. Robotics and Autom.*, Paris, France, 2019, pp. 2718–2724.

[97] P. Chang and T. Padif, " Sim2Real2Sim: Bridging the gap between simulation and real-world in flexible object manipulation," in *Proc. 4th IEEE Int. Conf. Robotic Computing*, Taichung, China, 2020, pp. 56–62.

[98] A. Allevato, E. S. Short, M. Pryor, and A. Thomaz, "TuneNet: One-shot residual tuning for system identification and sim-to-real robot task transfer," in *Proc. 3rd Annu. Conf. Robot Learning*, Osaka, Japan, 2019, pp. 445–455.

[99] O. M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. L. Weng, and W. Zaremba, " Learning dexterous in-hand manipulation," *Int. J. Robot. Res.*, vol. 39, no. 1, pp. 3–20, Jan. 2020.

[100] T. Power and D. Berenson, "Keep it simple: Data-efficient learning for controlling complex systems with simple models," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1184–1191, Apr. 2021.

[101] S. Scherzinger, A. Roennau, and R. Dillmann, "Contact skill imitation learning for robot-independent assembly programming," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Macau, China, 2019, pp. 4309–4316.

[102] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. L. Weng, Q. M. Yuan, W. Zaremba, and L. Zhang, "Solving Rubik's cube with a robot hand," arXiv preprint arXiv: 1910.07113, 2019.

[103] A. D. Allevato, E. S. Short, M. Pryor, and A. L. Thomaz, "Iterative residual tuning for system identification and sim-to-real robot learning," *Auton. Robot.*, vol. 44, no. 7, pp. 1167–1182, Sept. 2020.

[104] E. Heiden, C. E. Denniston, D. Millard, F. Ramos, and G. S. Sukhatme, " Probabilistic inference of simulation parameters via parallel differentiable simulation," in *Proc. Int. Conf. Robotics and Autom.*, Philadelphia, USA, pp. 3638–3645, 2022.

[105] M. Breyer, F. Furrer, T. Novkovic, R. Siegwart, and J. Nieto, "Flexible robotic grasping with sim-to-real transfer based reinforcement learning," arXiv preprint arXiv: 1803.04996, 2018.

[106] S. DI Castro Shashua, S. Mannor, and D. Di Castro, "Sim and real: Better together," in *Proc. 35th Conf. Neural Information Processing Systems*, 2021, pp. 6868–6880.

[107] A. Farchy, S. Barrett, P. MacAlpine, and P. Stone, "Humanoid robots learning to walk faster: From the real world to simulation and back," in *Proc. Int. Conf. Autonomous Agents and Multi-Agent Systems*, Saint Paul, USA, 2013, pp. 39–46.

[108] J. P. Hanna and P. Stone, "Grounded action transformation for robot learning in simulation," in *Proc. 31st AAAI Conf. Artificial Intelligence*, San Francisco, USA, 2017, pp. 4931–4932.

[109] J. Tan, T. N. Zhang, E. Coumans, A. Iscen, Y. F. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, " Sim-to-real: Learning agile locomotion for quadruped robots," arXiv preprint arXiv: 1804.10332, 2018.

[110] W. H. Yu, J. Tan, Y. F. Bai, E. Coumans, and S. Ha, "Learning fast adaptation with meta strategy optimization," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2950–2957, Apr. 2020.

[111] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Sci. Robot.*, vol. 4, no. 26, p. eaau5872, Jan. 2019.

[112] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Sci. Robot.*, vol. 5, no. 47, p. eabc5986, Oct. 2020.

[113] Z. W. Hong, Y. M. Chen, S. Y. Su, T. Y. Shann, Y. H. Chang, H. K. Yang, B. H. L. Ho, C. C. Tu, Y. C. Chang, T. C. Hsiao, H. W. Hsiao, S. P. Lai, and C. Y. Lee, "Virtual-to-real: Learning to control in visual semantic segmentation," arXiv preprint arXiv: 1802.00285, 2018.

[114] J. W. Zhang, L. Tai, P. Yun, Y. F. Xiong, M. Liu, J. Boedecker, and W. Burgard, "VR-goggles for robots: Real-to-sim domain adaptation for visual control," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1148–1155, Apr. 2019.

[115] A. Mitriakov, P. Papadakis, J. Kerdreux, and S. Garlatti, "Reinforcement learning based, staircase negotiation learning: Simulation and transfer to reality for articulated tracked robots," *IEEE Robot. Autom. Mag.*, vol. 28, no. 4, pp. 10–20, Dec. 2021.

[116] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra, " Sim2Real predictivity: Does evaluation in simulation predict real-world performance?" *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6670–6677, Oct. 2020.

[117] J. Truong, S. Chernova, and D. Batra, " Bi-directional domain adaptation for Sim2Real transfer of embodied navigation agents," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2634–2641, Apr. 2021.

[118] Z. W. Hong, Y. M. Chen, H. K. Yang, S. Y. S, T. Y. Shann, Y. H. Chang, B. H. L. Ho, C. C. Tu, T. C. Hsiao, H. W. Hsiao, S. P. Lai, Y. C. Chang, and C. Y. Lee, "Virtual-to-real: Learning to control in visual semantic segmentation," in *Proc. 27th Int. Joint Conf. Artificial Intelligence*, Stockholm, Sweden, 2018, pp. 4912–4920.

[119] F. Sadeghi and S. Levine, "CAD2RL: Real single-image flight without a single real image," in *Proc. Robotics: Science and Systems XIII*, Cambridge, USA, 2017.

[120] T. Du, J. Hughes, S. Wah, W. Matusik, and D. Rus, "Underwater soft robot modeling and control with differentiable simulation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 4994–5001, Jul. 2021.

[121] J. Collins, R. Brown, J. Leitner, and D. Howard, "Follow the gradient: Crossing the reality gap using differentiable physics (RealityGrad)," arXiv preprint arXiv: 2109.04674, 2021.

[122] D. Vázquez, A. M. López, J. Marín, D. Ponsa, and D. Gerónimo, "Virtual and real world adaptation for pedestrian detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 797–809, Apr. 2014.

[123] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proc. 14th European Conf. Computer Vision*, Amsterdam, the Netherlands, 2016, pp. 102–118.

[124] D. F. Liu, Y. Q. Wang, K. E. Ho, Z. W. Chu, and E. Matson, "Virtual world bridges the real challenge: Automated data generation for autonomous driving," in *Proc. IEEE Intelligent Vehicles Symp.*, Paris, France, 2019, pp. 159–164.

[125] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, USA, 2016, pp. 3234–3243.

[126] F. S. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, and J. M. Alvarez, " Effective use of synthetic data for urban scene semantic segmentation," in *Proc. 15th European Conf. Computer Vision*, 2018, pp. 86–103.

[127] Y. L. Tian, X. Li, K. F. Wang, and F. Y. Wang, "Training and testing object detectors with virtual images," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 2, pp. 539–546, Mar. 2018.

[128] X. Li, K. F. Wang, Y. L. Tian, L. Yan, F. Deng, and F. Y. Wang, "The ParallelEye dataset: A large collection of virtual images for traffic vision research," *IEEE Trans. Intell. Transp. Syst.* , vol. 20, no. 6, pp. 2072–2084, Jun. 2019.

[129] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, " Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *Int. J. Comput. Vis.* , vol. 126, no. 9, pp. 961–972, Sept. 2018.

[130] L. Z. Zhang, T. Wen, J. Min, J. C. Wang, D. Han, and J. B. Shi, "Learning object placement by inpainting for compositional data augmentation," in *Proc. 16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 566–581.

[131] Y. Chen, F. Rong, S. Duggal, S. L. Wang, X. C. Yan, S. Manivasagam, S. J. Xue, E. Yumer, and R. Urtasun, "GeoSim: Realistic video simulation via geometry-aware composition for self-driving," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Nashville, USA, 2021, pp. 7230–7240.

[132] X. L. Zhang, N. Tseng, A. Syed, R. Bhasin, and N. Jaipuria, "SIMBAR: Single image-based scene relighting for effective data augmentation for automated driving vision tasks," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, New Orleans, USA, 2022, pp. 3718–3728.

[133] A. Kar, A. Prakash, M. Y. Liu, E. Cameracci, J. Yuan, M. Rusiniak, D. Acuna, A. Torralba, and S. Fidler, " Meta-sim: Learning to generate synthetic datasets," in *Proc. IEEE/CVF Int. Conf. Computer Vision*, Seoul, Korea (South), 2019, pp. 4550–4559.

[134] J. Devaranjan, A. Kar, and S. Fidler, " Meta-sim2: Unsupervised learning of scene structure for synthetic data generation," in *Proc. 16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 715–733.

[135] A. Kishore, T. E. Choe, J. Kwon, M. Park, P. F. Hao, and A. Mittel, "Synthetic data generation using imitation training," in *Proc. IEEE/CVF Int. Conf. Computer Vision Workshops*, Montreal, Canada, 2021, pp. 3071–3079.

[136] Y. H. Chen, W. Li, C. Sakaridis, D. X. Dai, and L. Van Gool, "Domain adaptive faster R-CNN for object detection in the wild," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, Salt Lake City, USA, 2018, pp. 3339–3348.

[137] H. Zhang, G. Y. Luo, Y. L. Tian, K. F. Wang, H. B. He, and F. Y. Wang, " A virtual-real interaction approach to object instance segmentation in traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 863–875, Feb. 2021.

[138] X. Ouyang, Y. Cheng, Y. F. Jiang, C. L. Li, and P. Zhou, "Pedestrian-synthesis-GAN: Generating pedestrian data in real scene and beyond," arXiv preprint arXiv: 1804.02047, 2018.

[139] Z. Q. Zheng, Y. Wu, X. R. Han, and J. B. Shi, "ForkGAN: Seeing into the rainy night," in *Proc. 16th European Conf. Computer Vision*, Glasgow, UK, 2020, pp. 155–170.

[140] A. Vobecký, D. Hurych, M. Uřičář, P. Pérez, and J. Sivic, "Artificial dummies for urban dataset augmentation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 3, pp. 2692–2700, May 2021.

[141] A. El Sallab, I. Sobh, M. Zahran, and M. Shawky, "Unsupervised neural sensor models for synthetic LiDAR data augmentation," arXiv preprint arXiv: 1911.10575, 2019.

[142] J. Fang, D. F. Zhou, F. L. Yan, T. T. Zhao, F. H. Zhang, Y. Ma, L. Wang, and R. G. Yang, " Augmented LiDAR simulator for autonomous driving," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1931–1938, Apr. 2020.

[143] A. Lehner, S. Gasperini, A. Marcos-Ramiro, M. Schmidt, M. A. N. Mahani, N. Navab, B. Busam, and F. Tombari, "3D-VField: Adversarial augmentation of point clouds for domain generalization in 3D object detection," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, New Orleans, USA, 2022, pp. 17274–17283.

[144] M. Hahner, C. Sakaridis, M. Bijelic, F. Heide, F. Yu, D. X. Dai, and L. Van Gool, " LiDAR snowfall simulation for robust 3D object detection," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition*, New Orleans, USA, 2022, pp. 16343–16353.

[145] J. MarÍn, D. Vázquez, D. Gerónimo, and A. M. López, "Learning appearance in virtual scenarios for pedestrian detection," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, San Francisco, USA, 2010, pp. 137–144.

[146] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "VirtualWorlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Las Vegas, USA, 2016, pp. 4340–4349.

[147] X. Li, K. F. Wang, Y. L. Tian, L. Yan, F. Deng, and F. Y. Wang, "The paralleleye dataset: A large collection of virtual images for traffic vision research," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2072–2084, Jun. 2084.

[148] A. Savkin, T. Lapotre, K. Strauss, U. Akbar, and F. Tombari, "Adversarial appearance learning in augmented cityscapes for pedestrian recognition in autonomous driving," in *Proc. IEEE Int. Conf. Robotics and Autom.*, Paris, France, 2020, pp. 3305–3311.

[149] K. Strauss, A. Savkin, and F. Tombari, "Attention-based adversarial appearance learning of augmented pedestrians," arXiv preprint arXiv: 2107.02673, 2021.

[150] R. Zhi, Z. J. Guo, W. Q. Zhang, B. F. Wang, V. Kaiser, J. Wiederer, and F. B. Flohr, " Pose-guided person image synthesis for data augmentation in pedestrian detection," in *Proc. IEEE Intelligent Vehicles Symp.*, Nagoya, Japan, 2021, pp. 1493–1500.

[151] S. Manivasagam, S. L. Wang, K. Wong, W. Y. Zeng, M. Sazanovich, S. H. Tan, B. Yang, W. C. Ma, and R. Urtasun, "LiDARsim: Realistic LiDAR simulation by leveraging the real world," in *Proc. IEEE/CVF*

Conf. Computer Vision and Pattern Recognition, Seattle, USA, 2020, pp. 11164–11173.

[152] X. L. Pan, Y. R. You, Z. Y. Wang, and C. W. Lu, "Virtual to real reinforcement learning for autonomous driving," in *Proc. British Machine Vision Conf.*, London, UK, 2017.

[153] L. N. Yang, X. D. Liang, T. R. Wang, and E. Xing, "Real-to-virtual domain unification for end-to-end autonomous driving," in *Proc. 15th European Conf. Computer Vision*, 2018, pp. 553–570.

[154] Z. H. Yin, C. R. Li, L. T. Sun, M. Tomizuka, and W. Zhan, "Iterative imitation policy improvement for interactive autonomous driving," arXiv preprint arXiv: 2109.01288, 2021.

[155] J. Y. Zhou, R. Wang, X. Liu, Y. F. Jiang, S. Jiang, J. M. Tao, J. H. Miao, and S. Y. Song, "Exploring imitation learning for autonomous driving with feedback synthesizer and differentiable rasterization," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Prague, Czech Republic, 2021, pp. 1450–1457.

[156] M. Bansal, A. Krizhevsky, and A. S. Ogale, "ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst," in *Proc. Robotics: Science and Systems XV*, Freiburg im Breisgau, Germany, 2019.

[157] O. Scheel, L. Bergamini, M. Wołczyk, B. Osiński, and P. Ondruska, "Urban driver: Learning to drive from real-world demonstrations using policy gradients," in *Proc. 5th Conf. Robot Learning*, London, UK, 2021, pp. 718–728.

[158] A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, S. Karaman, and D. Rus, "Learning robust control policies for end-to-end autonomous driving from data-driven simulation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1143–1150, Apr. 2020.

[159] A. Amini, T. H. Wang, I. Gilitschenski, W. Schwarting, Z. J. Liu, S. Han, S. Karaman, and D. Rus, " VISTA 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles," in *Proc. Int. Conf. Robotics and Autom.*, Philadelphia, USA, 2022, pp. 2419–2426.

[160] B. Osiński, A. Jakubowski, P. Zięcina, P. Miłoś, C. Galias, S. Homoceanu, and H. Michalewski, " Simulation-based reinforcement learning for real-world autonomous driving," in *Proc. IEEE Int. Conf. Robotics and Autom.*, Paris, France, 2020, pp. 6411–6418.

[161] T. H. Wang, A. Amini, W. Schwarting, I. Gilitschenski, S. Karaman, and D. Rus, " Learning interactive driving policies via data-driven simulation," in *Proc. Int. Conf. Robotics and Autom.*, Philadelphia, USA, 2022, pp. 7745–7752.

[162] W. Yuan, M. Yang, C. X. Wang, and B. Wang, " VRDriving: A virtual-to-real autonomous driving framework based on adversarial learning," *IEEE Trans. Cogn. Dev. Syst.* , vol. 13, no. 4, pp. 912–921, Dec. 2021.

[163] A. El Sallab, I. Sobh, M. Zahran, and N. Essam, " LiDAR Sensor modeling and Data augmentation with GANs for Autonomous driving," arXiv preprint arXiv: 1905.07290, 2019.

[164] R. P. Saputra, N. Rakicevic, and P. Kormushev, "Sim-to-real learning for casualty detection from ground projected point cloud data," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Macau, China, 2019, pp. 3918–3925.

[165] R. Mitchell, J. Fletcher, J. Panerati, and A. Prorok, "Multi-vehicle mixed reality reinforcement learning for autonomous multi-lane driving," in *Proc. 19th Int. Conf. Autonomous Agents and Multiagent Systems*, Auckland, New Zealand, 2020, pp. 1928–1930.

[166] A. Stocco, B. Pulfer, and P. Tonella, "Mind the gap! A study on the transferability of virtual vs physical-world testing of autonomous driving systems," arXiv preprint arXiv: 2112.11255, 2021.

Qinghai Miao (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from the Graduate University of Chinese Academy of Sciences, in 2007. From 2017 to 2018, he was a Visiting Scholar with the School of Informatics, the University of Edinburgh, UK. He is currently an Associate Professor with the School of Artificial Intelligence, University of Chinese Academy of Sciences. His research interests include parallel intelligence, machine learning, computer vision, computer graphics, and intelligent transportation systems.

Yisheng Lv (Senior Member, IEEE) received B.S. and M.S. degrees from Harbin Institute of Technology, in 2005 and 2007, respectively, and Ph.D. degree from Chinese Academy of Sciences, in 2010. He is a Professor with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include artificial intelligence for transportation, autonomous driving, and intelligent transportation systems.

Min Huang (Member, IEEE) received the Ph.D. degree in computer sciences from Wuhan University, in 2007. From 2017 to 2018, she was a Visiting Scholar with the School of Informatics, the University of Edinburgh, UK. She is currently an Associate Professor with the School of Artificial Intelligence, University of Chinese Academy of Sciences. Her research interests include image processing, knowledge engineering, big data, and deep learning.

Xiao Wang (Member, IEEE) received the bachelor degree in network engineering from Dalian University of Technology, in 2011, and the Ph.D. degree in social computing from the University of Chinese Academy of Sciences, in 2016. She is currently a Professor with the School of Artificial Intelligence, Anhui University, and the President of the Qingdao Academy of Intelligent Industries. Her research interests include social network analysis, social transportation, cybermovement organizations, and multiagent modeling.

Fei-Yue Wang (Fellow, IEEE) received the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, USA, in 1990. He joined the University of Arizona in 1990 and became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences (CAS), under the support of the Outstanding Chinese Talents Program from the State Planning Council, and in 2002, was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS, and Vice President of Institute of Automation, CAS in 2006. He found CAS Center for Social Computing and Parallel Management in 2008, and became the State Specially Appointed Expert and the Founding Director of the State Key Laboratory for Management and Control of Complex Systems in 2011. His current research interests include methods and applications for parallel intelligence, social computing, and knowledge automation. He is a Fellow of INCOSE, IFAC, ASME, and AAAS. In 2007, he received the National Prize in Natural Sciences of China, numerous best papers awards from IEEE Transactions, and became an Outstanding Scientist of ACM for his work in intelligent control and social computing. He received the IEEE ITS Outstanding Application and Research Awards in 2009, 2011, and 2015, respectively, the IEEE SMC Norbert Wiener Award in 2014, and became the IFAC Pavel J. Nowacki Distinguished Lecturer in 2021. Since 1997, he has been serving as the General or Program Chair of over 30 IEEE, INFORMS, IFAC, ACM, and ASME conferences. He was the President of the IEEE ITS Society from 2005 to 2007, the IEEE Council of RFID from 2019 to 2021, the Chinese Association for Science and Technology, USA, in 2005, the American Zhu Kezhen Education Foundation from 2007 to 2008, the Vice President of the ACM China Council from 2010 to 2011, the Vice President and the Secretary General of the Chinese Association of Automation from 2008 to 2018, the Vice President of IEEE Systems, Man, and Cybernetics Society from 2019 to 2021. He was the Founding Editor-in-Chief (EiC) of the *International Journal of Intelligent Control and Systems* from 1995 to 2000, *IEEE ITS Magazine* from 2006 to 2007, *IEEE/CAA Journal of Automatica Sinica* from 2014−2017, *China's Journal of Command and Control* from 2015−2021, and *China's Journal of Intelligent Science and Technology* from 2019 to 2021. He was the EiC of the *IEEE Intelligent Systems* from 2009 to 2012, *IEEE Transactions on Intelligent Transportation Systems* from 2009 to 2016, *IEEE Transactions on Computational Social Systems* from 2017 to 2020. Currently, he is the President of CAA's Supervision Council, and the EiC of *IEEE Transaction on Intelligent*