

Interpreting the Latent Space of Generative Adversarial Networks using Supervised Learning

Toan Pham Van¹, Tam Minh Nguyen¹, Ngoc N. Tran¹, Hoai Viet Nguyen¹,
Linh Bao Doan¹, Huy Quang Dao¹, Thanh Ta Minh^{1,2}

¹R&D Lab, Sun* Inc

{pham.van.toan, nguyen.minh.tamb, tran.ngo.quang.ngoc, nguyen.viet.hoai,
doan.bao.linh, dao.quang.huyb, ta.minh.thanh}@sun-asterisk.com

²Le Quy Don Technical University, 236 Hoang Quoc Viet, Bac Tu Liem, Ha Noi
thanhtm@mta.edu.vn

Abstract—With great progress in the development of Generative Adversarial Networks (GANs), in recent years, the quest for insights in understanding and manipulating the latent space of GAN has gained more and more attention due to its wide range of applications. While most of the researches on this task have focused on unsupervised learning method, which induces difficulties in training and limitation in results, our work approaches another direction, encoding human’s prior knowledge to discover more about the hidden space of GAN. With this supervised manner, we produce promising results, demonstrated by accurate manipulation of generated images. Even though our model is more suitable for task-specific problems, we hope that its ease in implementation, preciseness, robustness, and the allowance of richer set of properties (compared to other approaches) for image manipulation can enhance the result of many current applications.

Index Terms—Latent space, Generative Adversarial Networks, Orthogonality Regularization, Supervised Learning

I. INTRODUCTION

The task of image generation has introduced many interesting applications in the world of computer vision, including image-to-image translation [1], [2], character drawing generation [3], and more. In recent years, a lot of effort has been done into enhancing Generative Adversarial Networks (GAN) [4] - a model that produced very promising results for the above applications. The main objective of this model is to generate realistic data from a random vector, in much lower dimension, sampled from a prior distribution. Called the “feature vector”, this random vector is believed to have encoded and condensed the important characteristics of the image; and the task of GAN is to generate an image from that information.

As the quality of image samples increases, more attention has also been drawn to latent space interpretation and image manipulation. This task studies how interactions between univariates in the latent space z of GAN results in generated images that we observe, particularly how to sample images with desired properties or manipulate attributes of synthesized images. The task can be usefully applied in Photograph Editing [5], Face Aging [6],...

Overall, many works on learning the latent space share a main limitation: they do not encode human’s prior knowledge into their studies. If one has some prior knowledge about

the domain, they can appropriately assume several factors of variation of the data. Taking human faces as an example, it’s common to presume that a successful GAN model’s latent space can encode numerous attributes including: faces pose, hair’s colors, attractiveness, baldness, smiling... This knowledge can be useful for task-specific training. Many approaches can be applied more generally; however, they can be either very time-consuming and difficult to train (such as InfoGAN [7]) or unable to learn specific factors of variation at all (for instance, vector arithmetic [8] and interpolation [8] between two images can only show that latent space indeed encodes meaningful semantics).

In this paper, we propose a method of learning the hidden latent space, and as a direct result, meaningfully manipulating GAN model’s generated images. Specifically, since the human prior knowledge is in the form of labels provided with the data, we shall call the domain of these features the label space of the input data, which we explore to learn a linear mapping between the latent space and its own attributes (i.e., the labels). The main idea is to control image generation by carefully manipulating the latent vectors through label variables.

Theoretically, we expect a high correlation between the latent variable z and the conditional label variable $y|G(z)$ given an image $G(z)$ generated from z . Therefore, we can learn a linear mapping between the 2 spaces. Upon further investigation, we found out that the high correlation of variables in label space induces highly correlated coefficients of the mapping, results in the attempt in changing one attribute will consequently change other related attributes. In order to alleviate this phenomenon, we apply the orthogonality regularization [9] into the loss function of the mapping to penalize similar coefficients. Our main contributions are summarized as followed:

- (i) We propose a supervised method to explicitly map the latent space with a meaningfully pre-defined semantic space. In contrast to other approaches; which often result in the *difficulties in training, limited number of properties* of the latent space that one can interpret and manipulate, and the *accuracy* of manipulation; our method attempt to resolve all those demerits with the help of supervision. Instead of worrying about training a new GAN model

(unlike InfoGAN), our model leverages of state-of-the-art GAN models to be responsible for generating most realistic images, then use simple model to perform the mapping and manipulation. This ease in implementing will allow many commercial applications to achieve competing results without too much cost of training.

- (ii) We provide a method reducing the effect of high correlation in the label space, resulting in more robust image manipulation. For instance, attractiveness is usually correlated with youth, heaviness of make-up, smiling.... Therefore, changing one attributes often results in the change of others. Reducing that effect will produce more robust manipulation.

II. RELATED WORKS

Generative Adversarial Networks (GAN) [4] decouple the quest of maximizing likelihood of images which looks real (given a distribution assumption) and creating unseen high quality images, to only focus on the later objective. The network relies on the competition of the generator, which generates images from a random noise vector that should look realistic enough to fool the discriminator, whose main job is to distinguish real and fake images. Due to its objective, GAN is able to sample the most realistic images compared to other approaches.

Training GAN is an tremendously challenging task, suffering from 4 main difficulties: non-convergence and instability [10], mode collapse [11], [12], unstable generator gradient [13], and highly sensitive to hyper-parameter tuning. PGGAN [14] overcomes these obstacles with a break-and-conquer method, progressively training the generator and discriminator from low-resolution to high, resulting in faster and more stable training. In order to fight mode collapse, they use a simpler version of mini-batch discrimination and some other training techniques to discourage the so-called “cat-and-mouse game” of the generator and discriminator. With those techniques implemented, PGGAN produces the most realistic images compared to other algorithms.

GAN latent space interpretation. An important point should be cleared out is that the latent space is structured by the generator, it itself has no meaning. Most researches about GAN’s latent space interpretation are based on unsupervised manner. For instance, paper [8] investigate the transition between two images by sampling a series latent vectors, which lies in a linear path connect two original random points, from a prior distribution. For example, an image of a man gradually transits to images of a woman through the above interpolation.

The method in [8] also employs the supervision of human classified images, attempts to show that latent space encodes meaningful semantics by using vector arithmetic. In particular, the mean of random vectors generating images of smiling women minus that of neutral women plus one of neutral men results in a vector that generates an image of a smiling man.

One novel idea in [7] is that instead of interpreting a fixed mapping, authors tried to create a controllable, interpretable mapping from the latent space to image space. Intuitively,

they used another easily, semantically understandable hidden variable c in a substantially lower dimension to encode the latent variable z . By control properties of c , they gain access to learn controllable, disentangled representation z of the image distribution.

Main drawbacks of other approaches to interpret GAN’s latent space. Overall, despite the potential of in learning controllable, interpretable latent space, InfoGAN and other referred methods in this paper suffer from the limited number semantics that they can interpret. Our approach offers a richer set of attributes, because of its supervised manner, the number of semantics we can manipulate depends on the number of attributes we can label. Note that, the desired human-label attributes should have as low correlation as possible.

Besides, difficulties in implementation and training (InfoGAN), lack of preciseness and robustness in manipulation are also major drawbacks in many other methods. Our method will enable us to resolve in a simpler manner.

III. PROPOSED METHODS

A. Classification model

We first utilize ResNet-34 [15] model pre-trained on ImageNet [16] as the backbone of our classification model. We then append the concatenation of adaptive average pooling and adaptive max pooling layers into the backbone, with the addition of 2 more dense blocks at last (each contains a dense layer, followed by Batch normalization, ReLU activation, and dropout). The model outputs a tensor in \mathbb{R}^{40} with each element from [0, 1] for every input image, indicating 40 predictions for 40 annotated attributes, where values near 0 indicate that the image almost does not have that attributes, and vice versa. We fine-tune the model with resized input images of size $128 \times 128 \times 3$ then use those weights to fine-tune the same model again with input size of $256 \times 256 \times 3$. This method enhances the accuracy of the model and will be described in more details in the training section.

With the aid of residual connection, we can increase the capacity of our model without the performance on training set getting hurt. We manage to prevent the over-fitting issue, induced by using ResNet-50 instead of ResNet-34, with heavy data augmentation and the same method on the input image size - which we used when training model with ResNet-34 as backbone. The promising result is further discussed in our Experiments section.

B. Linear mapping from latent space to semantic space

Interpreting the latent space in unsupervised learning is challenging, so we take advantage of the well - interpretable label space, to aid our understanding of the hidden space. We hypothesis that the hidden space can be compressed thus represented by another hidden space in lower dimension.

Since both latent and label variables can be perceived as factors explaining the data’s variation, we expect the label variable y to have high correlation with the latent variable z . Hence, our goal is to approximate the relationship between the 2 variables by a linear mapping.

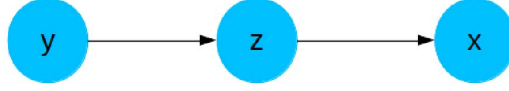


Fig. 1. Probabilistic Graphical Model. The graph indicate a dependent relationship between variables

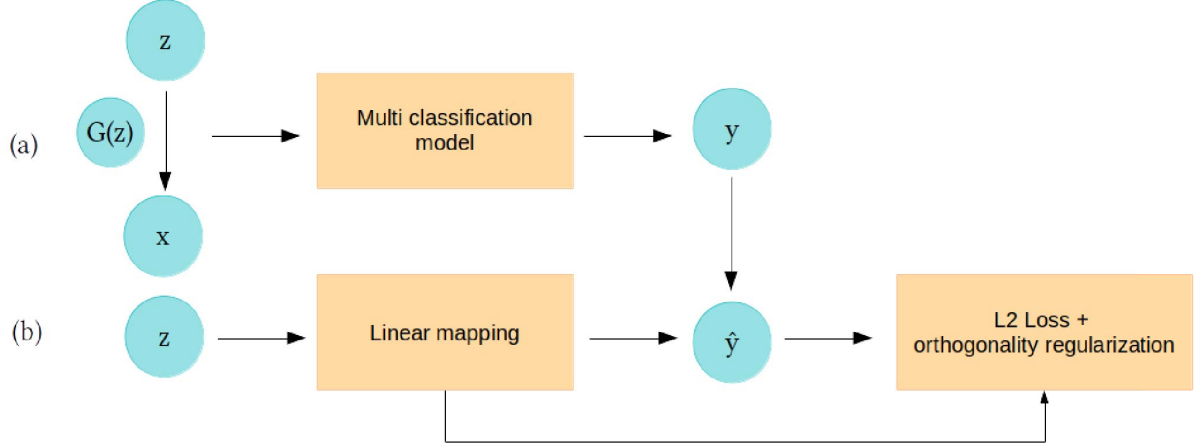


Fig. 2. Proposal Model Architecture

$$y = zW^T + b, \quad (1)$$

where W , b is the coefficient matrix and intercept term, respectively.

Our main quests are how to achieve a change in a latent vector gives a corresponding change in the generated image and how to change one attribute without changing others. In other words, for example, how can we make the girl in a generated image look younger without changing her hair or even her gender.

y_i is the attribute that we want to change by an amount of alpha, α . Gen-image is the result of passing random latent vector z through generator of GAN model. For instance, if y_i indicates attractiveness feature of the image and alpha is positive, we hope that the new latent vector: $z' = z + \alpha w_i$ will result in a generated image of the same person but looks younger.

$$z' = z + \alpha w_i \quad (2)$$

$$y' = z'W^T + b \quad (3)$$

$$\Leftrightarrow y + \Delta y = (z + \alpha w_i)W^T + b$$

$$\Leftrightarrow y + \Delta y = (zW^T + b) + \alpha w_i W^T$$

$$\Leftrightarrow \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \vdots \\ \Delta y_n \end{bmatrix} = \alpha w_i \begin{bmatrix} \Delta w_1^T \\ \Delta w_2^T \\ \vdots \\ \Delta w_n^T \end{bmatrix} = \begin{bmatrix} \Delta \alpha w_1^T w_i \\ \Delta \alpha w_2^T w_i \\ \vdots \\ \Delta \alpha w_n^T w_i \end{bmatrix}$$

$$\Rightarrow \Delta y > 0$$

However, the formula above pointed out that if w_i is 'similar' to other coefficients vector (measure by their cosine distance), change z to z' does not only manipulate y_i attributes but also causing changes in other attributes. Consequently, our desire is to make the person in the original generated image look prettier can end up with a totally different person who is actually more attractive.

In order to disentangle the linear mapping, we place an orthogonal penalty on the coefficient matrix. Therefore, the loss is computed as follows:

$$J(w) = \text{MSE}(f_w(z), y) + \lambda \|w^T w - I\|, \quad (4)$$

where $f_w \in \mathbb{R}^{512 \times 40}$ linearly transforms z to y
 λ is parameter coefficient, I is Identity matrix

TABLE I
MULTI-LABEL CLASSIFICATION MODEL’S RESULTS

epoch	train_loss	val_loss	acc_thres	fbeta	time (mm:ss)
14	0.207458	0.190790	0.916207	0.843260	04:19
15	0.193289	0.182742	0.919622	0.848778	04:20
16	0.188012	0.179203	0.921217	0.851056	04:21

The effect of this added regularization will be discussed in details in the Experiment section IV.

IV. EXPERIMENTS & RESULTS

A. Dataset

CelebFaces Attributes Dataset (CelebA) [17] is a large-scale face attributes dataset contains 10,177 celebrities images, each of which has 20 images. CelebA has large diversities, large quantities, rich annotations, including 202,599 number of face images, 5 landmarks locations, and 40 face attributes annotated by professional labeling company.

B. Classification model results

Beside the architecture design described in the above section, we apply several training techniques in order to boost up the performance of the multi-label classification model. In this part, we refer to FastAI [18] as the main source of inspiration.

Firstly, we set our learning rate schedule to a cyclical learning rate, also called the one-cycle policy [19]. If the learning rate is too small, mini-batch gradient cannot help parameters escape from an undesirable, narrow local minimum at early stage of training. Therefore, a practice of warming up the gradient (increase the gradient at first then decrease it later in the training course) can boost up learning at the beginning.

Additionally, learning rate discrimination is applied to fine-tune the model. This makes sense as lower convolutional layers detect low-level features of an image such as edges, curves... and those features for natural images are more or less the same despite of the fact that whether its the image of a car or a human face. However, higher convolutional layers detects more conceptual texture such as the existence of a tail or car headlights or the color of a human eyes. Consequently, their weights need to be updated more when training in a new dataset.

We use heavy augmentation as one of the main techniques to prevent over-fitting including horizontal flipping, wrapping, rotating but not cropping or brightening because these two transformations can partially delete or mutate important information of the image relating to attributes prediction.

Lastly, we apply a simple methodology of progressively multiple input size training as described in section III above. We experiment that training with this technique gives a slightly better result with a little help of reducing over fitting than starting with images size of $256 \times 256 \times 3$ in the first place. The technical reason that we are able to implement this technique is based on our usage of adaptive pooling layers, which enable us to obtain fixed output size despite various input sizes.

After 16 epochs of early stopped fine-tuning, our classification model ends up with 92.1% of accuracy and 0.85 F1 score.

C. Linear mapping and attributes manipulation

1) *Overall setting of experiments:* At first, we sample 3000 random noise vectors from multivariate standard Gaussian distribution. We then use our classification model to obtain 3000 label vectors, each contains 40 attributes, from 3000 generated images using these above random noises (Note that, each elements of each label vector will be in $[0, 1]$ instead of $\{0, 1\}$).

Its worth p out a non-trivial implementation detail is that we generate $512 \times 512 \times 3$ images for each random noise vector in \mathbb{R}^{512} , then feed to the classification model a resized $256 \times 256 \times 3$ images, instead of generating $256 \times 256 \times 3$ images from random vectors in \mathbb{R}^{256} . One reason is that the random noise in \mathbb{R}^{512} allows mapping to a higher qualification image.

Secondly, we apply a linear transformation to approximate the mapping between the latent space z and label space y . In order to increase cosine distance between coefficients vectors, we add the orthogonality regularization to the loss with weight λ equal 2. The result is robust to this choice of λ considering many trials.

2) *Analyze results:* Fig 3 shows the manipulations of generated images using formula (2) and (3) with and without penalty for high-cosine-similarity coefficients.

With orthogonality regularization, increasing the Young attributes only makes the man look younger, unlike without the regularization, the manipulation causes change in many other properties including gender. Take a closer look at table Table II, where we pass original generated; manipulated with and without penalties images through the multi-label classification model and observe the change in their confidence score, we recognize that, increasing younger changes the man’s gender, makes him wear heavy makeup, lipstick and other related attributes. While comparing to the changes in those same attributes, the table indicates that with the help of regularization, making the man look younger relatively does not cause severe effect on other properties. We observe that same phenomenon in manipulating many other images.

For numerically reasoning why, Table III compares the cosine distances between Young and other semantics’ coefficients. Coefficients of Young is highly similar to those of attributes, a heavy makeup, increasing Young also increases those attributes. The opposite happens to Male, Chubby, or Bald, and more. Intuitively, being bald makes us look older. However, orthogonality regularization successively prevents these changes. It makes Young’s coefficient much less ‘similar’ to others’. The same observation shares among other attributes’ coefficients.

Fig 3 also demonstrates the success of our approach (1st and 2nd row) through examples. Additionally notice that, women generally have more attributes that can be independently manipulated than men, typically heavy makeup, oval face,

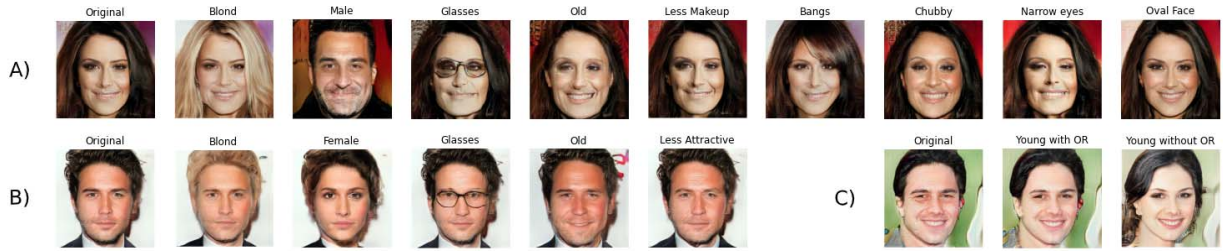


Fig. 3. Image Manipulation with latent vector

TABLE II
CHANGES IN HOW MANIPULATING YOUNG ATTRIBUTES AFFECTS OTHER ATTRIBUTES, WITH AND WITHOUT ORTHOGONALITY REGULARIZATION

	Original	tfm. w/o reg.	abs_diff_no_reg	tfm_attr_reg	abs_diff_reg
Male	0.999618	0.000217	0.999401	0.997077	0.002541
Makeup	0.001739	0.999788	0.998050	0.009618	0.007879
Lipstick	0.001942	0.999757	0.997815	0.014711	0.012769
Earrings	0.030412	0.814261	0.783849	0.088462	0.058051
BagsUnderEyes	0.782150	0.027579	0.754571	0.671357	0.110794
WavyHair	0.118599	0.870955	0.752356	0.190311	0.071713
SoClockShadow	0.688798	0.000454	0.688344	0.672848	0.015950
BigNose	0.750373	0.073389	0.676984	0.643183	0.107190
OvalFace	0.337116	0.944806	0.607690	0.531252	0.194136
DoubleChin	0.542601	0.001489	0.541111	0.3252063	0.290538
BushyEyebrows	0.960830	0.421305	0.539525	0.972407	0.011577
Young	0.612252	0.991196	0.378944	0.896836	0.284584
ArchedEyebrows	0.89847	0.463946	0.374099	0.065678	0.024169
NarrowEyes	0.391831	0.067585	0.324246	0.296349	0.095482

TABLE III
COMPARE COSINE DISTANCE OF YOUNG AND OTHER ATTRIBUTES, WITH AND WITHOUT ORTHOGONALITY REGULARIZATION

	Without Regularization	With Regularization
Young	1.000000	1.000000
Attractive	0.859559	0.313507
HeavyMakeup	0.795307	-0.002151
Lipstick	0.791769	0.282624
NoBeard	0.599438	0.262185
BigLips	0.542247	-0.000376
ArchedEyebrows	0.513396	0.042719
OvalFace	0.453567	-0.017463
WavyHair	0.430066	0.036656
PointyNose	0.413676	0.022878
DoubleChin	-0.786877	0.019917
Male	-0.769961	0.039103
BagsUnderEyes	-0.754783	0.015170
GrayHair	-0.749704	0.014643
BigNose	-0.734693	0.034192
WearingTie	-0.669467	0.038437
Bald	-0.585021	0.042678
RecedingHairline	-0.523914	0.009939
Goatee	-0.495711	0.038485

big lips... We suspect that the natural differences in men's and women's appearance and taste make the input data-set imbalance which has fewer men wearing makeup and possess

oval face than women and more.

D. Applications

With the ease of implementation, accuracy, robustness, and flexibility in image manipulation, we believe that our model can boost up and enrich the performance of several Face Editing, Age Editing and Photo Makeover applications in the market.

E. System configuration

Our experiments are conducted on a computer with Intel Core i5-7500 CPU @3.4GHz, 32GB of RAM, GPU GeForce GTX 1080 Ti, and 1TB SSD hard disk. The models are implemented with the PyTorch [20] framework.

V. CONCLUSION AND FUTURE WORK

Our model proposal utilizes human prior knowledge to learn the mapping of a rich set of meaningful, interpretable semantics to GAN's latent space. Besides its main advantages as the ease in learning the mapping using our model and the number of attributes it is able to manipulate, our model suffers from a major drawback. That is the expense of labeling due to the supervised manner. In the future, we would like to combine the unsupervised-based Data labeling tool developed by FastAI to tremendously aid the process of data annotation.

ACKNOWLEDGMENT

This work is partially supported by *Sun-Asterisk Inc.* We would like to thank our colleagues at *Sun-Asterisk Inc* for their advice and expertise. Without their support, this experiment would not have been accomplished.

REFERENCES

- [1] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," 2018.
- [2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2016.
- [3] Y. Jin, J. Zhang, M. Li, Y. Tian, H. Zhu, and Z. Fang, "Towards the automatic anime characters creation with generative adversarial networks," 2017.
- [4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.
- [5] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez, "Invertible conditional gans for image editing," 2016.
- [6] G. Antipov, M. Baccouche, and J. Dugelay, "Face aging with conditional generative adversarial networks," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 2089–2093.
- [7] X. Chen, Y. Duan, R. Houthoof, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," 2016.
- [8] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015.
- [9] N. Bansal, X. Chen, and Z. Wang, "Can we gain more from orthogonality regularizations in training deep cnns?" 2018.
- [10] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," 2016.
- [11] D. Berthelot, T. Schumm, and L. Metz, "Began: Boundary equilibrium generative adversarial networks," 2017.
- [12] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," 2016.
- [13] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," 2017.
- [14] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," 2017.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.
- [16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.
- [17] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- [18] J. Howard *et al.*, "fastai," <https://github.com/fastai/fastai>, 2018.
- [19] L. N. Smith, "A disciplined approach to neural network hyperparameters: Part 1 – learning rate, batch size, momentum, and weight decay," 2018.
- [20] N. Ketkar, *Introduction to PyTorch*. Berkeley, CA: Apress, 2017, pp. 195–208. [Online]. Available: https://doi.org/10.1007/978-1-4842-2766-4_12