Image Synthesis & Background Replacement

— Graph-to-Graph Triple U-net Generator & Patch-GAN Discriminator

MO Chenyun	QUAN Xinyuan	XU Penggan	ZENG Yang	ZHENG Haixun
122090406	122090442	122090625	122090714	122020291

Abstract

This paper presents an innovative approach to image synthesis and background replacement using a Graph-to-Graph Triple U-net Generator and Patch-GAN Discriminator. Unlike existing tools that merely replace backgrounds without considering object size and position, our model integrates three critical functions—object placement, background harmonization, and shadow generation—into a unified framework. By employing dynamic concatenation in the Triple U-net architecture and leveraging patch-level discrimination, the model achieves enhanced spatial consistency and computational efficiency. The proposed method is validated on the PPR10K dataset and fine-tuned with CUHKSZ-specific images, showing promising applications in campus event promotion and personalized student content creation. Limitations such as image blurriness and generalization challenges are discussed, with future improvements targeting architectural optimization and dataset expansion.

1. Introduction

1.1. Background

In the digital age, the demand for image - related content is skyrocketing, and image synthesis and background replacement techniques have become increasingly important.

For individuals, the ability to replace the background of an image breaks the constraints of time and space. In the past, people had to physically visit specific locations to capture photos with desired backgrounds. However, advanced image synthesis and background replacement technology provides a convenient way for individuals to create personalized photos for social media sharing, personal mementos, or professional use.

From the perspective of universities, image synthesis and background replacement can contribute to showcasing their unique campus culture, facilities, and events. For instance, by replacing the background of a high-table dinner photo to different students, it can better attract potential students and highlight distinctiveness and dynamism.

1.2. Advancements Over Existing Work

Firstly, most existing tools for image synthesis and background replacement are rather simplistic. They merely change the background without taking into account the size and position of the objects within the image (Figure 8 in Appendix). This oversight often leads to unrealistic - looking results, where the object appears out of proportion or inappropriately placed in the new background. In contrast, our model has the remarkable ability to learn and adjust the reasonable size and position of the object in the new background. This improvement is crucial as it provides a more realistic and visually - pleasing output, meeting the growing demands for high - quality image processing.

Secondly, existing research methods typically involve separate models for object placement, background harmonization, and shadow generation, and these processes are carried out sequentially or parallelly [2] (Figure 9 in Appendix). This approach is not only time - consuming but also results in inconsistent generated images. The lack of integration means that the output of one model may not be well - coordinated with the others, leading to an unharmonious overall effect. Our research addresses this issue by integrating these three functions into a single unified model. This integration brings about several benefits. It enhances the efficiency of the image - processing workflow by reducing the number of operational steps and thus saving valuable time. Moreover, the integrated model can generate more uniform and coordinated results. Through joint optimization of the three functions, the model can better balance object placement, background harmonization, and shadow generation, creating a more harmonious and realistic final image.

2. Data Collection and Preprocessing

2.1. Data Collection

For the purpose of this research, we assembled a diverse and comprehensive dataset to train and evaluate our model. We sourced 5661 single - person images from the publicly available repository https://github.com/csjliang/PPR10K. This dataset provides a rich source of images with varying poses, lighting conditions, and back-

grounds, which are essential for training a model with good generalization ability.

In addition to the general dataset, we also incorporated single - character images from CUHKSZ along with their corresponding backgrounds. These CUHKSZ - specific images are particularly valuable as they allow us to fine - tune the model for scenarios that are relevant to our local context, such as campus - related events.

2.2. Data Preprocessing

2.2.1. Graph Matting

The initial step in our data preprocessing pipeline is graph matting. We utilized the birefnet - portrait model, which is accessible as the rembg package in Python (https://github.com/danielgatis/rembg). By leveraging this method, we obtain three important outputs: a mask image that clearly demarcates the boundaries of the person, a separated person image, and an incomplete background image (Figure 10 in Appendix).

2.2.2. Graph Inpainting

Following graph matting, we perform graph inpainting using the LaMa [4] method (https://github.com/advimman/lama). The primary objective of graph inpainting is to eliminate any indication of the original position of the person in the background. When the person is removed from the original background during graph matting, it leaves behind a void or an area with missing information. Graph inpainting fills in this void in a way that makes the background appear natural and seamless, without any visible signs of the person's previous presence (Figure 11 in Appendix).

2.2.3. Augmentation

To enhance the robustness and generalization ability of our model, we implement data augmentation techniques (Figure 12 in Appendix). This involves a series of operations such as resizing, rotating, shifting, and flipping the images. This enables the model to learn invariant features that are not affected by changes in scale or rotation, thereby improving its ability to recognize and process objects in different poses.

Shifting and flipping operations further increase the diversity of the training data. These operations together create a more diverse training dataset, which is essential for training a model that can generalize well to unseen data.

Through these augmentation techniques, the model is able to learn the optimal size and position of the person in the new background, especially through the mechanism of the Generative Adversarial Network (GAN). By training on a more diverse set of images, the model can better adapt to different scenarios.

3. Methodology

3.1. Objective

The objective of our GAN can be expressed as

$$\begin{aligned}\mathcal{L}_{cGAN}(G,D) =& \mathbb{E}_{\mathbf{X},y}[\log D(\mathbf{X},y)] \\ &+ \mathbb{E}_{\mathbf{X}}[\log(1 - D(\mathbf{X}, G(\mathbf{X})))] \end{aligned} \quad (1)$$

where $\mathbf{X} = (x_{mask},\ x_{people},\ x_{background})$, y denotes the ground truth label. G tries to minimize this objective against an adversarial D that tries to maximize it, i.e. $G_* = \text{argmin}_G \max_D L_{cGAN}(G, D)$.

Previous approaches have found it beneficial to mix the GAN objective with a more traditional loss, such as L2 and L1 distance, in which shows that using L1 distance rather than L2 as L1 encourages less blurring [1]. The discriminator's job remains unchanged, but the generator is tasked to not only fool the discriminator but also to be near the ground truth output in L1 sense:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{\mathbf{X},y}[\|y - G(\mathbf{X})\|_1] \quad (2)$$

Additionally, we also introduce the perceptual loss function, which compares feature representations extracted from pretrained networks. By focusing on high-level semantic content rather than pixel-wise accuracy, perceptual loss enables the model to generate sharper and more visually coherent images, thus improving the overall perceptual quality. The formula of perceptual loss can be expressed as

$$\mathcal{L}_{\text{perc}}(\mathbf{X}, y) = \sum_{l=1}^{L} \frac{\lambda_l}{C_l H_l W_l} \left\|\phi_l(G(\mathbf{X})) - \phi_l(y)\right\|_2^2 \quad (3)$$

where $G(\mathbf{X})$ and y denote the generated and real images, respectively; $\phi_l(\cdot)$ is the feature map of VGG-16 extracted from the l-th layer of a pretrained convolutional network; C_l, H_l, and W_l are the number of channels, height, and width of that feature map (used to normalize the squared error); λ_l is a weighted hyperparameter for the loss at layer l; and the summation aggregates differences across layers to capture both low- and high-level semantic information.

The total loss of our model can be expressed as

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{cGAN}(G, D) + \mathcal{L}_{L1}(G) + \mathcal{L}_{\text{perc}}(G(\mathbf{X}), y) \quad (4)$$

3.2. Network architectures

We modify our generator and adopt our discriminator architectures from Pix2Pix in [1]. Details of the architecture are provided in the Appendix, with key features discussed below.

3.2.1. Triple U-net Generator

We propose a novel Triple U-Net Generator architecture (Figure 1) designed to synthesize composite images by integrating segmented human figures, background scenes, and binary masks. The generator extends the classical U-Net paradigm by introducing three parallel encoder streams-each dedicated to a distinct input modality-concatenated at bottleneck hidden layers-followed by a unified decoder that reconstructs a full-color synthetic image.

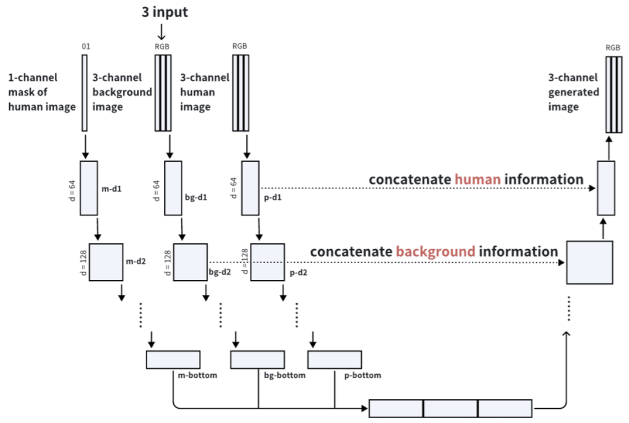

Figure 1. Triple-U-Net Generator

Down-Sampling Process (Encoder) For the encoder part, the network accept three inputs: one-channel binary mask delineating the human figure sihouette, a three-channel background image (rgb), and a three-channel human figure image (rgb) extracted from the original scene. Each input is processed by a identical sequence of down-sampling blocks. Upon reaching the bottleneck stage, the three encoded feature tensors-corresponding to human appearance, background context, and human mask-are concatenated along the channel dimension to form a comprehensive representation of all pertinent scene attributes. This tripartite fusion enables the generator to jointly reason about object-background compatibility, spatial alignment dictated by the mask, and color consistency.

Up-Sampling Process (Decoder with skip connection) The traditional U-Net integrates the corresponding consistency encoder layers during each up-sampling step [3]. However, three different encoding paths-human, background, mask-were provided by our Triple-U-Net down-sampling process. Hence, we novelly introduced a strategy-Alternating Connecting Method-that alternates between re-instating background scene context and refining foreground details. To be detailed, each up-sampling stage integrates skip connections that concatenate features from background and human features encoder layers iteratively. Consider at up-sampling layer n, features originating form the background encoder are merged with the up-sampled representation. Then, the subsequent up-sampling layer $n + 1$ selectively integrate human features from the human encoder. The hierarchical interplay between background and human features preserves multi-scale information from each modality and balances global structural integrity with local detail enhancement.

3.2.2. Patch GAN Discriminator

For the discriminator of our Generative Adversarial Network (GAN), we adopt the PatchGAN architecture, which is inspired by the discriminator used in the pix2pix framework [1] . Unlike traditional GAN discriminators that classify entire images as real or fake, the PatchGAN discriminator classifies each local image patch as real or fake. This method enhances the model's ability to focus on local details rather than global characteristics, and help capture high-frequencies.

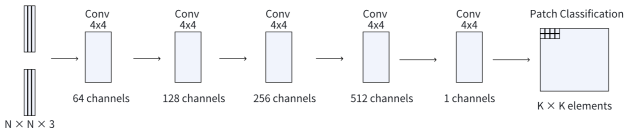

Figure 2. PatchGAN Discriminator

Architecture The PatchGAN discriminator processes the input image through a series of convolutional layers. Each layer consists of a 4x4 convolution kernel followed by a Leaky ReLU activation function. The input image is gradually transformed into feature maps with increasing depth, capturing different levels of spatial information. The final output of the PatchGAN is a probability map where each pixel represents a decision on whether the corresponding patch in the image is real or fake.

In the original implementation, the size of the patch is typically set to 70x70 pixels, but smaller or larger patch sizes can also be explored depending on the task.

Improvements over Traditional Discriminators The PatchGAN approach has several advantages over traditional discriminators:

- Whole-image classification in traditional GANs is replaced by patch-level probability classification.
- It preserves the spatial structure of the input image, maintaining local details that might be lost in global classifications.
- The computational efficiency is improved, as the discriminator only processes smaller, local regions instead of the entire image at once.

- By focusing on local regions, the PatchGAN helps in generating finer details, which is crucial for tasks like image-to-image translation and background synthesis.

These enhancements allow for better fine-grained control over the image generation process, leading to sharper, more realistic output images, which is suitable for our image synthesis and background replacement task.

Optimization The optimization alternates between updating the discriminator and the generator, applying the Adam optimizer with a learning rate of 0.0002, and momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. This ensures that both components of the GAN network improve simultaneously.

4. Numerical and Experimental Results

4.1. Numerical Results

In this section, we evaluate the performance improvements of our generator model using both L1 loss and perceptual loss metrics.

4.1.1. L1 Loss

The L1 loss, also known as the mean absolute error, measures the average absolute differences between the generated image and the ground truth. It encourages pixel-wise accuracy and reduces large deviations. During training, it decreased from an initial value of 1.4 to 1.0, representing a reduction of 28. 6%. This steady improvement indicates that the generator progressively learned to produce outputs that are more closely aligned with the target images at the pixel level.

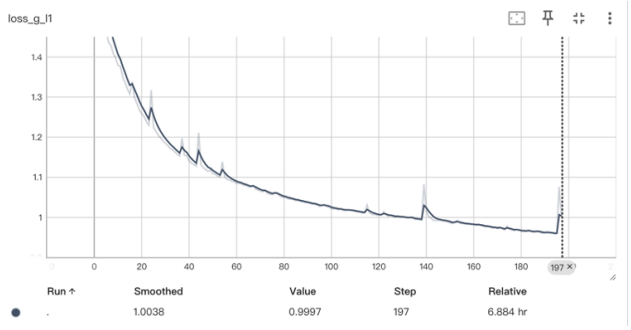

Figure 3. L1 Loss of Generator

4.1.2. Perceptual Loss

In addition to pixel-wise metrics, we evaluated the model using perceptual loss, which measures the semantic similarity between the generated and target images based on high-level feature representations extracted from a pre-trained convolutional neural network (typically VGG). This loss captures structural and perceptual fidelity beyond simple pixel alignment. During the course of training, the perceptual loss decreased from 0.044 to 0.032, resulting in a reduction of 27. 2%. This indicates that the generator not only improved in reproducing low-level image details but also became more effective at preserving semantic consistency and visual realism in the output.

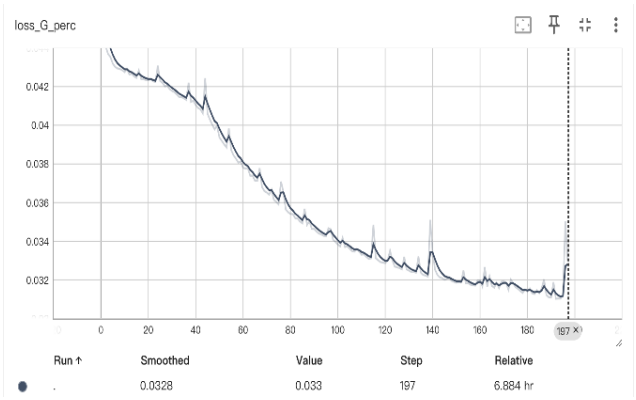

Figure 4. Perceptual Loss of Generator

4.2. Experimental Results

To qualitatively evaluate the effectiveness of our model, we conducted an image composition experiment. The input consisted of two elements: an isolated figure and a background image depicting a High Table Dinner scene. The goal was to seamlessly integrate the foreground figure into the background while preserving visual realism. The generated composite image demonstrates strong performance in several aspects. The figure is rendered at an appropriate scale and spatial position, maintaining consistency with the perspective and context of the background. Furthermore, the boundary blending between the foreground and background is handled smoothly, resulting in a natural appearance without visible artifacts. These results indicate the model's capacity to perform high-quality semantic-aware image composition.

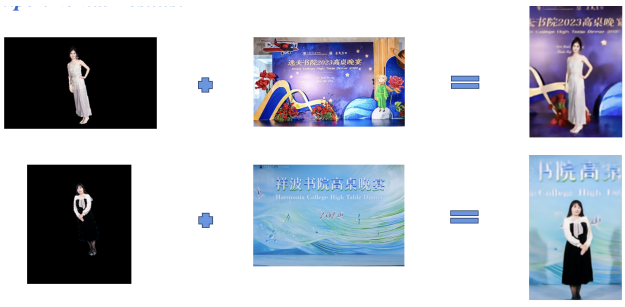

Figure 5. Generated composite image: the person is placed in a High Table Dinner background with realistic scale, position, and boundary blending.

5. Limitations and Future Improvements

5.1. Blurriness of the Image

Our framework exhibits reduced visual fidelity at lower training resolutions due to limited feature extraction capacity.

Figure 6. Progressive improvement in image sharpness with increasing training resolution : Test outputs from models trained at (a) 128×128, (b) 256×256, and (c) 512×512 pixel resolutions.

As shown in Figure 6, models trained at 128×128 pixels produce perceptually blurred outputs with poor high-frequency detail preservation (e.g., texture edges). Increasing resolution to 512×512 improves sharpness through multi-scale feature fusion in the Triple U-Net generator, though residual blurriness persists in complex regions (e.g., facial details). While scaling to 1024×1024 further enhances quality, computational overhead (tripled training time, memory saturation on RTX 4090) necessitated adopting 512×512 as a practical compromise. Architectural limitations of the Patch-GAN discriminator—prioritizing local realism over global coherence—may also contribute to this issue. Future solutions could integrate perceptual loss functions or advanced upsampling techniques to address high-resolution deficiencies.

5.2. Out-of-sample Performance Issue

The model's generalization capability is constrained by training data specificity (CUHKSZ dataset dominated by formal attire/controlled environments).

As illustrated in Figure 7, domain mismatches (e.g., casual outfits synthesized into formal dining scenes) result in semantic incoherence due to underrepresented category feature extraction. The single-framework integration of object placement, background harmonization, and shadow generation exacerbates vulnerability to novel compositions (e.g., cross-domain attire-scene pairings). This reflects a broader GAN challenge: edge-case data scarcity limits out-of-distribution robustness despite augmentation. Solutions require expanded dataset diversity or domain adaptation techniques to improve cross-domain transferability.

Figure 7. Sample test result of domain mismatch in out-of-sample inference.

6. Conclusions

This paper presents a unified framework for image synthesis and background replacement through a Graph-to-Graph Triple U-Net generator and Patch-GAN discriminator. By integrating object placement, background harmonization, and shadow generation into a single model, our approach achieves enhanced spatial consistency and computational efficiency compared to existing methods. The dynamic concatenation in the Triple U-Net enables multi-scale feature fusion, while the Patch-GAN discriminator prioritizes local realism, yielding visually coherent outputs validated on the PPR10K and CUHKSZ datasets. Experimental results demonstrate reduced L1 loss and perceptual loss, highlighting the model's ability to preserve semantic details and structural fidelity. However, limitations persist in low-resolution blurriness and generalization for out-of-distribution scenarios due to training data specificity. Future work will explore architectural refinements (e.g., attention-based upsampling), domain adaptation techniques, and expanded datasets to address these challenges. This work advances practical applications in personalized content creation and campus event promotion, offering a foundation for robust, end-to-end image composition systems.

Code and Data

Due to the size constraints, the complete dataset associated with this work cannot be hosted directly. However, all data are made publicly available for download via OneDrive. And the entirety of the codebase—including the model architecture (Triple_U_Net), data preprocessing pipelines, and augmentation scripts—is openly accessible in this repository `https://github.com/yang797/Image-Synthesis-Background-Replacement`. Researchers are encouraged to review the README.md for detailed instructions on dataset retrieval, dependency setup, and reproducibility.

References

[1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2, 3, 1

[2] Li Niu, Wenyan Cong, Liu Liu, Yan Hong, Bo Zhang, Jing Liang, and Liqing Zhang. Making images real again: A comprehensive survey on deep image composition. *arXiv preprint arXiv:2106.14490*, 2021. 1

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 3

[4] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021. 2

7. Appendix

7.1. Supplementary Materials for Advancement Over Existing Work

Figure 8. Performance of existing background replacement tools: We can see the existing tools do not consider the size and position of the person, just simply copy and paste the baby, making the background replacement performance not satisfying

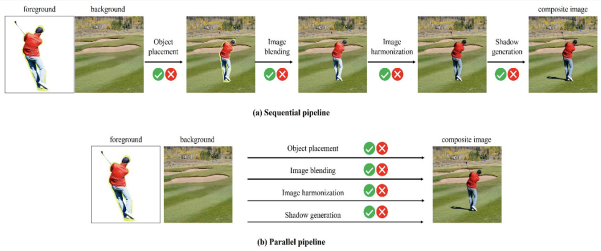

Figure 9. Previous research [2] perform multiple subtasks (e.g., object placement, image blending, image harmonization, shadow generation sequentially or parallelly to achieve the goal of image composition

7.2. Supplementary Materials for Data Preprocessing

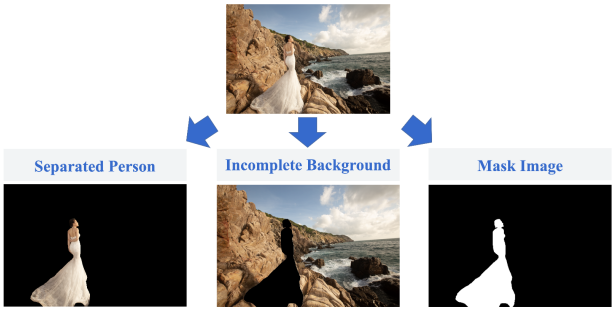

Figure 10. Sample Output of Graph Matting: Including the separated person, the incomplete background, and the mask image.

Figure 11. Example of Graph Inpainting: Input the incomplete background after graph matting, then inpainting the background. This aims to eliminate any indication of the original position of the person in the background

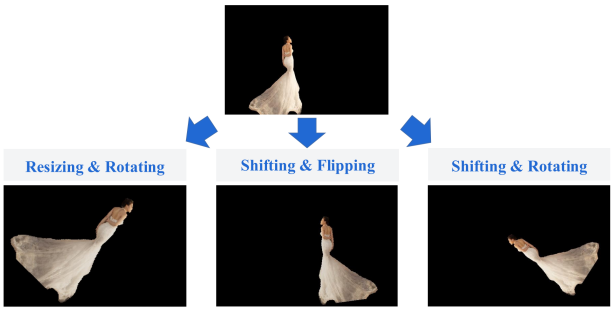

Figure 12. Different methods of graph augmentation, including resizing, rotating and shifting

7.3. Network Structure

We adapt our network architectures from those in [1]. Let $C(k)$ denote a Convolution-BatchNorm-ReLU layer with K filters. $CD(k)$ denotes a Convolution-BatchNorm-Dropout-ReLU layer with a dropout rate of 20%. All convolutions are 4×4 spatial filters applied with stride 2. Convolutions in the encoder, and in the discriminator, down-

sample by a factor of 2, whereas in the decoder they upsample by a factor of 2.

7.3.1. Generator Structure

The encoder-decoder structure consist of:

- **Human Encoder:**
  $C(64) - C(128) - C(256) - C(512) - C(512)$
- **Background Encoder:**
  $C(64) - C(128) - C(256) - C(512) - C(512)$
- **Mask Encoder:**
  $C(64) - C(128) - C(256) - C(512) - C(512)$
- **Decoder:**
  $C(1536) - C(1536) - C(1024) - C(512) - C(192)$

After the last layer in the decoder, a convolution is applied to map to the number of output channels 3, followed by a Tanh function. As an exception to the above notation, Batch-Norm is not applied to the first $C(64)$ layer in the encoder. All ReLUs in the encoder are leaky, with slope 0.2, while ReLUs in the decoder are not leaky.

The U-Net architecture is identical except with skip connections between each layer i in the encoder and layer $n-i$ in the decoder, where n is the total number of layers. The skip connections concatenate background activations from background encoder layer i to decoder layer $n-i$, and concatenate human activations from human encoder layer $i-1$ to decoder layer $n-i+1$. The whole skip connection is done in this manner iteratively.

7.3.2. Discriminator Structure

The 14×14 **discriminator** architecture is:
$C(64) - C(128) - C(256) - C(512)$

After the last layer, a convolution is applied to map to a 1-dimensional output, followed by a Sigmoid function. As an exception to the above notation, BatchNorm is not applied to the first $C(64)$ layer. All ReLUs are leaky, with slope 0.2.