



# OPEN Deeply supervised two stage generative adversarial network for stain normalization

Zhe Du<sup>1,2</sup>, Pujing Zhang<sup>1,2</sup>, Xiaodong Huang<sup>1,2</sup>, Zhigang Hu<sup>1</sup>, Gege Yang<sup>1,2</sup>, Mengyang Xi<sup>1</sup> & Dechun Liu<sup>2,3</sup>✉

The color variations present in histopathological images pose a significant challenge to computational pathology and, consequently, negatively affect the performance of certain pathological image analysis methods, especially those based on deep learning techniques. To date, several methods have been proposed to mitigate this issue. However, these methods either produce images with low texture retention, perform poorly when trained with small datasets, or have low generalization capabilities. In this paper, we propose a Deep Supervised Two-stage Generative Adversarial Network known as DSTGAN for stain-normalization. Specifically, we introduce deep supervision to generative adversarial networks in an innovative way to enhance the learning capacity of the model, benefiting from different model regularization methods. To make fuller use of source domain images for training the model, we drew upon semi-supervised concepts to design a novel two-stage staining strategy. Additionally, we construct a generator that can capture long-distance semantic relationships, enabling the model to retain more abundant texture information in the generated images. In the evaluation of the quality of generated images, we have achieved state-of-the-art performance on TUPAC-2016, MITOS-ATYPIA-14, ICIAR-BACH-2018 and MICCAI-16-GlaS datasets, improving the precision of classification and segmentation by 5.2% and 4.2%, respectively. Not only has our model significantly improved the quality of the stained images compared to existing stain normalization methods, but it also has a positive impact on the execution of downstream classification and segmentation tasks. Our method has further reduced the effect that staining differences have on computational pathology, thereby improving the accuracy of histopathological image analysis to some extent.

**Keywords** Computational pathology, Stain normalization, Generative adversarial networks (GAN), Deep supervision (DS), Semi-supervised learning

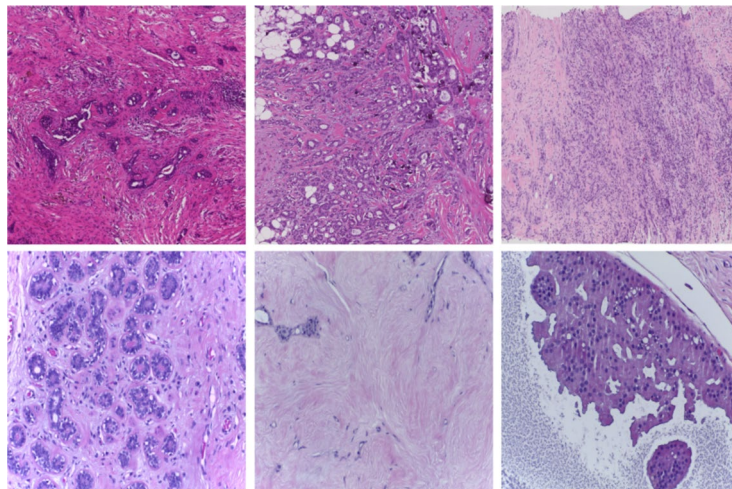
Histopathological images provide crucial information for understanding diseases and their effects at the cellular level<sup>1</sup>. Staining techniques significantly highlight and enhance the contrast of cellular and tissue characteristics under the microscope. However, this process may introduce certain disadvantages, leading to variations in stain texture. For instance, Fig. 1 reveals color inconsistencies within the same batch of sample data. Although pathology experts can interpret such fluctuations in color, these variations pose significant challenges for performance in digital image analysis conducted using machine learning algorithms.

In histopathological image datasets, the target domain consists of stain images with a consistent style, while collections lacking this uniform style form the source domain. The goal of stain normalization is to align the color distribution of source domain images with that of the target domain to mitigate the adverse effects of staining heterogeneity. This technique has been extensively applied in the preprocessing stage of image analysis<sup>2–5</sup>. However, the normalization process in these methods adopts conventional techniques by adjusting image colors to fit a specific template<sup>6,7</sup>. Despite these methods being based on mathematical models, there's still a risk of significant errors if the chosen template is not representative enough.

Recently, Generative Adversarial Networks (GAN)<sup>8</sup> have received extensive research attention for their application in stain normalization. DCGAN<sup>9</sup> was the first to employ a Convolutional Neural Network (CNN) architecture to extend GAN, allowing for the stable training of higher resolution and deeper generative models. Since then, every successful GAN in the field of computer vision has relied on CNN-based generators and

<sup>1</sup>School of Medical Technology and Engineering, Henan University of Science and Technology, Luoyang, China.

<sup>2</sup>Henan Engineering Research Center of Digital Pathology and Artificial Intelligence Diagnosis, Luoyang, China. <sup>3</sup>The First Affiliated Hospital of Henan University of Science and Technology, Luoyang, China. ✉email: liudechun2018@126.com



**Fig. 1.** Patch samples from the TUPAC-2016 (row one) and ICIAR-BACH-2018 (row two) datasets.

discriminators. Among these methods, those based on Cycle-consistent Generative Adversarial Networks (CycleGAN)<sup>10</sup> have been the most extensively researched. These methods can perform stain normalization without any template images and often achieve good results<sup>11</sup>. However, convolutional operators have a local receptive field that captures spatial local relationships in image data and extracts certain patterned features. To handle long-distance dependencies, CNNs need to go through a sufficient number of layers, which is inefficient and may lead to a loss of feature resolution and fine details. Consequently, ordinary CNN-based models are inherently less suitable for capturing the “global” statistics of input images. The effectiveness of self-attention<sup>12</sup> and non-local<sup>13</sup> operations in computer vision has proven this point. Recently, inspired by the huge success of Transformer<sup>14</sup> in the field of Natural Language Processing (NLP), researchers have sought to introduce Transformer into machine vision<sup>15</sup>. The success of Vision Transformer (ViT)<sup>16</sup>, Data-efficient image Transformer (DeiT)<sup>17</sup>, and Swin Transformer<sup>18</sup> in image recognition tasks demonstrates the potential application of Transformer in the visual domain. Models using Swin Transformer as the vision backbone have achieved state-of-the-art performance in image classification, object detection, and semantic segmentation. In light of this, combining the characteristics of CNN and Transformer, we use the Swin Transformer to construct our GAN’s generator, aiming to better preserve the consistency of structures and richness of texture details in histopathological images after stain normalization; while employing CNN to build the network’s discriminator, to endow the GAN with local perception capabilities.

On the other hand, GAN faces the challenge of significant differences between the source and target domains in the task of stain normalization. In one study, stain normalization was treated as an image colorization task, proving to have improved performance over methods based on CycleGAN<sup>19</sup>. However, such approaches require the output of image colorization as ground truth for supervised learning and, hence can only be trained on target domain images. This setup does not directly represent the goal of stain normalization, which is to normalize the color of source domain images to that of target domain images.

To benefit from supervised image colorization learning and incorporate source domain images into the colorization learning process, we propose a Deeply Supervised Two-stage Generative Adversarial Network (DSTGAN) for stain normalization. Inspired by the Deep Supervision (DS)<sup>20</sup> concept, we introduce deep supervision into GANs to enable the model to learn hierarchical representations from multi-scale aggregated feature maps<sup>21</sup>, enhancing the model’s learning capacity. Additionally, to fully leverage the image data from the source domain, we draw upon a semi-supervised learning framework<sup>22</sup> and design a Two-stage Staining (TS) strategy to further improve the model’s performance. We apply semi-supervised learning to the GAN, utilizing a two-stage staining approach in the source domain, where the result of the first colorization step serves as the ground truth for the second colorization, allowing the use of source domain images to enhance the learning of the colorization model without the need for paired ground truth images. In this paper, we focus on the application of DS and proxy-labeling<sup>23–26</sup> from semi-supervised learning in GANs, to fully train the model utilizing source domain images. Furthermore, inspired by the successful application of Transformer in GANs<sup>27–29</sup>, we enhance our generator based on Swin-Unet<sup>30</sup> to enable the model to establish long-range dependencies, applying it as a generator in GANs for the stain normalization task to better preserve the structural information of the images.

In this paper, we propose a novel generative adversarial network for stain normalization to address the issue of low-quality image generation present in current stain normalization methods. Specifically, our contributions are summarized as follows:

- We explore incorporating the Swin Transformer into the task of stain normalization to enhance the model’s ability to establish long-range dependencies, thereby preserving more structural and textural information in the images.
- Our model integrates semi-supervised concepts and is designed with an innovative two-stage staining strategy to fully leverage source domain images for training the model.

- To enhance the learning capability and stability of the stain normalization model, we innovatively introduce deep supervision into generative adversarial networks. To the best of our knowledge, this work might be the first attempt to combine adversarial generative networks with deep supervision for the task of stain normalization.
- Extensive experiments conducted on the TUPAC-2016, MITOS-ATYPIA-14, ICIAR-BACH-2018, and MIC-CAI-16-GlaS datasets demonstrate that our proposed model can effectively handle histopathological datasets with different statistical characteristics (i.e., different staining appearances from various pathology centers) and outperforms the state-of-the-art techniques.

## Related works

Stain normalization methods are typically divided into two categories: traditional approaches and deep learning-based approaches. Specifically, conventional stain normalization image processing methods heavily rely on domain experts' knowledge when selecting reference templates, whereas the development of deep neural networks and generative models has offered new avenues for stain normalization.

Traditional methods employ a mathematical framework to match image features with carefully selected template images. The Reinhard method<sup>7</sup> performs a set of linear transformations in the Lab color space and proposes a stain normalization approach that matches the mean and standard deviation of each channel of the image with those of the target. However, this method does not fully preserve the brightness of the background in the source image, which may ultimately reduce the contrast of the source image. Roy et al.<sup>31</sup> presented an improved Reinhard method that can preserve all the color variations in the source image (including pinkish spots). The Adaptive Color Deconvolution (ACD)<sup>32</sup> algorithm takes into account multiple priors of staining and can estimate the parameters for both color separation and normalization. It involves only pixel-level operations during the solving of the ACD model and its application process, which is efficient and suitable for the color separation and normalization of WSIs. The stain color adaptive normalization<sup>33</sup> based on segmentation and clustering strategies for cell structure detection can automatically perform color separation and normalization for hematoxylin and eosin-stained histological slides. It can improve the contrast between tissues and the background without changing the color of lumens or the background, preserving local structures. The Macenko method<sup>6</sup> uses a supervised pixel-level staining separation approach, requiring the addition of prior information in the training set. In contrast, the more advanced global stain normalization Vahadane method<sup>34</sup> does not guarantee the preservation of all the color information in the source image.

The deep learning-based CycleGAN can address the aforementioned issues, but CycleGAN tends to obscure many features in stained tissues in real-world settings, and its potential for model generalization is not ideal. Therefore, Jose et al.<sup>45</sup> summarized a new type of GAN method that retains the content of the source image while modifying the color style of the input image based on image style. Color transfer is considered more effective than traditional stain normalization methods<sup>46</sup>. It involves transferring the stain appearance of tissue images across different datasets to avoid color variations due to batch effects. Meanwhile, GAN represents a completely different strategy for stain normalization, and its effectiveness significantly surpasses that of classical stain normalization methods, making it the method of choice in recent years. Inspired by CycleGAN in an unsupervised setting, StainGAN<sup>11</sup> transfers the stain appearance, achieving high visual similarity with the target domain. Kang et al. proposed StainNet<sup>36</sup>, a method over 40 times faster than StainGAN, capable of normalizing images of a 100,000 × 100,000 slide within 40 min, changing the color appearance of the source image's staining matrix according to the color appearance of the reference image while preserving the structure of the illumination. Some supervised stain normalization methods<sup>19,47–50</sup> are purely trained for target domain images. They normalize or color back certain transformation representations of target domain images (e.g. grayscale space) to their original staining appearance (e.g. RGB space). Although these methods can produce high-quality standardized results, their performance is limited by the limited number of target domain images available in the dataset. Moreover, the target domain coloring formulas do not completely resemble the goal of stain normalization between the source and target domains. To include source domain images in the model training, Colour Adaptive Generative Networks for stain normalisation of histopathology images (CAGAN)<sup>42</sup> designed a dual-decoder structure. This approach leverages the concept of consistency regularization, enhancing the learning of the coloring model with source domain images and achieving better performance, though it relies on a substantial amount of data support. Meanwhile, the Self-Attentive Adversarial Stain Normalization (SAASN)<sup>38</sup> method normalizes the appearance of multiple stains to a common domain. Some researchers have proposed a method called MultiStain-CycleGAN<sup>39</sup>, which enables stain normalization for images from multiple different sources without the need for retraining or using different models. Additionally, other studies have applied GAN methods to fields such as digital pathology and dermatology (e.g., color constancy), framing the stain normalization task as an image-to-image translation problem to ensure pixel-to-pixel correspondence between the original and normalized images<sup>51</sup>. This unsupervised generative adversarial approach includes a self-attention mechanism, allowing for image synthesis with more detailed finesse. However, it falls short in the color consistency of the stain-normalized images. Some innovative approaches have also introduced a novel model named StainSWIN<sup>27</sup>, which leverages Swin Transformer<sup>18</sup> modules to capture long-range dependencies, achieving relatively advanced performance. Table 1 presents the key findings, training datasets, and evaluation metrics of various stain normalization methods.

Although current deep learning-based methods perform excellently in color normalization, they can easily cause detail information loss in complex tissue structure images. This might be unacceptable for pathological diagnosis, as subtle structural changes often play a critical role in disease diagnosis. Moreover, existing stain normalization methods often rely heavily on a large volume of high-quality data, posing a significant challenge to the model's learning capability when faced with small-volume datasets, which to some extent limits their practical applications. On the other hand, current GAN models based on CNNs mainly rely on local features,

Method	Key findings	Training dataset	Evaluation metrics
Macenko <sup>6</sup>	Introduced a histogram matching method for stain normalization; focuses on matching color distributions.	Not mentioned	Visual similarity
Reinhard <sup>7</sup>	Uses statistical matching in the Lab color space; effective for basic stain normalization tasks.	Not mentioned	Visual similarity
Vahadane <sup>34</sup>	Based on Non-Negative Matrix Factorization (NMF); provides robust stain decomposition and normalization.	MITOS-ATYPIA-14 dataset <sup>35</sup>	Pearson correlation and Quaternion Structural Similarity Index
StainGAN <sup>11</sup>	Applies GANs for unsupervised stain normalization; performs well on complex datasets.	MITOS-ATYPIA-14 dataset <sup>35</sup>	Structural Similarity Index, Feature Similarity Index for Image Quality Assessment, Peak Signal-to-Noise Ratio and Pearson correlation coefficient similarity
StainNet <sup>36</sup>	Uses deep learning to automate stain normalization without prior stain separation.	Camelyon16 dataset <sup>37</sup>	Structural Similarity Index and Peak Signal-to-Noise Ratio
SAASN <sup>38</sup>	Employs a self-attention mechanism to enhance stain normalization.	Duodenal biopsy dataset and MITOS-ATYPIA-14 dataset <sup>35</sup>	Structural Similarity Index
MultiStain-CycleGAN <sup>39</sup>	Handles stain normalization across multiple stain domains without retraining; versatile for diverse datasets.	Camelyon17 dataset <sup>40</sup> and SCC dataset <sup>41</sup>	Structural Similarity Index and Fréchet Inception Distance
CAGAN <sup>42</sup>	Incorporates adversarial learning to improve stain normalization consistency and quality.	TCGA-IDH dataset <sup>43</sup> , BreakHis dataset <sup>44</sup> , Camelyon <sup>10</sup> dataset <sup>37</sup> and Camelyon <sup>39</sup> dataset <sup>40</sup>	Structural Similarity Index, Peak Signal-to-Noise Ratio and Normalized Median Intensity
StainSWIN <sup>27</sup>	Leverages Swin Transformer for capturing long-range dependencies in stain normalization tasks.	MITOS-ATYPIA-14 dataset <sup>35</sup>	Structural Similarity Index, Peak Signal-to-Noise Ratio and Root Mean Square Error

**Table 1.** Some key findings of the stain normalization method, the dataset used for training, and evaluation metrics.

which could potentially limit their ability to capture global characteristics of images and address global color shifts. This could, in turn, impact their generalization capabilities and the quality of the stain-normalized images.

Methodology

In this section, we first introduce how to utilize target domain images for supervised learning approaches to stain normalization. Then, we describe the semi-supervised TS strategy learned on source domain images to fully utilize the dataset images for training the model. Additionally, we incorporate the idea of DS into our model, the overall architecture of which is illustrated in Fig. 2.

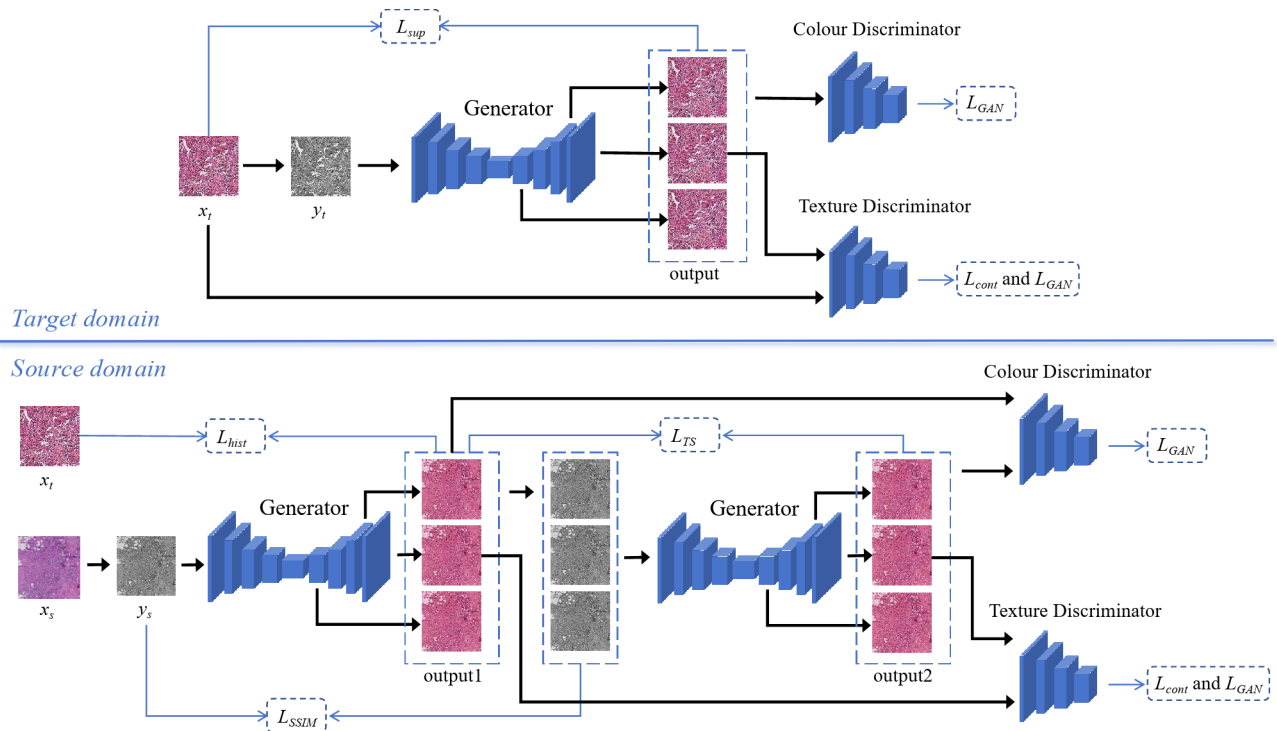
Problem formulation

For a given histopathological image dataset  $I$ , we define the target domain with relatively uniform staining colors as a subset  $I_p$  with the remaining images as the source domain, denoted as  $I_s$ . The goal of stain normalization is to reduce the color variations among images in dataset  $I$  without altering the original texture and structure of the images so that all images in  $I_s$  have the staining appearance of  $I_p$ . In this work, we construct our backbone network following the framework of pix2pix GAN<sup>52</sup>, which requires paired image data for training. Ideally, these image pairs consist of images of the same sample with different staining colors. However, obtaining such paired images is challenging in real-world datasets. Therefore, we follow prior work and use grayscale transformed images<sup>38,42</sup> and their corresponding RGB images as the paired image data. Specifically, given an image  $i \in I$ , its grayscale transformed image is represented as  $x_p$ , and then we train the model  $G_\theta$  to colorize  $x_p$  with the stain colors of  $I_p$ . Considering that converting the RGB image  $i$  to the grayscale image  $x_p$  might lead to information loss, we apply content loss<sup>53</sup> and structure similarity index measure (SSIM)<sup>54</sup> loss to directly regularize the model. To incorporate source domain images into the model's training, we have designed a semi-supervised TS strategy. Moreover, we introduce the idea of DS into the stain normalization model to learn hierarchical representations from multi-scale aggregated feature maps, thereby further enhancing the model's performance.

Network architecture

Our proposed stain normalization model consists of one generator and two discriminators. The structure of the generator is shown in Fig. 3. Here, we draw inspiration from the Swin-Unet<sup>30</sup> architecture to build our generator, which is a successful fusion of U-Net<sup>55</sup> and Swin Transformer<sup>18</sup>. Unlike Swin-Unet, in the downsampling stage, we do not use the Patch Partition module but only use the Patch Merging module for downsampling until the image size is reduced to one thirty-second of its original size. During the network's upsampling stage, we continue to use Skip Connection to reduce the loss of image information and employ Patch Expanding for 2x upsampling, adding Skip Connection from the input at the output to retain more of the original texture information. To be more effectively applied to the task of stain normalization, we add a convolutional layer at the output and use the hyperbolic tangent activation function. To implement DS, we use Patch Expanding 4x and Patch Expanding 16x modules to perform 4x and 16x upsampling of feature maps, respectively, to restore the size to that of the original image. As for the discriminators, one is a color discriminator and the other is a texture discriminator. The color discriminator focuses only on whether the color of the image comes from the target domain during training; here, we use a 5-layer PatchGAN<sup>52</sup> as our color discriminator. For the texture





**Fig. 2.** Overall structure of the model. The generator uses as input the gray-scale transformations  $y_{t/s}$  of the images in the target and source domains as input. Since we introduced deep supervision, the generator produces three normalized results. For the color discriminator, all output results from the generator need to be discriminated. For the texture discriminator, however, only the final output image from the generator is evaluated, and its deep texture difference  $L_{cont}$  is calculated. Specifically, we employ a two-stage staining strategy in the source domain, where the secondary generator processes the three outputs from the primary generator one by one, further enabling the generator to capture the texture details of the image.

discriminator, we construct it with several convolutional layers and a fully connected layer, focusing only on whether the texture of the image is consistent with the original during training.

### Supervised learning with target domain images

For target domain images  $I_p$ , we train our model using supervised learning. For convenience, let us denote any target domain image  $i_t$  ( $i_t \in I_p$ ) in RGB format as  $x_t$  with its grayscale transformation denoted as  $y_t$ . In this context, we describe the task of stain normalization as coloring  $y_t$  to  $x_t$  using a GAN model with supervised learning. Specifically, the task of coloring the input image with the desired stain colors is accomplished by the generator  $G$ , while the discriminator  $D$  evaluates whether the color of the colored image originates from the target domain color distribution and whether its texture is consistent with the input image. Therefore, our model includes one generator and two discriminators, named color discriminator  $D_C$  and texture discriminator  $D_T$  according to their functions. At this stage, we take  $y_t$  as the input to generator  $G$  and, adopting DS, train  $G$  to learn the color mapping from grayscale image  $y_t$  to target domain image  $x_t$ , that is,  $G(y_t)_i = \hat{x}_{ti}$ , where  $\hat{x}_{ti} \approx x_t$ . During this process, our color discriminator  $D_C$  follows the traditional GAN philosophy, which means using a single input, assigns higher values to actual RGB images  $x_t$  from the target domain, and lower values to the coloring results of  $G$ . Meanwhile, texture discriminator  $D_T$  focuses solely on the texture information of the images generated by  $G$ . Conversely,  $G$  attempts to deceive  $D_C$  and  $D_T$  by generating images more similar to  $x_t$  in both color and texture.

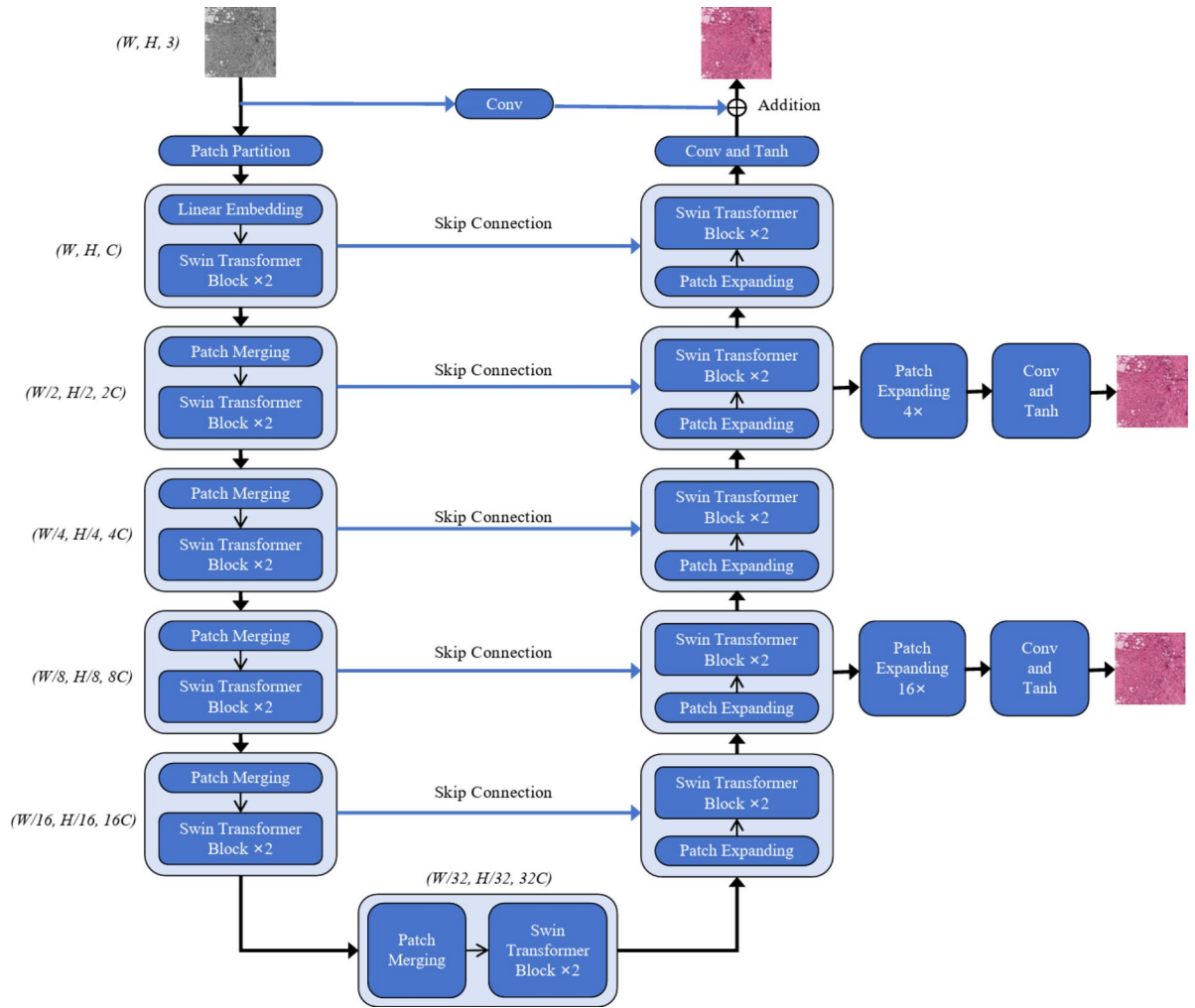
### Regularisation

Since the input image  $x_t$  can serve as the label during training in the target domain, we train the stain normalization model here using a supervised approach. The shape of the image is denoted as  $H \times W \times C$ . As the model employs DS, the supervised loss  $L_{sup}$  uses the sum of the  $L1$  losses for each colored image to regularize the model:

$$L_{sup} = \sum_i^m \omega_i \frac{1}{HWC} |\hat{x}_{ti} - x_t| \quad (1)$$

Where  $m$  is the total number of images predicted by the generator,  $\hat{x}_{ti}$  refers to the  $i_{th}$  image after coloring, and  $\omega$  is the weight assigned to each supervision.

We use the adversarial loss to evaluate the stained images, and for the adversarial loss  $L_{GAN}$  against different discriminators  $D_C$  and  $D_T$  we denote them respectively as  $L_{GAN}^C$  and  $L_{GAN}^T$ :



**Fig. 3.** The generator structure used in our model. The decoder section was augmented with 4x and 16x upsampling structures to support deep supervision. Where Conv denotes convolution and Tanh is the hyperbolic tangent activation function.

$$L_{GAN}^C = \sum_i^m \omega_i \{ E_{x_t \sim p_{data}(x_t)} [\log D_C(x_t)] + E_{y_t \sim p_{data}(y_t)} [\log(1 - D_C(G(y_t)_i))] \} \quad (2)$$

$$L_{GAN}^T = E_{x_s \sim p_{data}(x_s)} [\log D_T(x_s)] + E_{x_t \sim p_{data}(x_t)} [\log D_T(x_t)] + E_{y_t \sim p_{data}(y_t)} [\log(1 - D_T(G(y_t)))] \quad (3)$$

Here,  $L_{GAN}^C$  utilizes DS where  $G(y_t)_i = \hat{x}_{ti}$ , while  $L_{GAN}^T$  using only the output image from the last layer of generator  $G$  to compute the loss. To enable the color discriminator  $D_C$  to learn the color distribution of the target domain, we treat the target domain image  $x_t$  as a positive example during training, while the network-generated image  $\hat{x}$  is treated as a negative example. As for the texture discriminator  $D_T$ , both the target domain image  $x_t$  and the source domain image  $x_s$  are considered positive examples, while the network-generated image  $\hat{x}$  is treated as a negative example.

To retain richer structural features of the original image, we introduce the content loss<sup>53</sup>  $L_{cont}$  to regularize our model. Specifically, we use the discriminator  $D_T$ , which learns textural features, as a feature extractor to separately extract  $n$  deep features from the input image  $x_t$  and the predicted image  $\hat{x}_t$ . Since we have extracted the same number of image pairs of the same size, we can calculate the distance between each pair of images to serve as the content difference between them:

$$L_{cont}(\hat{x}_t, x_t) = \sum_j^n \lambda_j \frac{1}{C_j H_j W_j} \|\psi_j(\hat{x}_t) - \psi_j(x_t)\| \quad (4)$$

Where  $\psi_j$  represents the  $j_{th}$  feature map extracted by discriminator  $D_T$ ,  $C_j H_j W_j$  is the shape of the corresponding feature map, and  $\lambda$  represents the weight assigned to the feature map.

In summary, the loss function for supervised learning using the target domain images is defined as:

$$L_{target} = L_{sup} + L_{cont} + \alpha(L_{GAN}^C + L_{GAN}^T) \quad (5)$$

Here,  $\alpha$  is the weighting factor that controls the adversarial loss.

### Two-stage staining strategy with source domain images

To fully leverage the dataset, we also incorporate images from the source domain into the model's training. However, this is not a straightforward task. When training with target domain images, it's possible to use the target domain images as labels, allowing for supervised learning. However the color appearance of the source domain images differs significantly from that of the target domain, and the output of the model after coloration of these images is unpredictable. This makes the strategy of supervised learning less effective when training the model with source domain images. Therefore, inspired by the idea of proxy-labelling in semi-supervised learning, we designed a TS (Two-stage Staining) strategy. This strategy utilizes proxy-labeling generated by the first-stage coloring of the generator as the label for the second-stage supervised learning, to achieve the purpose of training the model using source domain images. During the first stage of image coloration, due to the color difference between the input and output images, we employ the grayscale images of both and combine them with SSIM loss to regularize our model. This ensures that the proxy-labeling provided for the second stage coloring has richer textural information. Additionally, with the introduction of DS in our model, the generator is capable of producing multiple coloring results, providing the second stage of model training for stain normalization with abundant and higher-quality proxy labeling. This allows the model to further enhance its performance by utilizing extensive training on source domain images.

#### First stage regularization

Due to the lack of one-to-one corresponding labels when training the model with source domain images, to preserve as much of the original texture information as possible during the first-stage coloring, we regularize our model by calculating the structural similarity of the input and output grayscale images:

$$SSIM(x, \hat{x}) = \frac{[2\mu(x)\mu(\hat{x}) + C_1][2\sigma(x, \hat{x}) + C_2]}{[\mu(x)^2 + \mu(\hat{x})^2 + C_1][\sigma(x)^2 + \sigma(\hat{x})^2 + C_2]} \quad (6)$$

$$L_{SSIM} = \sum_i^m \omega_i [1 - SSIM(y_s, \hat{y}_{si})] \quad (7)$$

Where  $\mu$  represents the average pixel value of the corresponding images,  $\sigma$  is the unbiased estimation of standard deviation.  $C_1$  and  $C_2$  are constants, which here are taken to be  $C_1=(0.01)^2$  and  $C_2=(0.03)^2$  respectively.  $y_s$  and  $\hat{y}_s$  represent the grayscale images of the model's input and output during the first stage, respectively. Since the model employs DS,  $\hat{y}_{si}$  denotes the grayscale image of each colored output produced by the generator.

To ensure that the color distribution of the model's predicted images in the first stage aligns with that of the target domain images, we have added an additional regularization term, namely the histogram loss<sup>56</sup>. Specifically, we need to estimate the contribution of each pixel of the output image and the target domain image to histogram bins in the log-chromaticity space, to construct the corresponding histogram features  $H_s$  and  $H_t$ . This can be controlled by an inverse-quadratic kernel  $k$  for each pixel's contribution:

$$k(I_{uc}, I_{vc}, u, v) = \frac{1}{1 + \left(\frac{I_{uc}-u}{\tau}\right)^2} \times \frac{1}{1 + \left(\frac{I_{vc}-v}{\tau}\right)^2} \quad (8)$$

Where  $I_{uc}$  and  $I_{vc}$  represent the pixel intensities in the image log-chromaticity space, while  $u$  and  $v$  are hyperparameters, and  $\tau$  is a decay parameter that controls the smoothness of histogram bins. By calculating the corresponding histogram features  $H_s$  and  $H_t$  using the kernel function, we are then able to construct the histogram loss  $L_{hist}$ :

$$L_{hist} = \frac{1}{\sqrt{2}} \sum_i^m \omega_i \|\sqrt{H_t} - \sqrt{H_{si}}\|_2 \quad (9)$$

Where  $\|\cdot\|_2$  denotes the standard Euclidean norm, and  $H_{si}$  represents the histogram feature of each image output by the model in the first stage.

**Arguments:** Generator  $G$ , Discriminator  $D_C$  &  $D_T$

**Inputs:**  $m$  batches of target domain images pairs  $\{x_{ti}; i=1, \dots, m\}$ ,  $m$  batches of source domain images  $\{x_{si}; i=1, \dots, m\}$

**Result:** Trained models  $G$ ,  $D_C$ , and  $D_T$

**for number of epochs do**

**for number of training iterations do**

$y_{ti} \leftarrow \text{graying}(x_{ti});$

$y_{si} \leftarrow \text{graying}(x_{si});$

Forward to obtain normalization results:

$$\hat{x}_{ti} = G(y_{ti});$$

$$\hat{x}_{si} = G(y_{si});$$

Updating discriminator:

**for number of steps do**

$$\nabla \theta_{d1} \frac{1}{m} \sum_1^m L_{GAN}^C(D_C, G, x_{ti}, y_{ti});$$

$$\nabla \theta_{d2} \frac{1}{m} \sum_1^m L_{GAN}^T(D_T, G, x_{ti}, y_{ti});$$

**end**

Updating generator:

$$L_{target} = L_{sup}(x_{ti}, \hat{x}_{ti}) + L_{cont}(x_{ti}, \hat{x}_{ti}) + L_{GAN}(D_C, D_T, G, x_{ti}, y_{ti});$$

$$L_{source}^1 = L_{SSIM}(y_{si}, \hat{y}_{si}^1) + L_{hist}(x_{ti}, \hat{x}_{si}^1) + L_{GAN}(D_C, D_T, G, \hat{x}_{si}^1, y_{si});$$

$$L_{source}^2 = L_{TS}(\hat{x}_{si}^1, \hat{x}_{si}^2) + L_{cont}(\hat{x}_{si}^1, \hat{x}_{si}^2) + L_{GAN}(D_C, D_T, G, \hat{x}_{si}^2, \hat{y}_{si}^1);$$

$$\nabla \theta_g \frac{1}{m} \sum_1^m (L_{target} + L_{source}^1 + L_{source}^2);$$

**end**

**end**

#### Algorithm 1. Obtaining a trained model of two-stage stain normalization

Similarly, we use Eqs. (2) and (3) to calculate the GAN loss for the coloring performed by the model at this stage, denoted as  $L_{GAN}^1$ . The content loss continues to be calculated using Eq. (4), referred to as  $L_{cont}^1$ . Therefore, the loss function used for the first stage coloring with source domain images is:

$$L_{source}^1 = L_{SSIM} + L_{hist} + \alpha L_{GAN}^1 \quad (10)$$

#### Second stage regularization

To further enhance the model's performance, we add second-stage coloring training for source domain images. Specifically, we use the grayscale image  $\hat{y}_s^1$  of the model's first-stage colored image as proxy-labeling to employ supervised learning in the model's second-stage coloring. Here, we continue to use the L1 loss to construct our loss function for this stage,  $L_{TS}$ :

$$L_{TS} = \sum_j^m \sum_i^m \omega_i \frac{1}{HWC} |\hat{x}_{sij}^2 - \hat{x}_{si}^1| \quad (11)$$

Since the model uses DS in both stages,  $i$  and  $j$  here represent the indices of the output images from the first and second stages respectively.

In the second stage of coloring, we continue to utilize the aforementioned GAN loss and content loss to regularize our model. Thus, the loss function we use for the second stage coloring with source domain images is:

$$L_{source}^2 = (L_{TS} + L_{cont}^2 + \alpha L_{GAN}^2) e^{-\beta L_{source}^1} \quad (12)$$



Here, to enable the model to adaptively learn based on the quality of proxy labeling, we have constructed an annealing function  $e^{-\beta L_{source}^1}$  using the loss from the first stage of coloring to avoid poor local minima and enhance the stability of model training.

In summary, the loss function we use for regularizing the model with source domain images is defined as follows:

$$L_{source} = L_{source}^1 + L_{source}^2 \quad (13)$$

The entire training process of the model is as shown in Algorithm 1.

## Experiments

### Dataset description

The tumor proliferation assessment challenge (TUPAC-2016) challenge dataset<sup>57</sup> includes images of 73 breast cancer patients from three different pathology centers. These images were generated using a Leica SCN400 scanner at a magnification of 40x, with a spatial resolution of 0.25  $\mu\text{m}/\text{pixel}$ . Here, we select the image of the first patient in the training set as the target domain, with the remaining images serving as the source domain. The test set within the dataset is used to evaluate the model's performance.

The mitosis & atypia 14 (MITOS-ATYPIA-14) challenge dataset<sup>35</sup> contains images at three different magnifications: 10X, 20X, and 40X. These images are scanned by two different scanners, the Aperio Scanscope XT and the Hamamatsu Nanozoomer 2.0-HT. In this paper, we use images scanned by the Aperio Scanscope XT from the training set as the source domain, while images scanned by the other scanner serve as the target domain, and we evaluate the model using the test set.

The ICIAR 2018 breast cancer histology (ICIAR-BACH-2018) grand challenge dataset<sup>58</sup> is scanned by a LeicaDM 2000 LED microscope, with a spatial resolution of 0.42  $\mu\text{m}/\text{pixel}$ . We select images with relatively balanced staining from the training set as the target domain, with the remaining images serving as the source domain. We continue to use the test set of this dataset to evaluate the model's performance. Additionally, we use this dataset to verify the performance of the stain normalization model in downstream classification tasks.

The MICCAI'16 gland segmentation (MICCAI-16-GlaS) challenge dataset<sup>59</sup> is scanned with a Zeiss MIRAX MIDI scanner at a 20X magnification. It includes 85 training images and 80 test images, where the test images are divided into a test part A with 60 images and a test part B with 20 images. All pathology images come with corresponding gland segmentation masks. In this paper, we combine the training set with test part A to form a new training set. For the stain normalization experiment, we select images with relatively uniform staining from the training set as the target domain, and the rest of the images as the source domain, while test part B is used to evaluate the model's performance. To further assess the model's impact on downstream tasks, we conducted gland segmentation experiments using this dataset.

### Experiment setup

To evaluate the performance of the stain normalization model, we set up two sets of experiments: the quality assessment of stain-normalized images and the impact of stain normalization on downstream tasks.

#### *Quality assessment of stain-normalized images*

To evaluate the quality of images output by the model, we trained stain normalization models separately on the training sets of TUPAC-2016, MITOS-ATYPIA-14, ICIAR-BACH-2018, and MICCAI-16-GlaS, and then assessed the quality of stain-normalized images on the corresponding test sets. To confirm the model's generalizability, we also designed cross-domain experiments, i.e., training on one dataset and testing on another. Specifically, we used the model trained on the MITOS-ATYPIA-14 dataset to test the quality of stain-normalized images on the TUPAC-2016, ICIAR-BACH-2018, and MICCAI-16-GlaS datasets. Here, we selected the structure similarity index measure (SSIM)<sup>54</sup>, Pearson correlation coefficient (PCC), and peak signal-to-noise ratio (PSNR)<sup>60</sup> metrics to evaluate the quality of stain-normalized images. Since images have different color appearances before and after staining, we use images transformed into grayscale for quality evaluation.

#### *The impact of stain normalization on downstream tasks*

In this experiment, we separately evaluated the effect of stain normalization on classification performance and segmentation performance. Here, we uniformly performed stain normalization on images using a model trained on the MITOS-ATYPIA-14 dataset. For evaluating classification performance, we first applied stain normalization to all images in the ICIAR-BACH-2018 dataset, then trained a ResNet50<sup>61</sup> classifier on its training set, and finally assessed classification performance on its test set. As evaluation metrics, we used accuracy, precision, and F1-score to measure the final impact of stain normalization on classification performance. For the assessment of segmentation performance, we first stain-normalized all images in the MICCAI-16-GlaS dataset and then trained a U-Net<sup>55</sup> segmentation network on its training set, before evaluating segmentation performance on its test set. Here, we employed the dice similarity coefficient (Dice), intersection over union (IOU), and pixel accuracy (PA) as the evaluation metrics.

All experimental results are obtained using the PyTorch framework on an NVIDIA RTX 4090 GPU. In this paper, we resize all input images to a dimension of  $256 \times 256$ . To enhance the model's robustness, we applied the same combination of data augmentation techniques during training, including Gaussian blur, contrast adjustment, and saturation adjustment. This approach is consistent with the image preprocessing steps used

Method	TUPAC-2016		MITOS-ATYPIA-14		ICIAR-BACH-2018		MICCAI-16-GlaS	
	SSIM	PCC	SSIM	PCC	SSIM	PCC	SSIM	PCC
Macenko	0.524±0.001	0.773±0.001	0.769±0.003	0.891±0.002	0.866±0.001	0.951±0.002	0.798±0.007	0.944±0.003
Reinhard	0.492±0.001	0.752±0.001	0.772±0.001	0.885±0.001	0.871±0.001	0.967±0.001	0.936±0.002	0.980±0.002
Vahadane	0.508±0.001	0.750±0.001	0.793±0.001	0.909±0.001	0.905±0.001	0.973±0.001	0.951±0.001	0.983±0.001
StainGAN	0.833±0.013	0.912±0.020	0.844±0.012	0.924±0.006	0.923±0.009	0.961±0.005	0.897±0.034	0.964±0.013
SAASN	0.975±0.003	0.986±0.002	0.972±0.003	0.983±0.002	0.970±0.013	0.983±0.006	0.939±0.014	0.976±0.003
CAGAN	0.955±0.024	0.977±0.017	0.935±0.010	0.964±0.007	0.923±0.019	0.969±0.007	0.957±0.006	0.983±0.007
DSTGAN	<b>0.985±0.001</b>	<b>0.992±0.001</b>	<b>0.984±0.002</b>	<b>0.991±0.001</b>	<b>0.984±0.003</b>	<b>0.992±0.001</b>	<b>0.975±0.003</b>	<b>0.990±0.001</b>

**Table 2.** SSIM and PCC comparison with different methods on each dataset. Significant values are in bold.

Method	TUPAC-2016	MITOS-ATYPIA-14	ICIAR-BACH-2018	MICCAI-16-GlaS	Time (s)
Macenko	16.108±0.017	19.231±0.031	22.108±0.236	19.714±0.397	–
Reinhard	15.088±0.051	17.401±0.111	21.752±0.068	18.650±0.252	–
Vahadane	15.057±0.010	20.520±0.002	23.547±0.006	25.665±0.061	–
StainGAN	25.732±0.444	24.419±0.519	26.493±0.528	24.196±0.521	2.208
SAASN	36.399±1.178	29.053±1.874	33.484±2.201	26.899±1.117	6.300
CAGAN	33.264±1.159	30.531±0.679	32.723±0.935	27.915±0.447	<b>1.954</b>
DSTGAN	<b>39.130±0.844</b>	<b>37.628±2.635</b>	<b>38.399±1.750</b>	<b>34.147±0.892</b>	3.274

**Table 3.** PSNR comparison for different methods on each dataset, along with the time required for 10 training updates (in seconds) with a batch size of 2. Significant values are in bold.

in CAGAN. We use the same learning rate ( $lr=0.0002$ ) for both the generator and discriminator and train the model for 50 epochs with a batch size of 2 using the Adam optimizer. All experiments present the mean and standard deviation of 5 independent runs. Hyperparameters  $\omega, \lambda, \alpha, \beta$  are set to  $[0.1, 0.3, 0.6], [0.1, 0.2, 0.3, 0.4], 0.5, 4$ , respectively.

Quality assessment of stain-normalized images

Quantitative comparison

For a fair comparison, we employ the same settings as existing methods and select the SSIM, PCC, and PSNR metrics to quantitatively evaluate the quality of the stain-normalized images. For these three metrics, higher values indicate that the textural information of the stain-normalized images is better preserved compared to the original images, the signal-to-noise ratio is higher, and thus the image quality is also better. Here, we compare traditional stain normalization methods with those based on deep learning. Specifically, traditional stain normalization methods include Macenko<sup>6</sup>, Reinhard<sup>7</sup>, and Vahadane<sup>34</sup>, while deep learning-based stain normalization methods comprise StainGAN<sup>11</sup>, SAASN<sup>38</sup>, and CAGAN<sup>42</sup>.

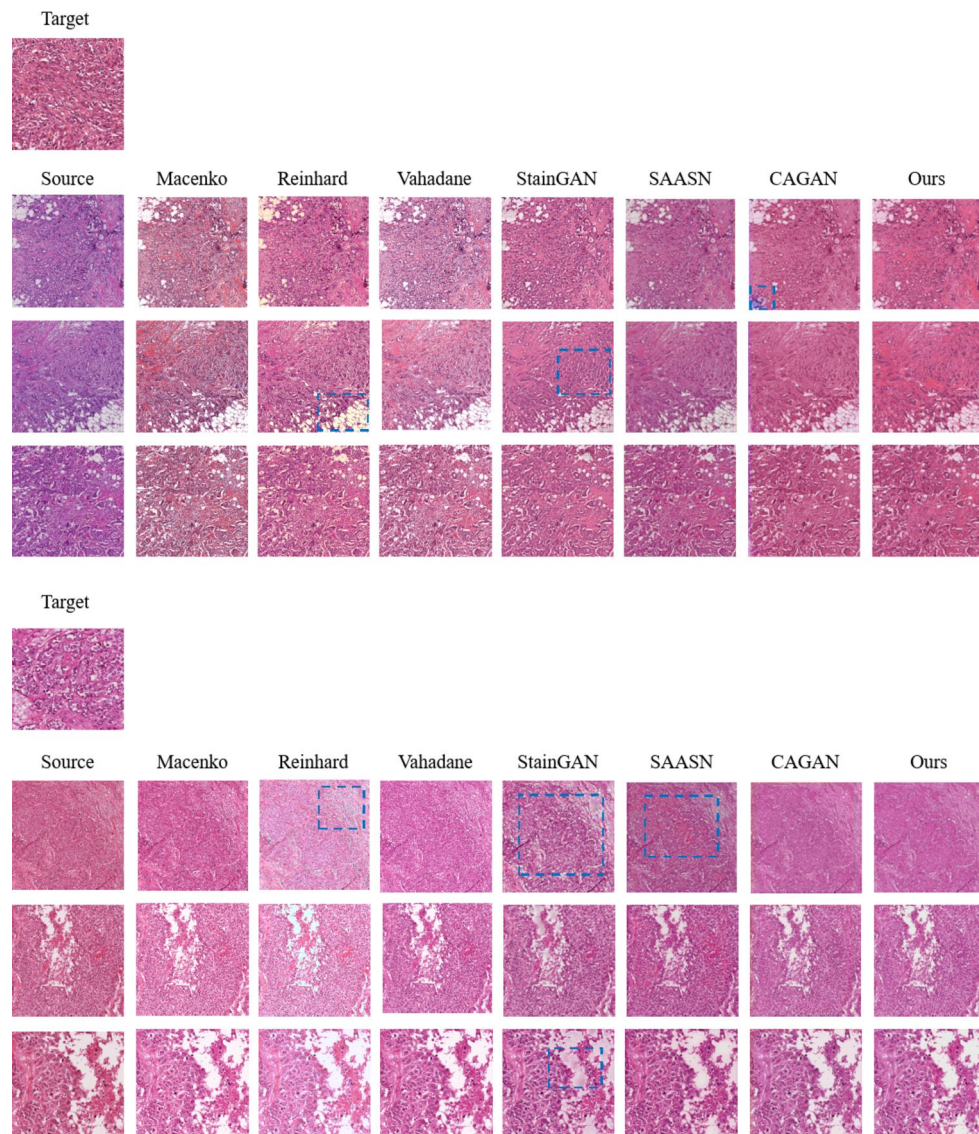
Quantitative comparisons of different methods across various datasets are shown in Tables 2 and 3. The SSIM metric considers three dimensions of an image: luminance, contrast, and structure; it focuses more on the visual effect and structural information of the image, aiming to simulate how the human eye evaluates image quality. As indicated in the tables, our proposed method achieves state-of-the-art performance in the SSIM criterion across all datasets, which is related to our utilization of SSIM loss on the grayscale images,  $L_{SSIM}$ . The PCC metric, on the other hand, measures the linear correlation between corresponding pixel values of images before and after stain normalization, seeking to characterize the consistency of the image’s statistical distribution. Our method also achieves the best performance on the PCC metric and has a value very close to that of the SSIM metric, suggesting that images after stain normalization retain higher structural integrity and linear consistency when compared to the original images. Moreover, compared to currently advanced methods, our model also shows a significant advantage in the PSNR metric, indicating less noise in the stain-normalized images. In summary, the quantitative results of the within-domain comparison suggest that our proposed method outperforms both traditional and deep learning-based state-of-the-art methods. Additionally, Table 3 reports the time required to update parameters 10 times during training for GAN-based stain normalization methods (with a batch size of 2) for readers’ reference.

In summary, our method achieved state-of-the-art performance in quantitative comparisons. Through analysis, it can be observed that methods based on linear models (e.g., Macenko, Reinhard, Vahadane) lack the ability to model nonlinear features effectively. While GAN-based methods introduce deep learning, they still face challenges in preserving high-frequency texture and structural information. StainGAN does not explicitly account for local texture details in high-resolution images, as its adversarial loss primarily focuses on global style matching rather than detail preservation. As a result, the generator may overlook high-frequency texture details, particularly when the GAN is undertrained or the dataset exhibits high diversity, leading to excessive texture smoothing. SAASN employs attention mechanisms to enhance local feature extraction, but it may overly focus on prominent regions while neglecting texture consistency in non-salient areas, resulting

in insufficient restoration of global texture information. CAGAN uses conditional GANs, which enhance flexibility in staining styles. However, this approach may introduce additional noise, especially in scenarios with imbalanced or insufficient training data. Under such conditions, the model tends to generate globally consistent styles at the expense of local details. Our proposed method introduces DS, which quickly learns the staining differences between two domains while retaining good texture information. Moreover, we utilized the concept of proxy-labeling in the source domain to design the TS strategy, making full use of the source domain dataset and to some extent overcoming the issue of small dataset sizes. Additionally, the generator built on the Swin Transformer architecture can capture global dependencies within images, enabling better preservation of detailed information in complex tissue structures.

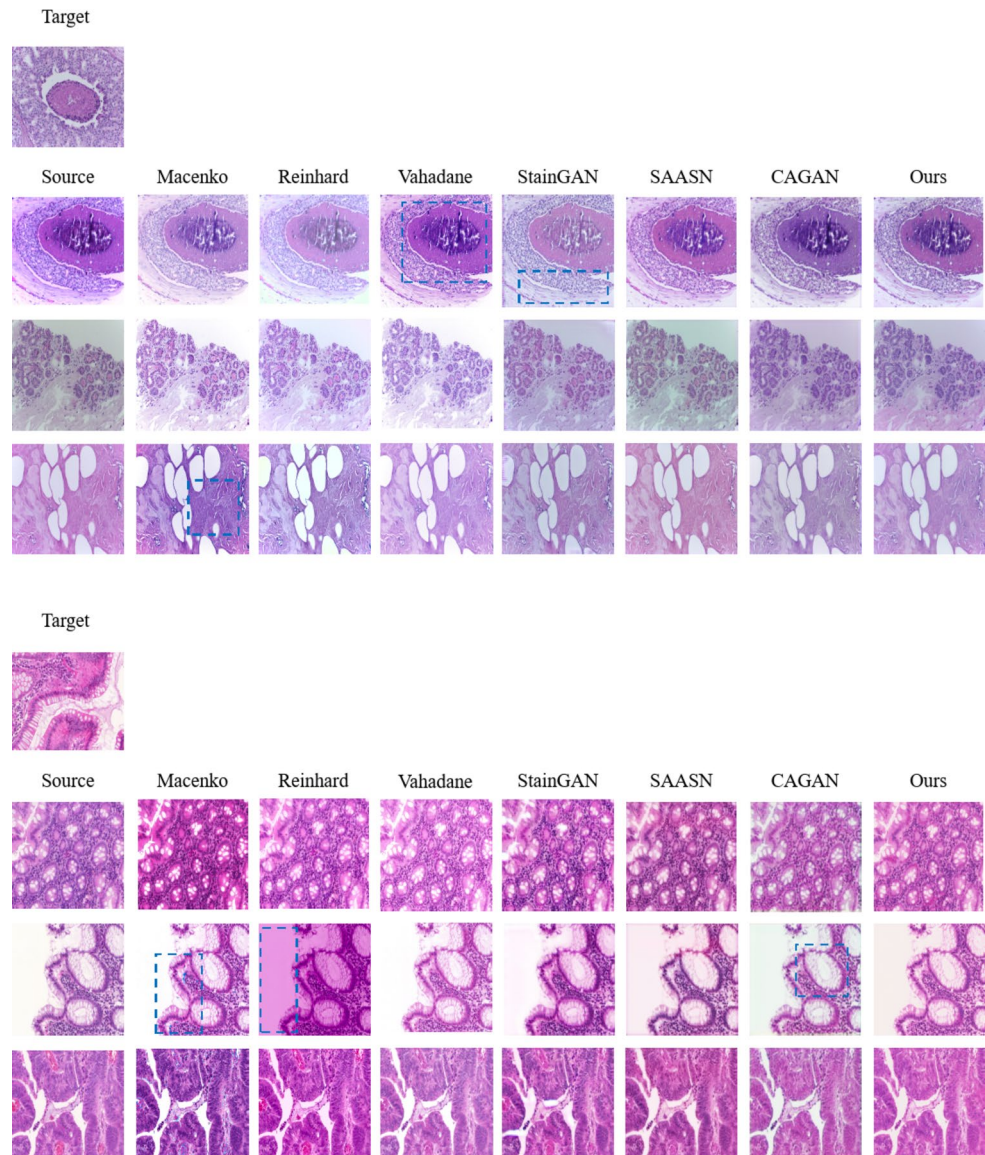
#### Qualitative comparison

Here, we still use the existing methods and datasets mentioned in the quantitative comparison for a qualitative comparison, with the visualization of images before and after staining shown in Figs. 4 and 5. From the results, it can be observed that traditional staining normalization methods, such as the Macenko method, can generate images with more consistent colors, but the stained images tend to produce artifacts and accompany more noise. In contrast, images after staining normalization by the Vahadane method have better quality, but when there is a significant color difference between the source and target domain images in the dataset, the color consistency of the stained images is poor. For methods based on deep learning, there is much more flexibility in staining normalization, and the staining is more uniform. However, the StainGAN method based on CycleGAN tends to produce artifacts in the stained images, and the quality of the generated images is lower, which might be a more general issue with the CycleGAN model. In comparison, SAASN can generate better quality images, but the



**Fig. 4.** Staining normalization results for different methods on the TUPAC-2016 dataset (row one) and the MITOS-ATYPIA-14 dataset (row two). Blue boxes indicate regions with problematic image texture or color.

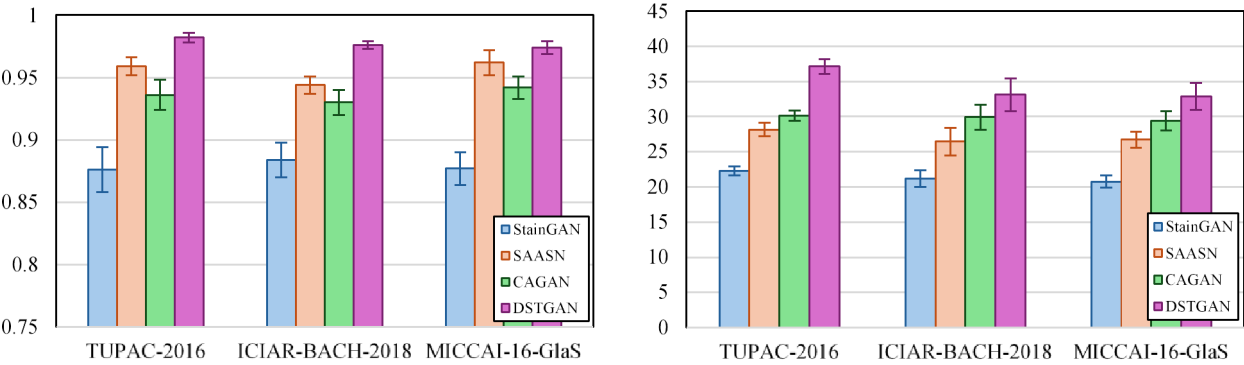




**Fig. 5.** Staining normalization results for different methods on the ICIAR-BACH-2018 dataset (row one) and the MICCAI-16-GlaS dataset (row two). Blue boxes indicate regions with problematic image texture or color.

stained-normalized images have a larger difference compared to the target domain. The more advanced CAGAN method introduces histogram loss, resulting in visually better staining consistency in the stained images but prone to edge artifacts, possibly due to the lack of constraints on image structure during training in the source domain. Since our method introduces structural similarity loss in the training within the source domain, it significantly retains structural information. Additionally, the design of the TS strategy in the training within the source domain further enhances the quality of the stained images. However, it is worth noting that GAN-based methods exhibit varying degrees of white calibration issues in the visual results on the MICCAI-16-GlaS dataset (as shown in the second row), while Non-Negative Matrix Factorization (NMF)-based methods, such as Vahadane, handle this problem more effectively. This discrepancy could be attributed to the fact that GANs rely on global context and the learning capability of the generator. When blank regions are not explicitly handled, the generated results may “overfit” these regions, leading to erroneous staining of white areas. On the other hand, NMF-based methods leverage matrix decomposition, allowing for more direct control over the weighting of different components, thereby exhibiting greater robustness in processing white and uniform areas in input images. However, this also highlights a limitation of NMF methods: they may perform suboptimally in complex staining patterns, especially when dealing with nonlinear staining distributions, where GANs demonstrate greater flexibility.

To verify the generalizability of our model, we also conducted cross-dataset experiments. As shown in Fig. 6, we trained our model on the MITOS-ATYPIA-14 dataset and then tested it on the remaining three datasets. Here, we use SSIM and PSNR metrics to evaluate the quality of stain-normalized images, and we only make comparisons with deep learning methods. As can be inferred from the graph, our method achieves the best



**Fig. 6.** Cross-domain comparison of SSIM (left) and PSNR (right) for different methods on each dataset (pretrained on MITOS-ATYPIA-14).

Method	Accuracy	Precision	F1-Score
Original	0.833±0.012	0.843±0.007	0.832±0.013
Macenko	0.853±0.022	0.861±0.024	0.851±0.022
Reinhard	0.835±0.017	0.841±0.015	0.833±0.019
Vahadane	0.858±0.012	0.863±0.013	0.857±0.013
StainGAN	0.860±0.013	0.864±0.012	0.858±0.013
SAASN	0.859±0.009	0.868±0.008	0.859±0.009
CAGAN	0.878±0.013	0.887±0.017	0.878±0.014
DSTGAN	<b>0.885±0.013</b>	<b>0.896±0.012</b>	<b>0.884±0.013</b>

**Table 4.** Comparison of classification performance of different methods on iciar-bach-2018 dataset. Significant values are in bold.

SSIM and PSNR results in all three datasets. This is perhaps due to our use of a generator capable of capturing global semantic relationships, and the application of image grayscale conversion as input. This reduces color interference from different input images, thereby enhancing the proposed method’s generalization performance.

**The impact of stain normalization on downstream tasks**

In this experiment, we used the ICIAR-BACH-2018 dataset to compare the impact of various stain normalization methods on classification performance. This dataset contains pathological images of breast tissue in four categories: normal, benign, in situ, and invasive. This presents a more challenging task than simple binary classification. Moreover, the significant differences in staining styles of the dataset images also pose a challenge to the generalization capability of stain normalization algorithms based on deep learning. We first trained models using different stain normalization methods on the MITOS-ATYPIA-14 dataset, then applied these models for stain normalization on the ICIAR-BACH-2018 dataset, and finally performed the classification task using the stained dataset. For the classifier, we uniformly employed the ResNet50 network to conduct the four-category classification of the dataset, utilizing the same training and testing processes. The classification results are shown in Table 4.

From the results, it is apparent that due to the significant differences in the colors of the dataset images, the classification results using the original data were not ideal. However, after applying different stain normalization methods to process the dataset, there was a varying degree of improvement in classification performance, which proves the effectiveness of stain normalization methods. Among all the results, our method achieved the best performance, demonstrating that our method significantly improves classification performance.

*Stain normalization in the segmentation task*

In this experiment, we performed a segmentation task on the MICCAI-16-GlaS dataset and used different stain normalization methods as a preprocessing step. Specifically, we trained various stain normalization models on the MITOS-ATYPIA-14 dataset and then applied stain normalization to the MICCAI-16- GlaS dataset. To ensure fairness, we trained the U-Net network for the same number of epochs using the stain-normalized dataset and tested the segmentation performance on the corresponding test set. The experimental results are shown in Table 5.

Since the model primarily relies on color and texture differences in the tissue to complete the segmentation of glandular structures, this places high demands on the quality and color richness of the post-staining images. As seen in the table, the segmentation results obtained using the Macenko method as a preprocessing step were poor, which could be due to excessive noise introduced during the stain normalization process. In comparison, deep learning methods based on GANs achieved better results. However, even though StainGAN can generate



Method	Dice	IOU	PA
Original	0.818±0.003	0.726±0.004	0.824±0.006
Macenko	0.811±0.003	0.717±0.004	0.815±0.002
Reinhard	0.826±0.003	0.733±0.004	0.831±0.002
Vahadane	0.827±0.005	0.732±0.009	0.832±0.005
StainGAN	0.844±0.004	0.750±0.007	0.847±0.003
SAASN	0.848±0.002	0.753±0.003	0.855±0.004
CAGAN	0.850±0.008	0.741±0.010	0.858±0.007
<b>DSTGAN</b>	<b>0.860±0.009</b>	<b>0.760±0.012</b>	<b>0.864±0.008</b>

**Table 5.** Comparison of segmentation performance of different methods on miccai-16-glas dataset. Significant values are in bold.

	TUPAC-2016			ICIAR-BACH-2018		
	SSIM	PCC	PSNR	SSIM	PCC	PSNR
Baseline	0.972±0.007	0.984±0.004	34.490±1.068	0.969±0.002	0.987±0.001	33.560±0.874
Add $L_{hist}$	0.965±0.007	0.980±0.004	32.902±1.896	0.976±0.001	0.990±0.001	34.194±0.849
Add $L_{SSIM}$	0.980±0.002	0.989±0.002	35.487±1.214	0.979±0.001	0.991±0.001	35.999±0.568
Add $L_{cont}$	0.979±0.002	0.989±0.001	35.972±0.685	0.979±0.002	0.991±0.001	36.303±0.525
Add DS	0.983±0.002	0.990±0.002	37.322±1.557	0.983±0.001	0.991±0.001	36.824±1.259
Add TS (DSTGAN)	<b>0.985±0.001</b>	<b>0.992±0.001</b>	<b>39.130±0.844</b>	<b>0.984±0.003</b>	<b>0.992±0.001</b>	<b>38.399±1.750</b>

**Table 6.** Base line models are added sequentially with different loss functions or strategies on the TUPAC-2016 and ICIAR-BACH-2018 datasets with stained normalized results. Significant values are in bold.

images with more uniform colors, the stained images are prone to artifacts and lower image quality, which therefore constrains the segmentation performance. Relatively speaking, the SAASN and CAGAN methods can produce higher-quality images. Since images stained with CAGAN exhibit better color consistency and a higher PSNR index in cross-dataset experiments compared to SAASN, they achieved better segmentation performance. The method we proposed not only generates images with a more extensive color distribution visually but also exhibits higher image quality after stain normalization, thus effectively enhancing image segmentation performance and achieving the best results.

### Ablation studies

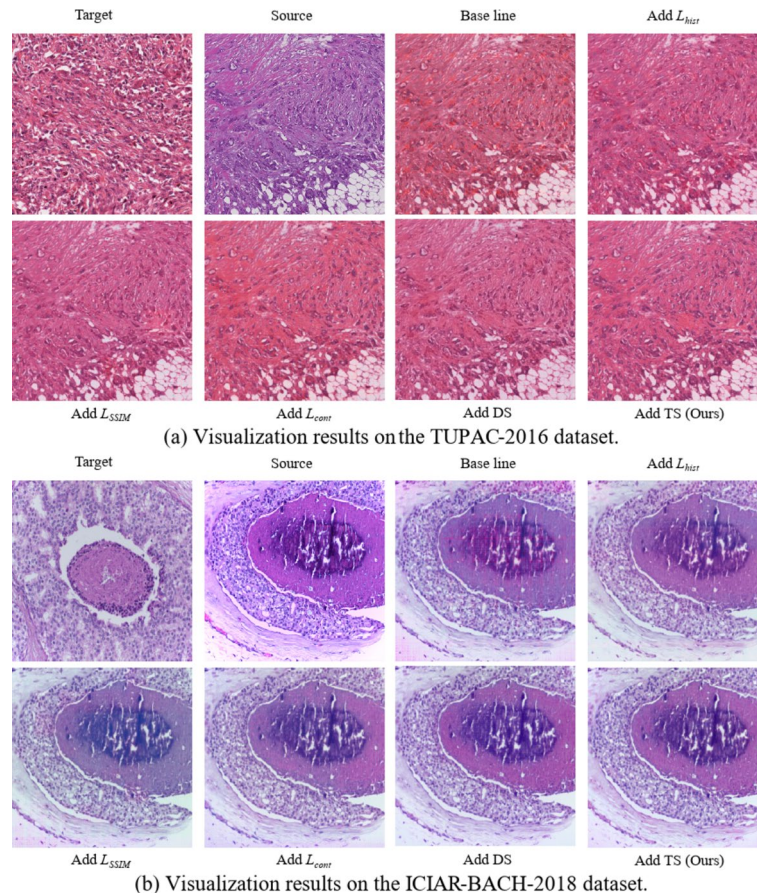
We also studied the effectiveness of different loss functions and strategies within DSTGAN. Specifically, we analyzed the impact of histogram loss  $L_{hist}$ , SSIM loss  $L_{SSIM}$ , content loss  $L_{cont}$ , the DS strategy, and the TS strategy on the quality of the stain-normalized images. Here, we utilized datasets with significant staining differences, TUPAC-2016 and ICIAR-BACH-2018, to perform both quantitative and qualitative analyses to verify the effectiveness of each component.

#### Quantitative comparison

Here, we continue to use SSIM, PCC, and PSNR metrics to measure the quality of the stain-normalized images. The experimental results are shown in Table 6. As can be seen from the table, due to the use of a generator built with the Swin Transformer, even if the model only uses the supervision loss from the target domain (Baseline model), it can still generate high-quality stain-normalized images. However, adding histogram loss  $L_{hist}$  in the TUPAC-2016 dataset led to a decrease in metrics, possibly because the staining style differences in that dataset are too large. To reduce the histogram distance between the target and source domain images, some areas were incorrectly stained, resulting in lower metrics. The inclusion of the SSIM loss  $L_{SSIM}$  significantly improved the metrics (especially the SSIM metric), highlighting its importance in enhancing image quality. From a quantitative perspective, the content loss  $L_{cont}$  improved the signal-to-noise ratio of the images. The introduction of DS and TS strategies further amplified the efficacy of each component, enhancing the overall performance of the model.

#### Qualitative comparison

In this experiment, we demonstrate the stain normalization results on the TUPAC-2016 and ICIAR-BACH-2018 datasets by sequentially adding different components, as shown in Fig. 7. From the qualitative results, we can observe that although the Baseline model without any components can achieve relatively good quantitative results, the color deviation in the images generated by the model is large. This was improved after the addition of histogram loss  $L_{hist}$ , validating the effectiveness of  $L_{hist}$ . However, at this stage, the stain-normalized images still exhibit obvious artifacts, likely due to the lack of constraints on image textures during training in the source domain. This issue was notably reduced after adding the SSIM loss  $L_{SSIM}$ , which enhanced the image quality. The content loss  $L_{cont}$  further refined the texture information and color richness of the images by extracting deep semantic relationships, constraining the color and structural information of the images. From a qualitative



**Fig. 7.** Visualization results of the Base line model sequentially adding different loss functions or strategies on the TUPAC-2016 and ICIAR-BACH-2018 datasets.

perspective, the DS and TS strategies that we introduced resulted in more uniformly stained images post-coloring and enriched the color and structure of the images, leading to a more realistic visual effect.

### Limitations

Despite the good performance demonstrated by the method we proposed on the aforementioned four datasets, there are still limitations. First, because our model uses a generator built with the Swin Transformer, it requires a significant amount of computational resources and has a lengthy training time. Additionally, due to the introduction of the DS and TS strategies, the batch size we can train on a single NVIDIA RTX 4090 machine does not exceed 2. Furthermore, our model still suffers from training instability, resulting in a large standard deviation in the results of independent experiments. Lastly, the datasets we currently use are relatively small in volume, which to some extent impacts the performance that relies on large datasets for training, such as CAGAN, and thereby affects the fairness of the results. In future work, we plan to introduce techniques such as pruning and quantization to compress the model size, making it suitable for resource-constrained clinical devices. Additionally, we aim to address the challenge of multi-locus variations by adopting strategies like multimodal data fusion or adaptive feature extraction. To tackle the previously discussed issue of white calibration, we will consider refining the loss function or employing blank region masks to prevent the generator from performing unnecessary transformations in these areas. Finally, we will collect multi-source datasets from different hospitals and devices and conduct multi-center experiments to validate the model's reliability and robustness in real-world applications, further enhancing the stability of model training.

### Conclusion

In this paper, we propose a Deep Supervision Two-stage Generative Adversarial Network (DSTGAN) for stain normalization, which effectively utilizes images from the source and target domains in the dataset and generates high-quality stain-normalized images. We constructed a powerful generator capable of capturing long-range dependencies using the Swin Transformer. Histogram loss was used to further constrain the color consistency of the images, and we introduced SSIM loss and content loss to regularize our model, enriching the texture information in the images. Additionally, we implemented deep supervision to further enhance the learning capability of the model, enabling it to generate high-quality images even in small-scale datasets. To fully utilize the images in the source domain, we designed a two-stage staining strategy, allowing the model to learn color

mapping relationships from unlabeled images in a semi-supervised manner. In the experimental stage, we conducted extensive evaluations of the model in four different datasets and evaluated the quality of the stain-normalized images quantitatively and qualitatively, as well as analyzed the impact of the model on classification and segmentation performance. Experimental results suggest that our model not only generates high-quality and uniformly stained images, but also exhibits excellent generalizability, and effectively enhances the performance of classification and segmentation.

## Data availability

The four datasets used in the paper are publicly available. The TUPAC-2016 dataset can be found at <https://tupac.grand-challenge.org/>; The MITOS-ATYP1A-14 dataset can be found at <https://mitos-atypia-14.grand-challenge.org/>; The ICIAR-BACH-2018 dataset can be found at <https://iciar2018-challenge.grand-challenge.org/>; The MICCAI-16-GlaS dataset can be found at <https://www.kaggle.com/datasets/sani84/glasmiccai2015-gland-segmentation>.

Received: 4 July 2024; Accepted: 21 February 2025

Published online: 27 February 2025

## References

- Gurcan, M. N. et al. Histopathological image analysis: A review. *IEEE Rev. Biomed. Eng.* **2**, 147–171 (2009).
- Ciampi, F. et al. 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017) 160–163 (IEEE).
- Gandomkar, Z., Brennan, P. C. & Mello-Thoms, C. MuDeRN: Multi-category classification of breast histopathological image using deep residual networks. *Artif. Intell. Med.* **88**, 14–24 (2018).
- Stanisavljevic, M. et al. *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.
- Kumar, A. et al. Deep feature learning for histopathological image classification of canine mammary tumors and human breast cancer. *Inf. Sci.* **508**, 405–421 (2020).
- Macenko, M. et al. 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro 1107–1110 (2009).
- Reinhard, E., Adhikhmin, M., Gooch, B. & Shirley, P. Color transfer between images. *IEEE Comput. Graph. Appl.* **21**, 34–41 (2001).
- Goodfellow, I. et al. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **27**, 1 (2014).
- Radford, A., Metz, L. & Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. Preprint at <http://arXiv.org/1511.06434> (2015).
- Zhu, J. Y., Park, T., Isola, P. & Efros, A. A. *Proceedings of the IEEE International Conference on Computer Vision* 2223–2232.
- Shaban, M. T., Baur, C., Navab, N. & Albarqouni, S. 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019) 953–956 (IEEE).
- Zhang, H., Goodfellow, I., Metaxas, D. & Odena, A. *International Conference on Machine Learning* 7354–7363 (PMLR).
- Wang, X., Girshick, R., Gupta, A. & He, K. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 7794–7803.
- Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, 1 (2017).
- Carion, N. et al. *European Conference on Computer Vision* 213–229 (Springer).
- Dosovitskiy, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. Preprint at <http://arXiv.org/2010.11929> (2020).
- Touvron, H. et al. *International Conference on Machine Learning* 10347–10357 (PMLR).
- Liu, Z. et al. *Proceedings of the IEEE/CVF International Conference on Computer Vision* 10012–10022.
- Salehi, P. & Chalechale, A. 2020 International Conference on Machine Vision and Image Processing (MVIP) 1–7 (IEEE).
- Wang, L., Lee, C. Y., Tu, Z. & Lazebnik, S. Training deeper convolutional networks with deep supervision. Preprint at <http://arXiv.org/1505.02496> (2015).
- Huang, H. et al. ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 1055–1059 (IEEE).
- Ouali, Y., Hudelot, C. & Tami, M. An overview of deep semi-supervised learning. Preprint at <http://arXiv.org/2006.05278> (2020).
- Lee, D. H. *Workshop on Challenges in Representation Learning, ICML* 896.
- Yalniz, I. Z., Jégou, H., Chen, K., Paluri, M. & Mahajan, D. Billion-scale semi-supervised learning for image classification. Preprint at <http://arXiv.org/1905.00546> (2019).
- Arazo, E., Ortego, D., Albert, P., O'Connor, N. E. & McGuinness, K. 2020 International Joint Conference on Neural Networks (IJCNN) 1–8 (IEEE).
- Fang, K. & Li, W. J. *International Conference on Medical Image Computing and Computer-Assisted Intervention* 532–541 (Springer).
- Kablan, E. B. & Ayas, S. StainSWIN: Vision transformer-based stain normalization for histopathology image analysis. *Eng. Appl. Artif. Intell.* **133**, 108136 (2024).
- Jiang, Y., Chang, S. & Wang, Z. Transgan: Two pure Transformers can make one strong Gan, and that can scale up. *Adv. Neural Inf. Process. Syst.* **34**, 14745–14758 (2021).
- Zhang, B. et al. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 11304–11314.
- Cao, H. et al. *European Conference on Computer Vision* 205–218 (Springer).
- Roy, S., Panda, S. & Jangid, M. 2021 29th European Signal Processing Conference (EUSIPCO) 1231–1235 (IEEE).
- Zheng, Y. et al. Adaptive color deconvolution for histological WSI normalization. *Comput. Methods Progr. Biomed.* **170**, 107–120 (2019).
- Salvi, M., Michielli, N. & Molinari, F. Stain color adaptive normalization (SCAN) algorithm: separation and standardization of histological stains in digital pathology. *Comput. Methods Progr. Biomed.* **193**, 105506 (2020).
- Vahadane, A. et al. Structure-preserving color normalization and sparse stain separation for histological images. *IEEE Trans. Med. Imaging* **35**, 1962–1971 (2016).
- Roux, L. 22nd International Conference on Pattern Recognition, Stockholm, Sweden.
- Kang, H. et al. Stainnet: a fast and robust stain normalization network. *Front. Med.* **8**, 746307 (2021).
- Bejnordi, B. E. et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *JAMA* **318**, 2199–2210 (2017).
- Shrivastava, A. et al. *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part I* 120–140 (Springer).
- Hetz, M. J., Bucher, T. C. & Brinker, T. J. Multi-domain stain normalization for digital pathology: A cycle-consistent adversarial network for whole slide images. *Med. Image Anal.* **94**, 103149 (2024).
- Bandi, P. et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE Trans. Med. Imaging* **38**, 550–560 (2018).
- Wilm, F. et al. *BVM Workshop* 206–211 (Springer).

42. Cong, C. et al. Colour adaptive generative networks for stain normalisation of histopathology images. *Med. Image Anal.* **82**, 102580 (2022).
43. Liu, S. et al. Isocitrate dehydrogenase (IDH) status prediction in histopathology images of gliomas using deep learning. *Sci. Rep.* **10**, 7733 (2020).
44. Spanhol, F. A., Oliveira, L. S., Petitjean, C. & Heutte, L. A dataset for breast cancer histopathological image classification. *IEEE Trans. Biomed. Eng.* **63**, 1455–1462 (2015).
45. Jose, L., Liu, S., Russo, C., Nadort, A. & Di Ieva, A. Generative adversarial networks in digital pathology and histopathological image processing: A review. *J. Pathol. Inf.* **12**, 43 (2021).
46. Jiao, Y., Li, J. & Fei, S. Staining condition visualization in digital histopathological whole-slide images. *Multimedia Tools Appl.* **81**, 17831–17847 (2022).
47. Cho, H., Lim, S., Choi, G. & Min, H. Neural stain-style transfer learning using GAN for histopathological images. Preprint at <http://arXiv.org/1710.08543> (2017).
48. Tellez, D. et al. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Med. Image Anal.* **58**, 101544 (2019).
49. Cong, C. et al. 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI) 1949–1952 (IEEE).
50. Zanjani, F. G., Zinger, S., Bejnordi, B. E. & van der Laak J. A. *Medical Imaging with Deep Learning*.
51. Salvi, M., Branciforti, F., Molinari, F. & Meiburger, K. M. Generative models for color normalization in digital pathology and dermatology: advancing the learning paradigm. *Expert Syst. Appl.* **245**, 123105 (2024).
52. Isola, P., Zhu, J. Y., Zhou, T. & Efros, A. A. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1125–1134.
53. Johnson, J., Alahi, A. & Fei-Fei, L. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II* 14 694–711 (Springer).
54. Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**, 600–612 (2004).
55. Ronneberger, O., Fischer, P. & Brox, T. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18 234–241 (Springer).
56. Afifi, M., Brubaker, M. A. & Brown, M. S. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 7941–7950.
57. Veta, M. et al. Predicting breast tumor proliferation from whole-slide images: the TUPAC16 challenge. *Med. Image Anal.* **54**, 111–121 (2019).
58. Aresta, G. et al. Grand challenge on breast cancer histology images. *Med. Image Anal.* **56**, 122–139 (2019).
59. Sirinukunwattana, K. et al. Gland segmentation in colon histology images: the Glas challenge contest. *Med. Image Anal.* **35**, 489–502 (2017).
60. Hore, A. & Ziou, D. 2010 20th International Conference on Pattern Recognition 2366–2369 (IEEE).
61. He, K., Zhang, X., Ren, S. & Sun, J. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 770–778.

## Author contributions

Z.D. was responsible for the planning of the project and overall research supervision; P.Z. performed the model construction and experimental analysis, and drafted the manuscript; X.H. provided the methodology and ideas for the experiments; Z.H. and D.L. conceptualized and revised the manuscript; and G.Y. collected the dataset; M.X. organized the experimental results. All authors approved this version of the manuscript to be published.

## Funding

This research was funded by Henan Province Medical Science and Technology Tackle Key Problems Project, LHGJ20230459.

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to D.L.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025