sensors

MDPI

Article

Controllable Unsupervised Snow Synthesis by Latent Style Space Manipulation

Hanting Yang [1,*], Alexander Carballo [2,3,4], Yuxiao Zhang [1] and Kazuya Takeda [1,3,4]

[1] Graduate School of Informatics, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan; yuxiao.zhang@g.sp.m.is.nagoya-u.ac.jp (Y.Z.); takeda@g.sp.m.is.nagoya-u.ac.jp (K.T.)
[2] Faculty of Engineering, Graduate School of Engineering, Gifu University, 1-1 Yanagido, Gifu City 501-1193, Japan; alexander@g.sp.m.is.nagoya-u.ac.jp
[3] Institute of Innovation for Future Society, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan
[4] Tier IV Inc., Nagoya University Open Innovation Center, 1-3, Mei-eki 1-chome, Nakamura-Ward, Nagoya 450-6610, Japan
* Correspondence: yang.hanting@g.sp.m.is.nagoya-u.ac.jp

Abstract: In the field of intelligent vehicle technology, there is a high dependence on images captured under challenging conditions to develop robust perception algorithms. However, acquiring these images can be both time-consuming and dangerous. To address this issue, unpaired image-to-image translation models offer a solution by synthesizing samples of the desired domain, thus eliminating the reliance on ground truth supervision. However, the current methods predominantly focus on single projections rather than multiple solutions, not to mention controlling the direction of generation, which creates a scope for enhancement. In this study, we propose a generative adversarial network (GAN)–based model, which incorporates both a style encoder and a content encoder, specifically designed to extract relevant information from an image. Further, we employ a decoder to reconstruct an image using these encoded features, while ensuring that the generated output remains within a permissible range by applying a self-regression module to constrain the style latent space. By modifying the hyperparameters, we can generate controllable outputs with specific style codes. We evaluate the performance of our model by generating snow scenes on the Cityscapes and the EuroCity Persons datasets. The results reveal the effectiveness of our proposed methodology, thereby reinforcing the benefits of our approach in the ongoing evolution of intelligent vehicle technology.

Keywords: intelligent vehicles; snow scenes; unpaired image-to-image translation; diversity; style latent space; Gaussian distribution

Citation: Yang, H.; Carballo, A.; Zhang, Y.; Takeda, K. Controllable Unsupervised Snow Synthesis by Latent Style Space Manipulation. *Sensors* **2023**, *23*, 8398. https://doi.org/10.3390/s23208398

Academic Editor: Felipe Jiménez

Received: 15 September 2023
Revised: 28 September 2023
Accepted: 8 October 2023
Published: 12 October 2023

1. Introduction

Intelligent vehicles and other advanced mobile agents are engineered to navigate through a spectrum of adverse weather conditions. This poses a formidable challenge to perception algorithms [1]. To enhance the robustness of these algorithms, a prevalent strategy involves augmenting the training dataset [2–4]. However, operating vehicles under such severe conditions contravenes road safety regulations, and data acquisition, in this case, becomes significantly time-consuming.

A viable alternative to traditional data collection is to synthesize weather effects on existing public benchmarks [5–9]. Conventionally, this is accomplished by modeling the impact of weather effects, such as fog, rain, and snow, as a function [10]. The derived function is subsequently applied to images or videos, thus simulating the desired weather conditions. This method facilitates the creation of diverse datasets, which serve as valuable resources for training and testing various perception algorithms, encompassing object detection, intention estimation, trajectory prediction, etc.

In light of the widespread adoption of deep learning methodologies, several researchers have begun considering the use of physical synthesis datasets as sources to

train more universally applicable convolutional neural networks (CNNs) or generative adversarial networks (GANs) [11,12]. These datasets furnish diverse and realistic training samples, without the need for actual data collection in perilous weather conditions. Still, the efficacy of this approach highly depends on the accuracy of the weather effect model and the quality of the synthesized images or videos.

A recently proposed concept involves utilizing unpaired image-to-image translation models. This type of model is capable of learning how to map visual features from a source domain to a target domain without one-to-one correspondence [13–15]. A prominent example in this domain is the CycleGAN architecture [14], designed to generate images based on GANs that are virtually indistinguishable from real photographs. The key innovation introduced through CycleGAN is the implementation of a cycle consistency loss. This feature encourages the mapping of an image from one domain to another to be consistent and vice versa. Consequently, the model is able to learn a mapping between two image collections, effectively capturing correspondences between higher-level appearance structures.

In our previous research [16], we implemented a CycleGAN-based model to synthesize realistic snow on driving scene images. We used the Cityscapes and EuroCity Persons datasets as source domains, while a self-captured snow collection functioned as the target. The image generation performance was assessed using a variety of image quality metrics. Benefiting from semantic information, each sample was effectively transformed into convincing snow scenes, while maintaining the integrity of the original image's structure and texture. However, the adopted method yielded a single output conditioned on the given input image, which does not fully leverage the inherent multimodality of the mapping between two visual domains. This limitation overlooks the potential diversity of snow scenes that may be present.

In the present study, we present a novel framework, controllable unsupervised snow synthesis (CUSS), devised to overcome the limitations inherent in existing snow synthesis methodologies. The reason we focus on snow is that snow can drastically reduce visibility, often more than rain or haze, and accumulating snow will cover the road surface and points of interest. The novelty of this work stems from the presumption that the snow representation can be decomposed into a texture-invariant content code and a snow-specific style code. Further, in the middle of the training process, we explore the latent space by a self-regression module. The module linearly interpolates the style code of the clear domain and the snow domain. After training, we can twist the style code with a hyperparameter that theoretically controls the size of the snow as shown in the Figure 1. This strategy facilitates a more comprehensive capture of the full distribution of potential outputs, marking a significant advancement within the domain. The key contributions of our research are as follows:

- Content and style disentanglement. The CUSS model employs an architectural framework comprising a content encoder and a style encoder. In order to separate the content and style latent spaces, we introduce a supplementary content discriminator that distinguishes the content codes of clear and snow images.
- Multimodal controllable output generation. The CUSS model allows for the generation of multiple and diverse outputs based on a single input image through the sampling of distinct style codes. Moreover, the incorporation of a self-regression module facilitates the linear interpolation of the style code, thereby enabling manual adjustment of the generated size of snow.
- Evaluations on public datasets. The model undergoes evaluation employing the Cityscapes and EuroCity Persons datasets. Various image quality metrics, encompassing the traditional PSNR, SSIM, and perceptual loss VGG distance, are employed to substantiate the effectiveness of the model. The source code will be available on https://github.com/HantingYang/Controllable-Snow-Synthesize (accessed on 13 September 2023).

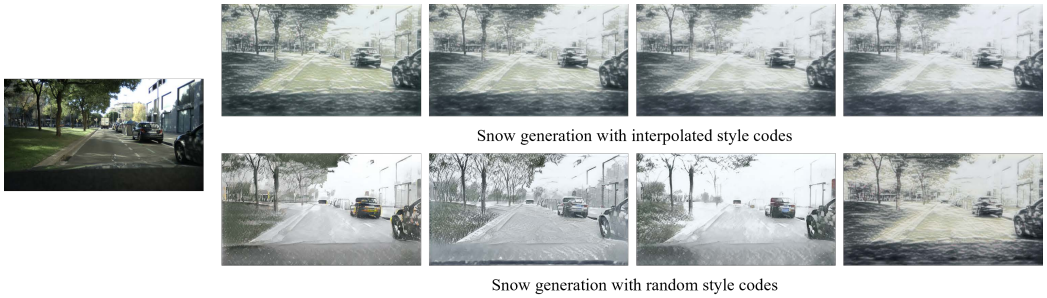

Figure 1. Examples of the output produced by the proposed snow synthesis model. In the upper row, the dimensions of the snow are progressively enlarged, an effect accomplished through the interpolation of style codes. Conversely, the images in the lower row are generated using randomly sampled style codes that follow a Gaussian distribution.

2. Related Work

2.1. Weather Generation

Despite extensive research on improving visibility during inclement weather, there has been limited attention given to incorporating artificial weather effects into existing driving datasets. Our literature review delves into the current techniques for synthesizing weather, particularly those leveraging deep learning. Utilizing a generative model for weather simulation could offer a versatile instrument for generating authentic conditions to evaluate and enhance AI-driven systems.

2.1.1. Fog and Rain

The fog generation process, as detailed in a study conducted by Christos et al. [6], comprises two primary steps. It is initialized with the estimation of a transmission map and atmospheric light based on a clear scene image. Following this, depth denoising and completion techniques are implemented to enhance the accuracy of the depth map. This refined depth map is subsequently employed to simulate fog, culminating in the generation of synthetic fog images that closely resemble real-world foggy scenarios. Zhang et al. [17] introduce a technique to generate haze images from clear ones, employing a network that includes two encoders with shared feature extraction layers. The essence of their approach lies in separating the style feature, used exclusively for haze synthesis, from the content feature that conveys consistent semantic information. The authors control a parameter α to synthesize multidensity haze images, with the networks learning to distinguish between thick and thin haze.

For the rain generation, Garg and Nayar [7] propose an adaptive image-based rendering technique that utilizes a database of rain streak appearances, requiring only the positions and attributes of light sources and a rough depth map, to add photorealistic rain to a single image or a recorded video with moving objects and sources. Venceslas et al. [18] discuss a real-time rendering algorithm for the realistic representation of fog in animations. The algorithm works by rendering fog for each pixel of the screen. The authors also introduce the concept of equipotential functions, which allow for the creation of complex shapes of fog.

2.1.2. Snow

Liu et al. [8] suggest a snow synthesis method that utilizes base masks representing different snow particle sizes—small, medium, and large. The snow synthesis process involves overlaying these base masks with images and introducing random factors such as snow brightness and random cropping to increase variation.

Ohlsson et al. [19] present a real-time method for rendering accumulated snow, which involves determining which regions should receive snow based on surface inclination and exposure to the sky, and then rendering the snow convincingly at those locations. The rendering process uses a Phong illumination model and a noise function to distort the

surface normally, creating a realistic snow cover appearance and an illusion of snow depth around occlusion boundaries. The exposure function is implemented using a depth buffer, similar to shadow mapping, and the depth map is sampled multiple times to calculate a fractional value for occlusion, creating a smooth transition between snow-covered and non-snow-covered areas.

Alexey et al. [20] introduce an innovative approach for snow simulation through a user-adjustable elastoplastic constitutive model paired with a hybrid material point method (MPM). The MPM employs a consistent Cartesian grid to facilitate self-collision and fracture automatically. Moreover, it utilizes a grid-centric semi-implicit integration scheme not reliant on the count of Lagrangian particles. This technique adeptly simulates diverse snow behaviors, especially the intricate dynamics of dense and wet snow, and incorporates rendering methods for a true-to-life visual depiction of snow.

As an evaluation work, Thomas et al. [5] take multiple cutting-edge image-to -mage (I2I) translation models for comparison and CycleGAN [14] as the baseline. The models used include UNIT [21] and MUNIT [22]. This work attempts to generate all kinds of bad weather images; the main focus is snow scenes. Authors believe that the identity loss that is calculated by the Manhattan distance between input and reconstructed images plays an essential role in the translation process. Therefore, they train the model several times and each time specify a different weight for the loss. In addition, to make up for the shortness of current datasets, images retrieved from the image engine Flickr are fed to their model.

Our previous work [16] presents a novel method for synthesizing realistic snow images on driving datasets using cycle-consistent adversarial networks. We introduce a multi-modality module that uses a segmentation map to accurately generate snow according to different regions of an image. We also propose a deep supervision module that adds extra side outputs to the discriminator, improving the network's learning of discriminative features. The model is evaluated using the same loss functions as CycleGAN [14]. The evaluation results on the Cityscapes and EuroCity Persons datasets show that the model outperforms other methods in generating realistic snow images.

From the above, generative models such as GANs are able to generate scenes under a variety of challenging conditions and make the output convincible. Models based on cycle consistency are able to generate images from the target domain without paired data. However, these models can only produce one output based on one input, and the degree of style transfer cannot be controlled. This work aims to further explore the latent space of extracted features and make the model produce diverse results.

2.2. Unpaired Image-to-Image Translation

Image-to-image (I2I) translation focuses on learning the mapping between two domains [23,24]. This involves capturing correspondences between higher-level appearance structures. The goal is to transform an image from a source domain to a target domain while preserving the underlying structure or context. Unpaired I2I translation further improves the training process, and it does not require paired input–output examples [13–15]. Instead, it assumes that there is some underlying relationship between the two domains and seeks to learn that relationship. This approach is particularly useful when image pairs are unavailable or the sensing environment is dangerous like driving scenes under challenging conditions.

Latent Space Constraint

With the success of unpaired I2I translation, researchers are now directing their attention to generating a more diverse range of output. This is achieved by a latent space constraint.

Zhu et al. [25] introduce the BicycleGAN model, designed to enhance image-to-image translation by producing diverse and realistic outcomes using a concise latent vector, and this model utilizes a combined technique that ensures a one-to-one consistency between latent encoding and output modes to avoid mode collapse. In their research, they explored

various methods of incorporating the latent code into the generator and observed similar performance levels while also investigating the balance between result diversity and sampling complexity by adjusting the latent code's dimensionality.

Lee et al. [26] suggest a technique that projects input images into a joint content space and domain-distinct attribute areas. The content encoders relay the mutual details shared across domains to shared content space, and attribute encoders relay the domain-unique data to specific attribute space. To handle datasets without pairs, they introduced a cross-cycle consistency loss leveraging the separate representations.

Huang et al. [22] showcase the multimodal unsupervised image-to-image translation (MUNIT) structure, which differentiates image representation into a universal content code and a domain-centered style code, and incorporates two autoencoders, competitive objectives, and two-way reconstruction objectives to produce a range of results from a single source image. Furthermore, the model introduces the concept of style-enhanced cycle consistency, ensuring that the original image is recovered when converted to a target domain and reverted using its initial style.

Choi et al. [27] discuss a unified model called StarGAN that handles I2I translations across multiple domains. The generator takes an input image and a target domain label to generate a fake image. This target domain label is represented as a binary or one-hot vector for categorical attributes. The generator focuses on the explicitly given label and ignores unspecified labels by setting zero vectors, enabling the model to generate high-quality images.

Liu et al. [28] introduce a technique named the unified feature disentanglement network (UFDN) designed for self-supervised feature decomposition. They utilize a variational autoencoder (VAE) structure to achieve disentangled representations across various data domains. The encoder accepts an image, processes its representation, and then merges it with the domain vector. These combined data are then used by the generator to recreate the image.

Inspired by the above work, we introduce a content and style representation from our previous snow synthesis framework [16]. For intuition, we divide the translation step into three parts: encoding, translation, and decoding. The encoding network encodes the input image into one style code and one content code. By swapping and twisting the style code generated by the style encoder, we can obtain diverse but high-quality outputs. In addition, we interpolate the style code between the clear and snow domains to obtain gradually increasing snow effects. This is achieved by disentangling the latent space.

3. Controllable Unsupervised Snow Synthesis

To synthesize realistic snow on the driving datasets, we focus on the GAN with cycle consistency. The goal is to learn the mapping between the snow domain and the clear weather domain. In our previous unpaired I2I methods [29,30], two generators are employed to transfer images into the expected domain. Two corresponding discriminators are employed to differentiate real images and fake images. The cycle consistency ensures that translated images can be reconstructed into original input images.

Recently, when researchers use similar methods for weather removal or synthesis, they follow an assumption that weather images can be decomposed into a content partition and a weather partition [17,31,32]. The partition could be any mathematical format, such as vectors or tensors. In general image translation tasks, the weather partition refers to the style representation. This technique will disentangle the translation process and preserve the structural feature of the background. Therefore, we follow the assumption and split the generator into three networks, which are a style encoder, a content encoder, and a decoder.

In the field of representation learning, incomplete disentanglement is often more prevalent. This concept suggests that images from varying domains have a shared content representation space, but the style representation space remains unique to each domain. This idea is also known as the shared latent space assumption. In our task, the style is

related to snow, and different classifications detail the attributes of weather events that produce snow.

Intuitively, the content codes and style codes should be disjoint in the representation space. To better achieve representation disentanglement, we apply a content discriminator to distinguish the domain membership of the encoded content features. The goal is to force content encoders to generate features that cannot be identified, which means the content code does not contain style details.

Further, in order to make the size of synthesized snow controllable, we need to explore the space of style partition S. Inspired by the work of Zhang et al. [17], we transform the snow domain into a continuous space by associating the style code vectors with a linear manipulation. With the help of the content discriminator, the style code will not contain information on image attributes. Ideally, the interpolated style code should represent an intermediate snow density.

3.1. Fundamental Basis

To illustrate the framework of controllable unsupervised snow synthesis (CUSS) in an intuitive way, suppose that $x_1 \in X_1$ and $x_2 \in X_2$ are images from the clear domain and the snow domain, respectively. In statistics, the images belong to two marginal distributions, $p(x_1)$ and $p(x_2)$. The joint distribution $p(x_1, x_2)$ is inaccessible due to a lack of paired data. The goal is to learn an I2I translation model that can estimate two conditionals, $p(x_{1\to2}|x_1)$ and $p(x_{2\to1}|x_2)$, where $x_{1\to2}$ is a sample of synthesized snow images and $x_{2\to1}$ is a sample of synthesized clear images (recovered from real snow samples). In general, the synthesis outputs do not fall into a single mode. There are multiple solutions corresponding to the transform problem.

To obtain other possible solutions, we adopt the partially shared latent space assumption from MUNIT [22] to produce diverse snow effects. This theory posits that each image $x_i \in X_i$ originates from a content latent code c_i, shared across both domains and a unique style latent code s_i tied to its respective domain. For snow synthesis, a matching pair of clear and snow images (x_1, x_2) from the combined distribution is created by $x_1 = F_1(c_1, s_1)$ and $x_2 = F_2(c_2, s_2)$, with F_1, F_2 as the foundational generators with the inverse encoders E_1 and E_2, with $E_1 = (F_1)^{-1}$ and $E_2 = (F_2)^{-1}$.

The structure of the CUSS model is depicted in Figure 2. As displayed in Figure 2a, our conversion model has an encoder E_1 and a decoder F_1 for the clear domain X_1, and an encoder E_2 and a decoder F_2 for the snow domain X_2. Each image fed into the encoder becomes converted into a content code c and a style code s, represented as $(c, s) = E(x)$. The translation between images occurs by interchanging encoder–decoder pairs, as depicted in Figure 2b. For instance, to transform a clear image $x_1 \in X_1$ to X_2, we first capture its content latent code $c_1 = E_1^c(x_1)$ and draw a style latent code s_2 from the normal distribution $q(s_2) \sim \mathcal{N}(0, I)$. Then, we employ F_2 to generate the ultimate snow image $x_{1\to2} = F_2(c_1, s_2)$.

In earlier research [16], we harnessed the cycle consistency loss [14], measured by the L1 norm of the input image. This aimed to deter the secondary generator from producing arbitrary target domain images. However, Huang et al. [22] demonstrated that if cycle consistency is imposed, the translation model becomes deterministic. As a result, we integrated a style-enhanced cycle consistency in the image-style joint spaces, which aligns better with multimodal image conversion. As illustrated in Figure 2c, we derive the content code $c_{1\to2}$ and style code $s_{1\to2}$ from the synthetic snow image $x_{1\to2}$. We then feed the content code $c_{1\to2}$ and the identical style latent code s_2 to the clear decoder F_1. The result image is named cycle clear image $x_{1\to2\to1}$. The idea behind style-enhanced cycle consistency is that by translating an image to a target domain and then back with the original style, we should retrieve the initial image. We do not apply explicit loss measures to ensure this style-enhanced cycle consistency, but it is suggested by the bidirectional reconstruction loss. We show the pseudo-code of CUSS in Algorithm 1.

Algorithm 1 Controllable Unsupervised Snow Synthesis (CUSS)

Input: Training data pairs (X_1, X_2) ▷ In order of clear and snow
Output: Encoders E_1, E_2, Decoders F_1, F_2 ▷ F_1 generate clear images, F_2 generate snow images

Initialize encoders, decoders, and discriminators
Define loss functions
Define optimizers for generator and discriminator
while $epoch \leq total\_epoches$ **do**
  for data pair (X_1, X_2) **in** data_loader **do**
    Get content codes and style codes of input images: $(c_1, s_1) = E_1(x_1)$, $(c_2, s_2) = E_2(x_2)$, $(s_{n1}, s_{n2}) \sim \mathcal{N}(0, 1)$ ▷ s_{n_i} means style code sampled from normal distribution
    Generate fake images: $x_{1\to2} = F_2(c_1, s_{n1})$, $x_{2\to1} = F_1(c_2, s_{n2})$
    Generate reconstruct images: $x_{1\to1} = F_1(c_1, s_1)$, $x_{2\to2} = F_1(c_2, s_2)$
    Get content codes and style codes of fake images: $(c_{21}, s_{21}) = E_1(x_{2\to1})$, $(c_{12}, s_{12}) = E_2(x_{1\to2})$
    Generate cycle translation images: $x_{1\to2\to1} = F_1(c_{12}, s_1)$, $x_{2\to1\to2} = F_2(c_{21}, s_2)$
    Update Discriminator D_1, D_2, and D_c
    Update Generator E_1, E_2, F_1, and F_2
  end for
end while

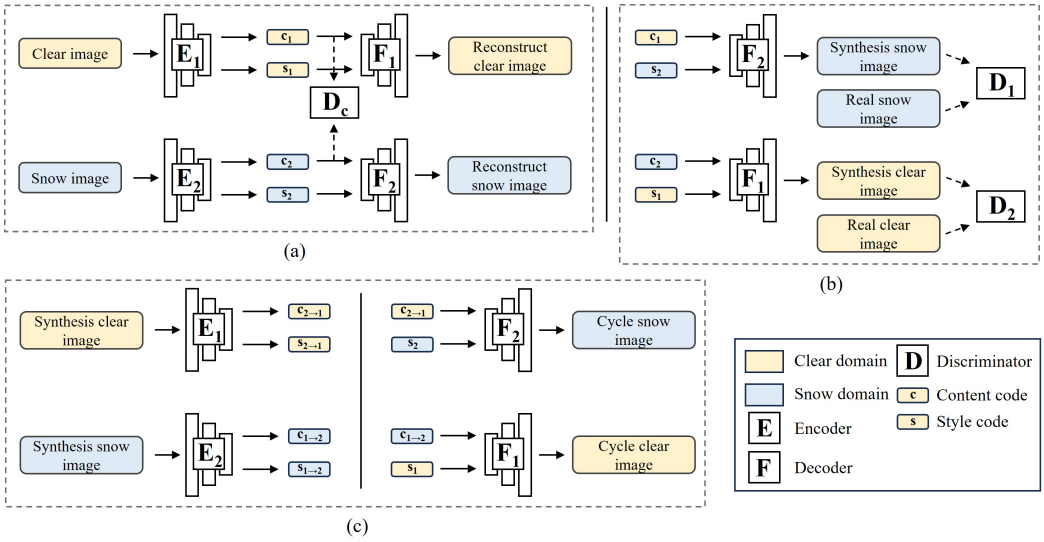

Figure 2. The architectural design of the proposed controllable unsupervised snow synthesis (CUSS) network is outlined as follows. The solid arrows show the forward process of the generators. The dashed arrows show the input to the discriminators. CUSS comprises two encoders, which assume the responsibility of encoding images from both the clear and snow domains, yielding a content code and a style code, respectively. Additionally, CUSS incorporates two decoders, which accept a content code and a style code as input, subsequently generating synthetic images pertaining to the target domain. Moreover, there exist two discriminators, whose purpose is to discern images originating from each domain, alongside a content discriminator, which endeavors to discriminate between content codes. (**a**) illustrates the fundamental pipeline, wherein the encoded images ought to be recoverable utilizing identical codes. Conversely, (**b**) portrays the process of translation accomplished by substituting the style code with a randomly sampled one. Lastly, (**c**) exemplifies the translation process's cycle consistency, whereby the translated synthetic images ought to revert to the original input, utilizing the initially extracted style code in conjunction with their own content code.

3.2. Disentanglement of Content and Style

A disentangled representation captures the underlying structure of the data so that individual factors can be modified independently without affecting others. The goal is to

achieve complete disentanglement, where both content and style features are extracted independently. To achieve this, a content discriminator D_c is used to remove style information from the content feature. At the same time, we use self-supervised style coding to reduce content information from the style feature.

To enhance the content encoder, we employ the content feature discriminator proposed by Lee et al. [26]. Initially, the content encoder extracts content codes, denoted as c_1 and c_2, from the respective inputs x_1 and x_2. The content discriminator D_c takes input images and classifies their source domain. Then one objective of the content encoder is to deceive D_c with distinct features. As a result, the content encoder and discriminator refine each other through adversarial training. Once equilibrium is reached, the content features extracted no longer retain any stylistic information of the image.

When this game of generators and discriminators stabilizes at the Nash equilibrium [33], it becomes impossible for D_c to ascertain the image domain of the content feature, implying an absence of snow details in the content feature. A successful separation of style from content is achieved when the content encoder exclusively captures the image's content characteristics.

It is proved that utilizing the content discriminator can prevent content codes from containing style details [26]. Naturally, the next step is to remove content details from style codes. The purpose is to make the generation more stable without being affected by other factors. Consequently, we implement the self-supervised style coding to remove any excess content details from the style codes, illustrated in Figure 3.

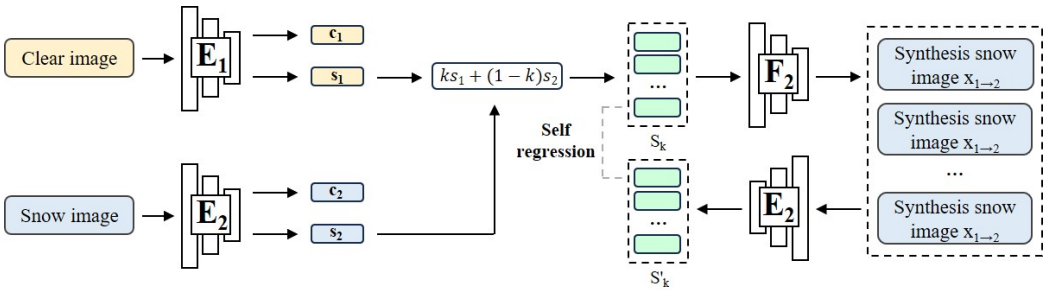

Figure 3. Snow sizing through self-regression style coding. Our methodology involves extracting style codes from two distinct domains. By using linear interpolation between these domains, guided by the parameter k, we are able to generate a range of snow sizes. A higher value of k gives greater significance to the snow dimensions. We then merge the content and interpolated style features to create a snow scene image with a specific snow density determined by the aforementioned parameter k. Throughout the training process, we use a randomly selected k value from the interval [0, 1] to derive a novel style code that represents an intermediate level of snow density. This newly generated style code serves as a self-supervised pseudo-label, effectively guiding the updating process of the style encoder.

Using a nonlinear function f to denote the style encoder, we initially perform interpolation on two style codes, one from the clear domain and the other from the snow domain to acquire s_k, as shown in Equation (1).

$$s_k = f(kx_1 + (1-k)x_2) \tag{1}$$

In the scope of our problem, the need to disentangle the style feature from the content feature makes sure that the operation on the style code is consistent with those on the input image.

According to the derivation of Zhang's work [17], because s_k is calculated by the linear projection of s_1 and s_2, it should contain snow detail that is also a linear relation of x_1 and x_2. Use s_k and a content code of a clear input c_1 and a new snow image x_k can be

generated. We then encode s_k again to obtain its style code, which will be supervised by s_k itself; the loss function is defined as Equation (2).

$$\begin{aligned} &\mathcal{L}_s(E_1^c, E_2^s, F_2) = \\ &\mathbb{E}_{x_1,x_2}\left[\|E_2^s(F_2(E_1^c(x_1), s_k)) - s_k\|_1\right. \end{aligned} \tag{2}$$

Even though the function f is nonlinear, we continually generate x_k and optimize the encoder with s_k in the training phase to maintain a consistent relationship between input images and corresponding style codes linearly.

In the early stages, the style code will contain an extra content detail because of latent space entanglement. At every forward iteration, the extra details become separated with the style codes. Instinctively, the style encoder will identify content details and ignore them.

In the absence of manually assigned labels, the process relies on s_k as a self-generated label to guide the updates of networks. The desired situation is that the style code will generate snow according to each object distribution at every distance and not decrease the information density of traffic sign areas.

Due to the stochastic choice of k at each forward iteration, the encoder is compelled to project the snow-related detail into a linear space. As a result, we engage in linear adjustments to style codes to generate images with varying snow densities.

For example, we can specify the k value presenting the $k \times 100\%$ snow density of the input snow image. Then we extract the style code and content code of the input clear image. After that, we feed the content code and interpolate the style code to the decoder to obtain the output.

The factor k governs the snow density. Since s_2 originates from the baseline snow, it can be scaled up or down using k to yield a background invariant image featuring different levels of snow density as depicted in Equation (3).

$$x_k = F_2(E_1^c(x_1), kE_1^s(x_1) + (1-k)E_2^s(x_2)) \tag{3}$$

3.3. Loss Function

The comprehensive loss function discussed in this paper comprises several components: the adversarial loss $\mathcal{L}_{adv}$, image reconstruction identity loss $\mathcal{L}_{id}$, style reconstruction loss $\mathcal{L}_{recon}^s$, content reconstruction loss $\mathcal{L}_{recon}^c$, style regression loss $\mathcal{L}_{regre}$, cycle consistency loss $\mathcal{L}_{cc}$, and content loss $\mathcal{L}_{cont}$. The overall objective function is formulated as the weighted sum of these individual loss components:

$$\begin{aligned} \mathcal{L} =& \lambda_{adv}\mathcal{L}_{adv} + \lambda_{id}\mathcal{L}_{id} + \lambda_{recon}^s\mathcal{L}_{recon}^s + \lambda_{recon}^c\mathcal{L}_{recon}^c \\ &+ \lambda_{regre}^s\mathcal{L}_{regre}^s + \lambda_{cc}\mathcal{L}_{cc} + \lambda_{cont}\mathcal{L}_{cont} \end{aligned} \tag{4}$$

Here, $\mathcal{L}_{adv} = \mathcal{L}_{D_1} + \mathcal{L}_{D_2}$. $\mathcal{L}_D$ represents the adversarial losses in the clear and snow images. The various λ terms act as the model's hyperparameters, modulating the significance of each loss component.

3.3.1. Adversarial Loss

Adversarial loss is employed in both the clear and snow domains to enhance the realism of the generated images. In the domain of clear images, the adversarial loss is specified as follows:

$$\begin{aligned} \mathcal{L}_{D_1} =& \mathbb{E}_{x_1 \sim P_{X_1}}[\log D_1(x_1)] \\ &+ \mathbb{E}_{x_2 \sim P_{X_2}}[\log(1 - D_1(F_1(E^c(x_2), s_1)))] \end{aligned} \tag{5}$$

D_1 serves the purpose of differentiating real clear images from their synthesized counterparts, striving to maximize the aforementioned loss function. On the other hand, F_2

aims to reduce the loss in order to make the generated clear images appear more authentic. Likewise, $\mathcal{L}_{D_2}$ for the snow domain is defined as

$$\begin{aligned}\mathcal{L}_{D_2} = &\mathbb{E}_{x_2 \sim P_{X_2}}[\log D_2(x_2)] \\ &+ \mathbb{E}_{x_1 \sim P_{X_1}}[\log(1 - D_2(F_2(E^c(x_1), s_2)))]\end{aligned} \tag{6}$$

We consider both adversarial losses to have equal impact and straightforwardly sum them up to compose the ultimate adversarial loss.

$$\mathcal{L}_{adv} = \mathcal{L}_{D_1} + \mathcal{L}_{D_2} \tag{7}$$

3.3.2. Identity Loss and Latent Space Reconstruction Loss

When provided with a snow image and a clear image, the encoders are required to recreate the input image based on the same content code and style code. As such, the disparity between the reassembled image and the initial image serves as the reconstruction loss, adding additional constraints to the encoder:

$$\begin{aligned}\mathcal{L}_{id} = &\mathbb{E}_{x_1 \sim P_{X_1}}\left[\|F_1(E_1^c(x_1), E_1^s(x_1)) - x_1\|_1\right] \\ &+ \mathbb{E}_{x_2 \sim P_{X_2}}\left[\|F_2(E_2^c(x_2), E_2^s(x_2)) - x_2\|_1\right]\end{aligned} \tag{8}$$

Additionally, we aim for the decoded images to have content and style features that closely resemble those in the original images. As a result, we define the following losses for the reconstruction of content code and style code:

$$\begin{aligned}\mathcal{L}_{recon}^c = &\mathbb{E}_{x_1 \sim P_{X_1}}[\|E_1^c(x_{2\to1}) - E_1^c(x_1)\|_1] \\ &+ \mathbb{E}_{x_2 \sim P_{X_2}}[\|E_2^c(x_{1\to2}) - E_2^c(x_2)\|_1]\end{aligned} \tag{9}$$

$$\begin{aligned}\mathcal{L}_{recon}^s = &\mathbb{E}_{x_1 \sim P_{X_1}}[\|E_1^s(x_{2\to1}) - E_1^s(x_1)\|_1] \\ &+ \mathbb{E}_{x_2 \sim P_{X_2}}[\|E_2^s(x_{1\to2}) - E_2^s(x_2)\|_1]\end{aligned} \tag{10}$$

It is important to note that we treat the reconstruction loss of a style code as falling under the umbrella as the self-supervised style coding loss. We sum these two up, applying the same weight to both, to arrive at the final style coding loss:

$$\mathcal{L}_{recon}^s = \mathcal{L}_{recon}^s + \mathcal{L}_s \tag{11}$$

3.4. Cross-Cycle Consistency Loss

Our model incorporates the cross-cycle consistency loss, as referenced in [26], to facilitate the learning of domain mappings. For the generated snow image $x_{1\to2}$, its corresponding clear image x_2 can be recovered through a desnowing transformation. The cross-cycle consistency loss constrains the scope of the generated image while maintaining the background information of the input images. The Manhattan distance between the cyclically reconstructed image and the original image serves as the measure for this cross-cycle consistency loss. The image conversion process, which involves converting the clear image to the snow image and the other way around, proceeds in Equation (12):

$$\begin{aligned}x_{1\to2} &= F_2(E_1^c(x_1), E_2^s(x_2)) \\ x_{2\to1} &= F_1(E_2^c(x_2), E_1^s(x_1))\end{aligned} \tag{12}$$

The reverse translation operation, which entails reconstructing the original input from the generated image, is outlined in Equation (13):

$$\begin{aligned}x_{1\to2\to1} &= F_1(E_2^c(x_{1\to2}), E_1^s(x_{2\to1})) \\ x_{2\to1\to2} &= F_2(E_1^c(x_{2\to1}), E_2^s(x_{1\to2}))\end{aligned} \tag{13}$$

The formulation of the cross-cycle consistency loss for both the snow and clear image domains is

$$\begin{aligned}\mathcal{L}_{cc} = \mathbb{E}_{x_1 \sim P_{X_1}}[\|x_1 - x_{1\to2\to1}\|_1] \\ + \mathbb{E}_{x_2 \sim P_{X_2}}[\|x_2 - x_{2\to1\to2}\|_1]\end{aligned} \tag{14}$$

4. Experiments

To validate the efficacy of the approach described in this paper, this section delves into the influence of various modules and loss functions on the generated outcomes. It also benchmarks these outcomes against existing methods through both quantitative and qualitative metrics. Initially, we provide an overview of the datasets used and the implementation specifics of the approach. Subsequently, we offer an in-depth examination of our model, comparing it with current methodologies in the field. We further support the model's effectiveness by showcasing visualizations of intermediate outcomes and conducting a generalization analysis. The final portion of this section focuses on ablation studies to scrutinize the model's components. All testing and experimentation are performed on an NVIDIA RTX A6000 GPU equipped with 24 GB of memory.

4.1. Datasets

4.1.1. Urban Sceneries: Cityscapes

The Cityscapes collection [34] consists of 5000 detailed images showcasing urban landscapes, predominantly captured in various German cities during daylight hours. This dataset, which primarily focuses on street vistas, intersections, and vehicular scenes, has gained significant popularity for training and evaluating machine vision systems. We use all images as a clear source to train the model, as the number of images is close to our snow set.

4.1.2. European Urban Scenes: EuroCity Persons

The EuroCity Persons collection [35] comprises a vast array of photographs depicting pedestrians, cyclists, and other moving figures within city traffic scenarios. These images were captured from a mobile vehicle across 31 cities in 12 European nations. Each image in this collection is accompanied by a comprehensive set of precise annotations, including bounding boxes around pedestrians and cyclists, as well as additional information, such as direction, visibility, and potential obstructions. This extensive collection is further divided into separate segments for daylight and nighttime scenes, encompassing a grand total of over 47,300 images. To maintain consistency with the snow dataset, we handpicked 5921 snapshots from the daytime training segment.

4.1.3. Snow Condition Driving Dataset

In order to provide an authentic and realistic benchmark for learning, we recorded a comprehensive video during adverse snowfall conditions. Employing a high-resolution camera positioned behind the windshield of the vehicle, we captured footage at an impressive frame rate of 120 frames per second. From this extensive collection, we carefully selected the highest-quality images and resized them to a resolution of 960×540. Some of the examples are shown in the Figure 4. As a result, our curated snow dataset [30] comprises a total of 6814 meticulously curated photographs. The number of intercepted images we keep is the same as for cityscapes because GAN training is prone to problems such as mode collapse, which leads to training failure.

Figure 4. A collection of self-captured videos depicting urban driving amidst intense snowfall [30]. The images are carefully screened. The footage includes various road users, including cyclists, automobiles, buses, and pedestrians.

4.2. Implementation Details

Our proposed model's network comprises two encoders, two decoders, two discriminators, and a content encoder. Among the encoders, one is designed for style, and the other for content. Their structure aligns with what is described in [26]. Breaking it down,

- The content encoder has five convolutional layers.
- The style encoder includes an initial residual layer, two downsampling layers, and one adaptive average pooling layer.
- The content encoder features an initial residual layer, two downsampling layers, and four residual blocks.
- Each decoder is made up of four residual blocks and two upsampling layers. It employs adaptive instance normalization, while the encoders use standard instance normalization.
- All the discriminators take specific image patches with the same resolution as input, which is inspired by Demir's work [36]. This structure includes five convolutional layers.

For training, we implement minibatch stochastic gradient descent with a batch size of 12, and the Adam optimization technique (parameters: $\beta_1 = 0.5, \beta_2 = 0.999$). We initiate with a learning rate of 0.0001, reducing it linearly from the 100th epoch. In the training phase, the input is cropped to a 256×256 resolution for input. The weight of each loss function is listed below: $\lambda_{adv} = 1, \lambda_{id} = 10, \lambda^{c}_{recon} = 1, \lambda^{s}_{recon} = 1 \lambda^{s}_{regre} = 1, \lambda_{cc} = 1$.

4.3. Performance Assessment

In this section, we present a comprehensive analysis comparing the outputs CUSS with the current state-of-the-art (SOTA) I2I transition methods. We delve deeply into the impact of disentanglement and conclude with a meticulous examination of the uniqueness and significance of each module based on ablation studies.

4.3.1. Assessment Criteria

Initially, we evaluate the quality of image synthesis using traditional computer vision measures, namely, the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM). Additionally, we employ metrics that specifically focus on the depth and the perception features of the images, including the Fréchet inception distance (FID) [33], the LPIPS distance [37], and the VGG distance [38].

PSNR serves as a reliable objective measure for images, quantifying the discrepancy between corresponding pixel values. Higher PSNR values indicate reduced distortion in the generated images.

SSIM, on the other hand, assesses the similarity between two images by considering their luminance, contrast, and structure. An SSIM value of 1 indicates that the two compared images are identical in terms of structural information and quality, while a value of 0 indicates that the images are entirely different in these respects.

FID calculates the Fréchet distance between two sets of images based on the features extracted by the inception network [39]. It provides a measure of similarity between the generated images and their respective benchmarks. A lower FID value suggests that the generated images closely resemble the benchmark images.

LPIPS, or learned perceptual image patch similarity, is a metric used to evaluate perceptual differences between images [25]. Unlike traditional metrics, such as mean squared error (MSE), which measure pixel-level differences or structural similarities, LPIPS employs deep learning to better align with human visual perception. In essence, LPIPS offers a more perceptually meaningful measure of image similarity, especially useful in tasks like image synthesis, where the objective is not just to reproduce pixel-accurate outputs but to generate outputs that are perceptually indistinguishable or pleasing to humans.

VGG distance refers to a perceptual loss metric based on the VGG network that was originally designed for image classification tasks [38]. Similar to LPIPS, it is used to measure the difference between two images in a feature space. The activations from one or more layers of the VGG network capture higher-level content and texture information about the images.

4.3.2. Qualitative Results

We generate snow images in varying sizes by adjusting the previously discussed parameters and contrast our proposed model against the leading state-of-the-art methods.

In our experimental setting, we take the Cityscapes dataset and the EuroCity Persons dataset as the target set. These two datasets contain over 5000 thousand road scenarios under different urban and weather conditions. There are also variations of road users, such as pedestrians, cyclists, and moving vehicles.

Figure 5 displays images with varying amounts of snow. It is important to note that we assume that the k value of the input clear image is 0, while the value of the input snow image is 1. We then adjust the snow feature within the range of 0 to 1. The differences in snow density across images with distinct parameter values demonstrate our success in differentiating the generated snow through manipulation. As an illustration, as the value increases, objects at the far end of the image become less distinguishable. Due to the impact of the style feature coding, the style encoders can identify between large and small snowflakes.

The results of the qualitative comparison are shown in Figure 6. This experiment compares the generated snow images with mainstream I2I translation methods (CycleGAN [14], CUT [40], MUNIT [40], and DRIT [26]). The first two models use single projections, while the last two can produce diverse outcomes. For a more accurate comparison, we consistently use ResNet [41] as the backbone for the generator in all methods. The training data use Cityscapes [34] and EuroCity Persons [35] as clear sources and the self-captured snow set as target sources. The methods used for comparison all require no paired data.

The qualitative comparison shows that models such as CUT [40], MUNIT [40], and DRIT [26] mainly exhibit three primary defects. First, after translating images to represent snowy scenes, the original colors are often distorted, diminishing the natural appearance of the scene. Second, these models sometimes introduce artifacts that were not present in the original image, leading to inconsistencies and jarring visual outcomes. Lastly, they inadequately handle the far end and sky regions, resulting in uneven or unrealistic snow representation in these areas. In contrast, the method proposed in this paper offers several advantages. Our approach naturally integrates snow, ensuring that its boundaries fade out seamlessly across the image, providing an authentic representation in both the foreground and background. By distinguishing between snow style and actual image content, our method is able to capture and reproduce the intrinsic properties of snow, resulting in a synthesis that feels genuine and consistent throughout the image. Moreover, while other models might render trees or other objects as if they were buried under un-natural snow formations, our technique retains the original structure and detail, providing a more balanced and realistic representation.

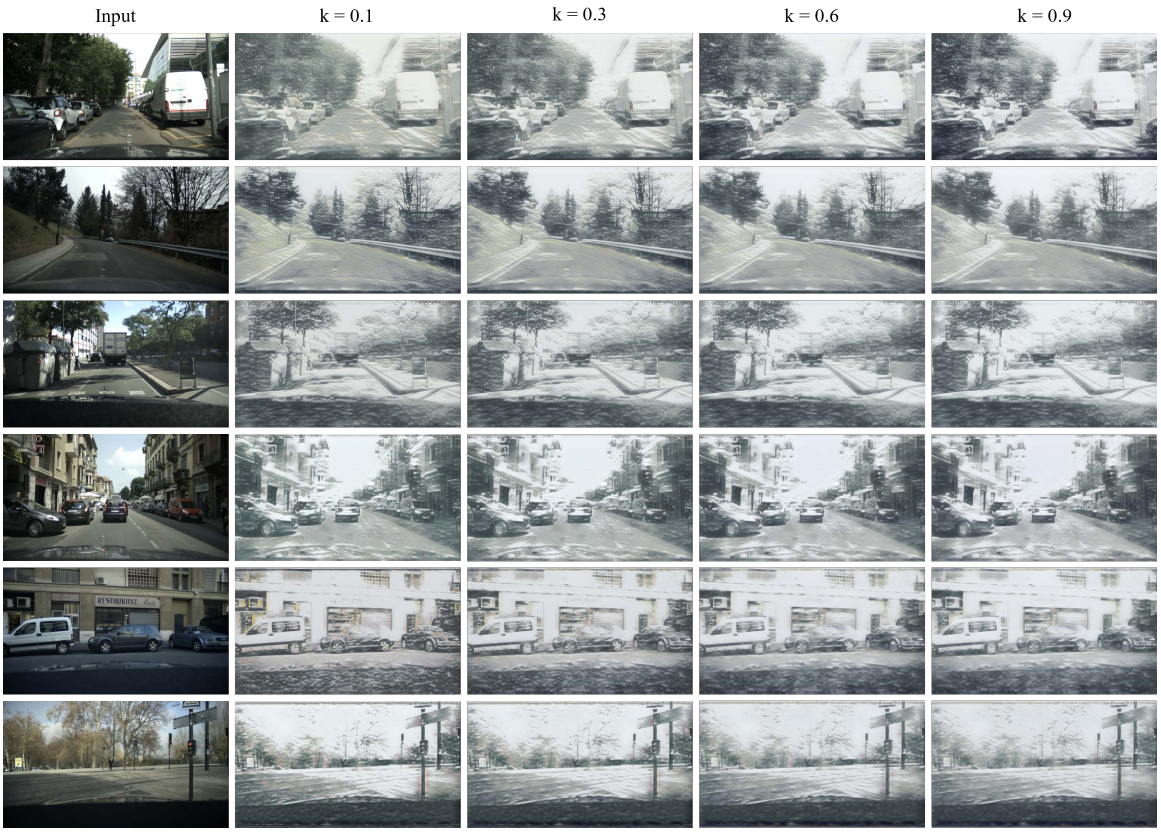

Figure 5. Synthesis of multidensity results on the EuroCity Persons dataset by adjusting the parameter *k*. From the left to the right column, objects such as vehicles, people, and trees are covered with snowflakes and haze that gradually increase in size.

Figure 6. Comparisons between the synthesized snow images produced by our method and SOTA unsupervised image translation methods. In particular, CUT deviates from the utilization of cycle consistency and the associated loss, as observed in CycleGAN. Conversely, the remaining models, such as CUSS (controllable unsupervised snow synthesis), incorporate a form of partial style cycle consistency.

4.3.3. Qualitative Results

As reported in other I2I works, the constraint of cycle consistency is strong so that the ability to generate diverse output is suppressed. However, the output image will retain a high similarity of the original image, which explains why CycleGAN [14] achieves the best SSIM value as shown in Tables 1 and 2. Compared with other methods, CUSS combines the content discriminator and style code manipulation, and both turn out effective for high-quality synthesizing. Therefore, CUSS achieves better results on those metrics. CUT is the only method that does not employ any format of cycle generation pipeline, but instead uses contrastive learning. The data used in our experiment cannot satisfy the requirement of a large batch size, which cannot make full use of contrastive loss. The results of CUSS prove that our method is available even when the data are not sufficient.

Table 1. A comparison on the Cityscapes dataset is made between SOTA image translation techniques through numerical evaluation. We generate images with $k = 1$. The metrics used for evaluation are d_{VGG} and d_{LPIPS}, which respectively represent the VGG and LPIPS distances. It should be noted that down arrows mean lower values for d_{VGG} and FID indicate more favorable experimental outcomes. Conversely, up arrows mean that higher values for the other metrics suggest superior results. Optimal values are denoted in bold.

Methods	$SSIM \uparrow$	$PSNR \uparrow$	$d_{VGG} \downarrow$	$FID \downarrow$	$d_{LPIPS} \uparrow$
CUT	0.392	15.853	6.063	26.156	0.047
CycleGAN	**0.471**	16.072	5.892	26.342	0.048
MUNIT	0.452	16.118	5.923	25.447	0.049
DRIT	0.446	16.432	5.426	25.874	0.049
CUSS	0.465	**16.912**	**5.122**	**25.103**	**0.047**

Table 2. A comparison on the EuroCity Persons dataset is made between SOTA image translation techniques through numerical evaluation. We generate images with $k = 1$. The same image quality metrics are used. Down arrows mean lower metric values are better and up arrows mean higher values are better. The best results are shown in bold.

Methods	$SSIM \uparrow$	$PSNR \uparrow$	$d_{VGG} \downarrow$	$FID \downarrow$	$d_{LPIPS} \uparrow$
CUT	0.396	15.912	6.123	26.245	0.048
CycleGAN	**0.501**	16.233	5.927	26.581	0.049
MUNIT	0.479	16.483	6.012	25.847	0.048
DRIT	0.469	16.741	5.756	26.007	0.048
CUSS	0.494	**16.983**	**5.386**	**25.402**	**0.051**

To understand the individual contribution of different components of CUSS, we conduct an ablation study with respect to the loss functions. Since loss functions reflect the direction of model optimization, we not only validate the new module of the content discriminator and self-supervised style coding but also test the improvement from reconstructing the image, style code, and content code. In Table 3, we observe that each component is crucial to the CUSS model presented in the decrease of the metrics. We generate images with three sets of k (0.3, 0.6, 1). Smaller k values indicate the small size of the synthesized snow, i.e., closer to the input clear image. The results show that the model produces the best quality output at the smallest k values.

Table 3. Results from quantitative model comparisons after eliminating various loss factors are presented. We generate images with three sets of k values. We examine the impact of the content discriminator, cross-cycle consistency loss, reconstruction losses, and style regression loss. Down arrows mean lower dVGG and FID values signify improved experimental performance, while up arrows mean that higher values for the remaining metrics indicate better results. Figures in bold highlight the best values.

Module	$SSIM \uparrow$	$PSNR \uparrow$	$d_{VGG} \downarrow$	$FID \downarrow$	$d_{LPIPS} \uparrow$
w/o $\mathcal{L}_{id}^{x}$	0.459	16.868	6.044	26.156	0.046
w/o $\mathcal{L}_{recon}^{c}$	0.461	16.831	6.052	26.133	0.045
w/o $\mathcal{L}_{recon}^{s}$	0.466	16.843	6.032	26.092	0.045
w/o $\mathcal{L}_{cc}$	0.462	16.857	6.063	26.112	0.046
w/o $\mathcal{L}_{cont}$	0.468	16.801	6.015	26.127	0.046
CUSS ($k = 1$)	0.471	16.912	6.003	26.089	0.047
CUSS ($k = 0.6$)	0.471	16.934	5.962	26.035	0.047
CUSS ($k = 0.3$)	**0.472**	**16.967**	**5.925**	**25.983**	**0.048**

4.3.4. Discussion

I2I translation methods such as CycleGAN, which uses the principle of cycle consistency, produce deterministic outputs. For a given clear image, it will produce the same translated snow image every time. To produce diverse outputs, researchers manipulate the latent space of extracted image features by dividing them into style codes and content codes. In our experiments, we found that latent space manipulation inevitably splits the translation network into two or more parts. This leads to performance degradation. In this work, the solution is to use a content discriminator to distinguish the content code from different domains. With the requirement of generating an indistinguishable content code, the encoder can achieve better disentangled representations.

When obtaining the disentangled style code, the operation on it will reflect on the output snow images [42]. Therefore, we interpolate the style codes of the input clear image and the snow image. Since the input snow image represents the maximum snow size, we can control the degree of snow effects. Note that we cannot obtain snow effects larger than the input image.

The controllable output is high quality and reasonable. The scenes are gradually covered with stronger snow effects. However, the generated snow is not invariant to objects. The snow covering the trees and the snow covering the building should be different; i.e., the snow effects should change appearance according to scenario changes. However, it looks similar in Figure 5. To improve CUSS, we need to consider the semantic information in the latent space.

Our method basically belongs to domain translation, which learns knowledge from the source domain and transfers it to the target domain. In the case of snow generation, the output will only contain a similar snow effect with input snowy images. To obtain more variety of snow like real snow scenes, we can add more snowy datasets that have different snow shapes and sizes. However, it will lead to mode collapse if there are too many modes in the GAN training process. In addition, the data should be collected in the same region to avoid large domain gaps, such as Asian driving scenes and European driving scenes.

5. Conclusions

The study presents an innovative approach to unsupervised snow synthesis, wherein a controllable method is introduced that incorporates latent space manipulation. To effectively separate the features of snow style and content, an additional content discriminator is incorporated along with a self-regression style coding module. To transition smoothly from clear to snow-affected images, a partial style cycle consistency loss is employed to refine the latent representation space. Furthermore, comparative analyses are conducted to comprehend the impact of each loss component or module within the model on the outcomes. When subjected to quantitative and qualitative evaluation, against various

techniques using the Cityscapes and EuroCity Persons datasets, our approach consistently produces diverse and high-quality traffic scenes under snowy conditions. Moving forward, our future research endeavors can be classified into two distinct paths:

- Expanding the proposed technique to tackle generation tasks in more demanding driving conditions, such as heavy rain, dense fog, nighttime, and strong light;
- Delving deeper into the relationship between generative methods and latent space manipulation for I2I translation tasks by integrating existing insights from self-supervised and contrast learning methodologies.

Author Contributions: Conceptualization, H.Y. and Y.Z.; methodology, H.Y.; formal analysis, A.C.; writing—original draft preparation, H.Y.; supervision, K.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Tokai National Higher Education and Research System (THERS) "Interdisciplinary Frontier Next Generation Researcher Scholarship" (RG211057), supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan and Nagoya University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, Y.; Carballo, A.; Yang, H.; Takeda, K. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS J. Photogramm. Remote Sens.* **2023**, *196*, 146–177. [CrossRef]
2. Ding, Q.; Li, P.; Yan, X.; Shi, D.; Liang, L.; Wang, W.; Xie, H.; Li, J.; Wei, M. CF-YOLO: Cross Fusion YOLO for Object Detection in Adverse Weather With a High-Quality Real Snow Dataset. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 10749–10759. [CrossRef]
3. Zhang, W.; Yao, R.; Du, X.; Liu, Y.; Wang, R.; Wang, L. Traffic flow prediction under multiple adverse weather based on self-attention mechanism and deep learning models. *Phys. A Stat. Mech. Its Appl.* **2023**, *625*, 128988. [CrossRef]
4. Qin, Q.; Chang, K.; Huang, M.; Li, G. DENet: Detection-driven Enhancement Network for Object Detection Under Adverse Weather Conditions. In Proceedings of the Asian Conference on Computer Vision, Macao, China, 4–8 December 2022; pp. 2813–2829.
5. Rothmeier, T.; Huber, W. Let it snow: On the synthesis of adverse weather image data. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; IEEE: New York, NY, USA, 2021; pp. 3300–3306.
6. Sakaridis, C.; Dai, D.; Van Gool, L. Semantic foggy scene understanding with synthetic data. *Int. J. Comput. Vis.* **2018**, *126*, 973–992. [CrossRef]
7. Garg, K.; Nayar, S.K. Detection and removal of rain from videos. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, Washington, DC, USA, 27 June–2 July 2004; IEEE: New York, NY, USA, 2004; Volume 1, p. I.
8. Liu, Y.F.; Jaw, D.W.; Huang, S.C.; Hwang, J.N. DesnowNet: Context-aware deep network for snow removal. *IEEE Trans. Image Process.* **2018**, *27*, 3064–3073. [CrossRef] [PubMed]
9. Zhang, K.; Li, R.; Yu, Y.; Luo, W.; Li, C. Deep dense multi-scale network for snow removal using semantic and depth priors. *IEEE Trans. Image Process.* **2021**, *30*, 7419–7431. [CrossRef] [PubMed]
10. He, K.; Sun, J.; Tang, X. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 2341–2353. [PubMed]
11. Engin, D.; Genç, A.; Kemal Ekenel, H. Cycle-dehaze: Enhanced cyclegan for single image dehazing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 825–833.
12. Guo, Y.; Ma, Z.; Song, Z.; Tang, R.; Liu, L. Cycle-Derain: Enhanced CycleGAN for Single Image Deraining. In Proceedings of the Big Data and Security: Second International Conference, ICBDS 2020, Singapore, 20–22 December 2020; Revised Selected Papers 2; Springer: Berlin/Heidelberg, Germany, 2021; pp. 497–509.
13. Kim, T.; Cha, M.; Kim, H.; Lee, J.K.; Kim, J. Learning to discover cross-domain relations with generative adversarial networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 1857–1865.

14. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
15. Yi, Z.; Zhang, H.; Tan, P.; Gong, M. Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2849–2857.
16. Yang, H.; Ming, D.; Alexander, C.; Zhang, Y.; Kento, O.; Yinjie, N.; Maoning, G.; Yan, F.; Kazuya, T. Synthesizing Realistic Snow Effects in Driving Images Using GANs and Real Data with Semantic Guidance. In Proceedings of the IEEE Intelligent Vehicles Symposium, Anchorage, AK, USA, 4–7 June 2023; p. 34.
17. Zhang, C.; Lin, Z.; Xu, L.; Li, Z.; Tang, W.; Liu, Y.; Meng, G.; Wang, L.; Li, L. Density-aware haze image synthesis by self-supervised content-style disentanglement. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 4552–4572. [CrossRef]
18. Biri, V.; Michelin, S. Real Time Animation of Realistic Fog. In Proceedings of the Poster Session of 13th Eurographic Workshop on Rendering, Berlin, Germany, 24–28 August 2002; pp. 9–16.
19. Ohlsson, P.; Seipel, S. Real-time rendering of accumulated snow. In Proceedings of the Sigrad Conference, Citeseer, Online, 24–25 November 2004; pp. 25–32.
20. Stomakhin, A.; Schroeder, C.; Chai, L.; Teran, J.; Selle, A. A material point method for snow simulation. *ACM Trans. Graph. (TOG)* **2013**, *32*, 1–10. [CrossRef]
21. Liu, M.Y.; Breuel, T.; Kautz, J. Unsupervised image-to-image translation networks. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
22. Huang, X.; Liu, M.Y.; Belongie, S.; Kautz, J. Multimodal unsupervised image-to-image translation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 172–189.
23. Pang, Y.; Lin, J.; Qin, T.; Chen, Z. Image-to-image translation: Methods and applications. *IEEE Trans. Multimed.* **2021**, *24*, 3859–3881. [CrossRef]
24. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
25. Zhu, J.Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A.A.; Wang, O.; Shechtman, E. Toward multimodal image-to-image translation. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
26. Lee, H.Y.; Tseng, H.Y.; Huang, J.B.; Singh, M.; Yang, M.H. Diverse image-to-image translation via disentangled representations. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 35–51.
27. Choi, Y.; Choi, M.; Kim, M.; Ha, J.W.; Kim, S.; Choo, J. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8789–8797.
28. Liu, A.H.; Liu, Y.C.; Yeh, Y.Y.; Wang, Y.C.F. A unified feature disentangler for multi-domain image translation and manipulation. *Adv. Neural Inf. Process. Syst.* **2018**, *31*.
29. Yang, H.; Carballo, A.; Takeda, K. Disentangled Bad Weather Removal GAN for Pedestrian Detection. In Proceedings of the 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, 19–22 June 2022; pp. 1–6.
30. Yang, H.; Carballo, A.; Zhang, Y.; Takeda, K. Framework for generation and removal of multiple types of adverse weather from driving scene images. *Sensors* **2023**, *23*, 1548. [CrossRef] [PubMed]
31. Wang, M.; Su, T.; Chen, S.; Yang, W.; Liu, J.; Wang, Z. Automatic Model-Based Dataset Generation for High-Level Vision Tasks of Autonomous Driving in Haze Weather. *IEEE Trans. Ind. Inform.* **2022**, *19*, 9071–9081. [CrossRef]
32. Ni, S.; Cao, X.; Yue, T.; Hu, X. Controlling the rain: From removal to rendering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 6328–6337.
33. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
34. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
35. Braun, M.; Krebs, S.; Flohr, F.; Gavrila, D.M. The eurocity persons dataset: A novel benchmark for object detection. *arXiv* **2018**, arXiv:1805.07193.
36. Demir, U.; Unal, G. Patch-based image inpainting with generative adversarial networks. *arXiv* **2018**, arXiv:1803.07422.
37. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 586–595.
38. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
39. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
40. Wu, H.; Qu, Y.; Lin, S.; Zhou, J.; Qiao, R.; Zhang, Z.; Xie, Y.; Ma, L. Contrastive learning for compact single image dehazing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10551–10560.

41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Higgins, I.; Amos, D.; Pfau, D.; Racaniere, S.; Matthey, L.; Rezende, D.; Lerchner, A. Towards a definition of disentangled representations. *arXiv* **2018**, arXiv:1812.02230.