Optimizing Generative Models for Monet-Style Image Synthesis in the Kaggle Competition “I’m Something of a Painter Myself”

Cameron Rader, Sujit Tripathy, Suneil Patel, Md Safaiat Hossain
Team 12

***Abstract*—This project explores the use of Generative Adversarial Networks (GANs) for artistic style transfer in the Kaggle competition “I’m Something of a Painter Myself”, aiming to generate Monet-style images. Our baseline approach was based on Amy Jang’s CycleGAN implementation with minor modifications, achieving a MiFID score of 62.31. To improve performance, we systematically tested several enhancements. First, we introduced ResNet-based generators in our baseline CycleGAN-based model, which resulted in a MiFID of 85.637. Next, we introduced an incremental change by modifying the adversarial loss, replacing the original binary cross-entropy (BCE) with mean squared error (MSE) as used in Least Squares GAN (LSGAN). However, this combination further degraded performance, increasing the MiFID to 93.64, indicating reduced perceptual quality. We also implemented a variant of CUT, an alternative architecture leveraging contrastive learning, which achieved a MiFID of 66.47. Overall, these modifications such as architectural refinements, loss function adjustments, and adopting a different framework did not outperform the baseline. This suggests that the baseline implementation was already highly optimized.**

***Index Terms*—CycleGAN, Monet, CUT, MiFID**

I. Introduction

This project aims to generate high-quality images in the style of Claude Monet for the Kaggle competition “I’m Something of a Painter Myself” with a primary objective of improving a baseline developed based on Amy Jang’s CycleGAN implementation [1]. The study explores a range of enhancements to more effectively capture Monet’s distinctive artistic style while maintaining structural coherence and visual realism. To accomplish this, several strategies are investigated, including modifying the existing architectures, adjustment of loss functions, and experimentation with novel architectures.

II. Baseline Model

A. Baseline Implementation

This project builds on Amy Jang’s Monet CycleGAN tutorial from the Kaggle competition “I’m Something of a Painter Myself” [1]. CycleGAN addresses the unpaired image-to-image translation problem by learning bidirectional mappings between two domains without requiring paired data. The architecture consists of two generators (Photo$\rightarrow$Monet and Monet$\rightarrow$Photo) and two PatchGAN discriminators, which output patch-level real/fake predictions to capture local texture details. Generators use a U-Net-based model with instance normalization, while discriminators employ LeakyReLU. This design provides a strong foundation for generating Monet-style images while preserving structural content [1].

B. Training and Loss Functions

Training combines adversarial loss (sigmoid + BCE with logits), cycle-consistency loss (L1 penalty weighted by $\lambda_{\text{cycle}} = 10$), and identity loss (weighted at $\lambda_{\text{id}} \leq 0.5 \times \lambda_{\text{cycle}}$) to balance realism and content fidelity. The total objective encourages generators to produce convincing outputs while maintaining original structure and color consistency. Models are optimized using the Adam optimizer, and training alternates between updating generators and discriminators [1]. To establish our baseline, we made minor modifications to Amy Jang’s original implementation while keeping the other hyperparameter settings unchanged (please see appendix to check hyperparameter values). Specifically, we reduced the number of training epochs from 100 to 50 and switched the input format from .tfrec to .jpeg to simplify data handling.

C. Baseline Performance

Figure 1 shows the the mean generator and discriminator losses averaged over an epoch during the baseline training. The plot indicates that the training of the generator (Photo$\rightarrow$Monet) was more stable compared to the generator (Monet$\rightarrow$Photo). Figure 2 shows a sample Monet-style image generated by the generator (Photo$\rightarrow$Monet) after training. Running the modified baseline CycleGAN implementation on Kaggle produced a MiFID score of **62.31** (29th on the Kaggle leaderboard when writing this report) as listed in Table I. This score serves as the reference point for evaluating all subsequent enhancements and experiments.

III. Enhancements to Baseline Model

After establishing the CycleGAN implementation as a stable and reproducible baseline for Monet-style image generation, a series of targeted enhancements were explored to improve model performance. The primary objective was to systematically investigate how modifications to network architecture and loss functions could impact the quality of generated images, as measured by the MiFID score. Each enhancement was informed by theoretical considerations or best practices identified in the literature, and was evaluated individually or

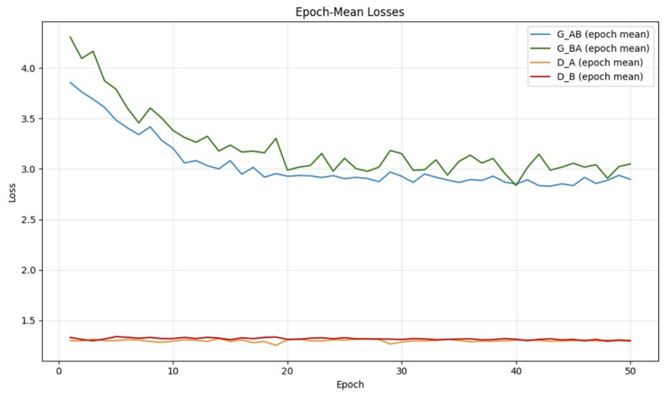


Fig. 1: Mean generator and discriminator losses vs epoch obtained during baseline training.

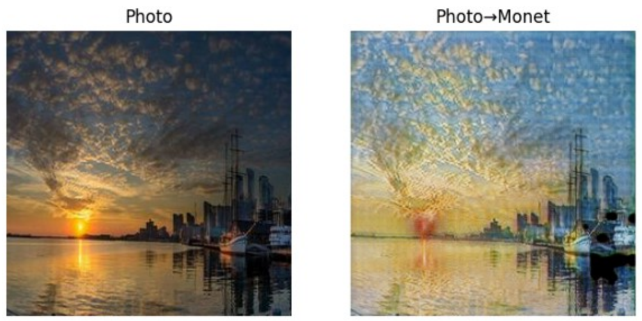


Fig. 2: Example Monet-style image generated by baseline.

in combination to clearly determine its effect relative to the baseline [2], [3].

Each experiment involved training the model, followed by evaluation on Kaggle with the new configuration. MiFID scores were then recorded to quantify any improvements or regressions resulting from the changes. The following sections present each enhancement or combination of enhancements, the rationale for its selection, the observed results, and an analysis of its effectiveness. This structured methodology yielded insights into the factors most critical for advancing Monet-style image synthesis within the competition context.

A. Implementing ResNet based Generators with 9 residual blocks

a) Description: ResNet-based generators are widely employed in image-to-image translation and style transfer tasks due to their capacity to learn complex transformations while preserving the structural integrity of the input image. Through the use of residual blocks with skip connections, these generators effectively model transformations that approximate the identity function, thereby enhancing training stability and supporting the development of deeper network architectures. This architectural approach enables the network to capture global modifications, including color shifts, texture changes, and artistic patterns, which are critical for successful style adaptation [2].

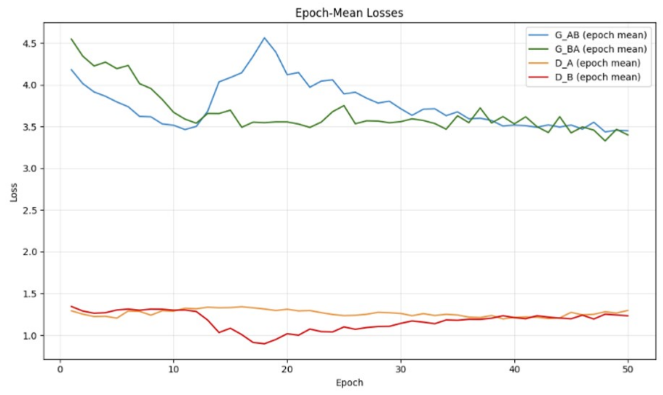


Fig. 3: Mean generator and discriminator losses vs epoch obtained during training of the baseline enhanced with ResNet-based generators.

The ResNet-based generator in our implementation was constructed by first applying a 7×7 convolutional layer to the input image, followed by two downsampling layers that reduced spatial dimensions and increased feature depth. The core of the network comprised nine residual blocks, each containing two convolutional layers and a skip connection that added the block's input to its output. After the residual blocks, two upsampling layers restored the features to the original image size. A final convolutional layer with a tanh activation produced the stylized output in the range [-1, 1], balancing content preservation with the capacity to learn rich, domain-specific styles [2], [4]. Please refer to the appendix for detailed layer-wise information on the ResNet-based generator architecture.

b) Result: As can be seen in Figure 3, the training of the ResNet-based generators was more unstable compared to the baseline, which prompted us to consider MSE loss, based on LSGAN [3], as the next enhancement to help stabilize training. Additionally, as shown in Figure 4, the quality of the Monet-style images generated was noticeably worse, and the MiFID score increased to 85.637, which is higher than the baseline score of 62.31 as listed in Table I. These observations motivated further adjustments to the adversarial loss function in an effort to improve perceptual quality.

c) Discussion: The reduced performance observed with the ResNet-based generator compared to the U-Net was likely due to several factors. The skip connections in U-Net would facilitate the preservation of finer spatial details, which often would lead to sharper and more realistic outputs. In contrast, ResNet generators might have lost some of these details if the residual blocks emphasized global transformations. Keeping same training duration and hyperparameter settings also could have influenced outcomes, as ResNet architectures sometimes require extended training or alternative loss weightings to achieve optimal results. Furthermore, the network's depth and capacity, along with choices in normalization and initialization, significantly could affect GAN stability and output quality. The appropriateness of each architecture ultimately

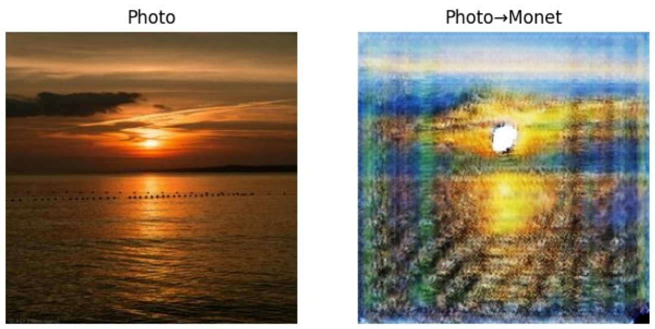


Fig. 4: Example Monet-style image generated by baseline enhanced with ResNet-based generators.

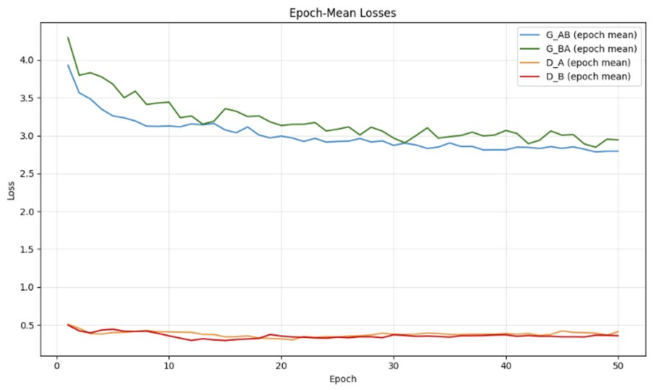


Fig. 5: Mean generator and discriminator losses vs epoch obtained during training of the baseline enhanced with ResNet-based generators and MSE loss function.

depends on the specific application, with U-Net typically preferred for tasks demanding precise content retention and ResNet demonstrating strengths in style adaptation.

B. Switching Adversarial Loss: BCE $\rightarrow$ MSE (LSGAN)

a) Description: The adversarial loss function was modified from BCE to MSE following the LSGAN methodology [3]. This adjustment aims to provide smoother gradients enhancing the stability of GAN training. The modification was implemented for both discriminator and generator adversarial losses, while the cycle-consistency and identity loss functions remained unchanged. The model was trained for 50 epochs using the generator architecture with 9 residual blocks.

b) Result: As shown in Figure 5, training with the LSGAN (MSE adversarial loss) was more stable compared to the baseline enhanced with ResNet-based generators. However, as illustrated in Figure 6, the quality of the generated Monet-style images was noticeably worse, which was reflected in a higher MiFID score of 93.64, as reported in Table I. This result indicates that, despite improved training stability, the perceptual quality of the outputs declined compared to both the baseline and the ResNet generator enhancement.

c) Discussion: Despite the theoretical advantages of LSGAN, the MiFID score increased. MSE loss can produce smoother but weaker gradients when the discriminator is too strong, causing slower convergence and less effective generator updates. Additionally, hyperparameters set for BCE might not be well-suited for MSE, further impacting performance. Without retuning, the generator may produce less diverse or lower-quality outputs, as shown by the higher MiFID score.

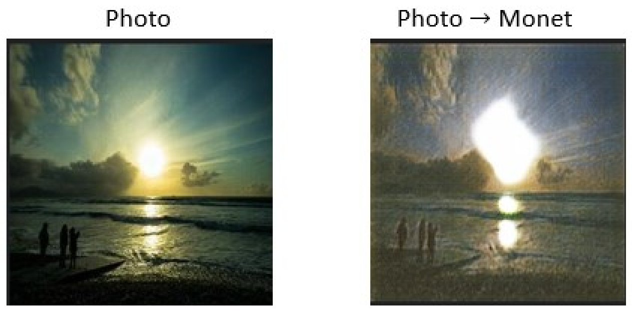


Fig. 6: Example Monet-style image generated by baseline enhanced with ResNet-based generators and MSE loss function.

IV. New Architecture Approach: CUT Variant

This variant is a reimplementation of the CUT (Contrastive Unpaired Translation) framework [4] with additional stability and optimization improvements.

A. Motivation

Rather than incrementally modifying CycleGAN, we pursued a fundamentally different approach that removes the need for a reverse generator and cycle consistency loss. CUT uses contrastive learning [4] (PatchNCE) to preserve image structure while transferring style, enabling faster training, higher batch sizes, and stronger content retention.

B. Core Architectural Differences

- Single generator (Photo $\rightarrow$ Monet) instead of two CycleGAN generators.
- Single PatchGAN discriminator rather than one per domain.
- PatchNCE contrastive loss replaces cycle-consistency loss.
- Uses ResNet-9 architecture instead of U-Net.
- Incorporates modern training enhancements: EMA, DiffAugment, R1 regularization, mixed precision, and cosine LR scheduling.

C. Model Components

a) Generator Design: The generator follows a ResNet-9 architecture, which includes two downsampling layers to reduce spatial dimensions, nine residual blocks for feature transformation, and two upsampling layers to restore the image to its original size. This design closely follows the architectural enhancements previously discussed for CycleGAN. Reflection padding and instance normalization are used throughout. Intermediate features from multiple layers are extracted to support PatchNCE contrastive learning [4].

b) Discriminator Design: A single PatchGAN discriminator classifies local patches rather than whole images. It uses a sequence of 4×4 convolutions with LeakyReLU activations. Multiscale and spectral normalization support exist in the codebase but remain disabled in the initial configuration.

c) Loss Functions: The model uses hinge adversarial loss for stability, PatchNCE contrastive loss [4] to enforce feature correspondence between input and generated patches, and a limited identity warmup loss that is phased out early in training. R1 gradient penalty is applied lazily to regularize the discriminator.

d) Training Strategy: Training uses the Adam optimizer with cosine learning rate scheduling, DiffAugment for differentiable augmentation, gradient clipping for both networks, and mixed precision for speed and memory efficiency. EMA maintains a smoothed generator used for inference. Batch size is significantly larger than CycleGAN (e.g., 12) due to the single generator design.

D. Results andDiscussion

To evaluate the performance of our CUT variant, we tracked training dynamics, submitted our generated images to Kaggle for MiFID scoring, and inspected qualitative examples.

a) Training Behavior: Figure 7 shows both the raw and smoothed discriminator and generator losses over 40k training steps. The discriminator loss remained stable near its expected hinge-loss equilibrium, while the generator loss gradually decreased with periodic spikes corresponding to R1 regularization steps. Overall, the loss curves indicate stable adversarial training without collapse.

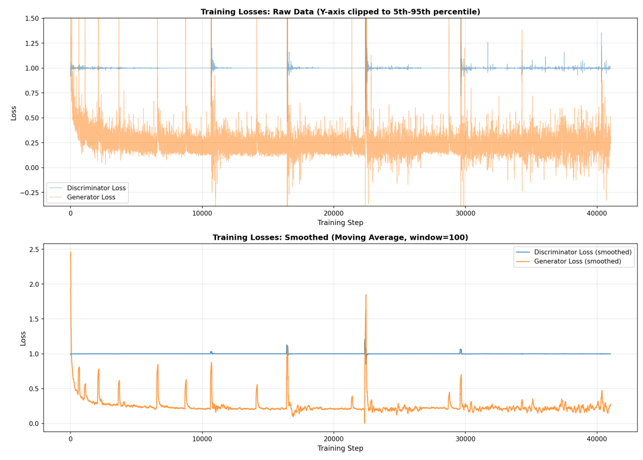


Fig. 7: Training losses for the CUT model across 40k steps (raw and smoothed).

b) Quantitative Evaluation: Our CUT variant achieved a **MiFID score of 66.47** as listed in Table I. At the time of writing this report, this submission was ranked **36th overall** on the leaderboard in the Kaggle competition "I'm Something of a Painter Myself."

c) Qualitative Results: Figure 8 displays a representative Monet-style output produced by our model. The CUT generator captures the global color palette and brushstroke texture characteristic of Monet paintings, while preserving the high-level structure of the input photograph.

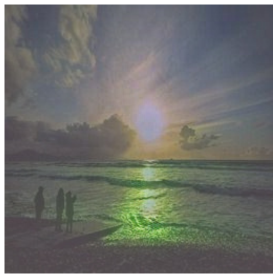


Fig. 8: Example Monet-style image generated by our CUT variant.

TABLE I: MiFID scores and delta compared to baseline for different model modifications.

Model / Modification	**MiFID**	**Δ**
Baseline (Amy Jang CycleGAN)	62.31	0.00
ResNet Generator	85.64	+23.33
LSGAN (MSE Loss)	93.64	+31.33
CUT Variant	66.47	+4.16

V. Challenges and Lessons Learned

In this project, we evaluated several architectural and training modifications to a CycleGAN-based baseline for Monet-style image synthesis. Although our experiments with ResNet generators, LSGAN adversarial loss, and a custom CUT variant offered valuable insights into the sensitivity of generative models to architectural and loss-function changes, none of the tested enhancements surpassed the performance of the original baseline. The CUT variant achieved competitive results, placing 36th on the Kaggle leaderboard with a MiFID score of 66.47 (at the time of writing the report), demonstrating strong structural preservation and stylistic transfer despite removing cycle consistency. Overall, our findings suggest that the baseline implementation was already highly optimized for this competition, and that further improvements may require more extensive hyperparameter tuning, larger training budgets, or hybrid architectural approaches beyond incremental GAN modifications.

References

[1] A. Jang, "Monet CycleGAN tutorial," *Kaggle Notebook*, 2020. [Online]. Available: https://www.kaggle.com/code/amyjang/monet-cyclegan-tutorial

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. https://arxiv.org/abs/1703.10593

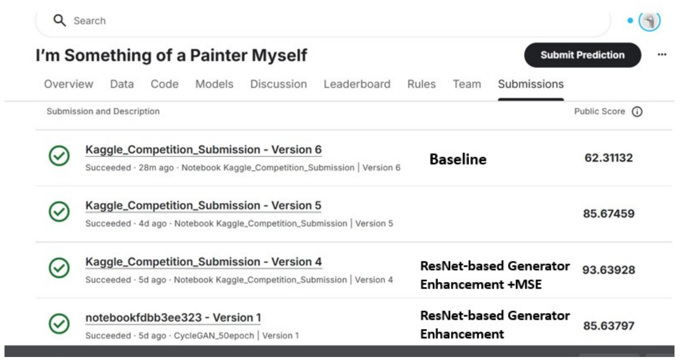
[3] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. https://arxiv.org/abs/1611.04076

[4] T. Park, A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive Learning for Unpaired Image-to-Image Translation," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 319–345.

Contribution Statement

Name	Contribution
Cameron Rader	Implemented the CUT-GAN variant, conducted experiments, generated Kaggle submissions, report writing.
Sujit Tripathy	Literature review,modified Amy's code to implement our baseline and enhancements to baseline, conducted experiments, generated Kaggle submissions, report writing.
Suneil Patel	Contributed to the literature review, helped Sujit implement baseline model, and report preparation and review.
Md Safaiat Hossain	Contributed to the literature review, assisted Cameron with CUT-GAN implementation and experiments, and supported report preparation and review.

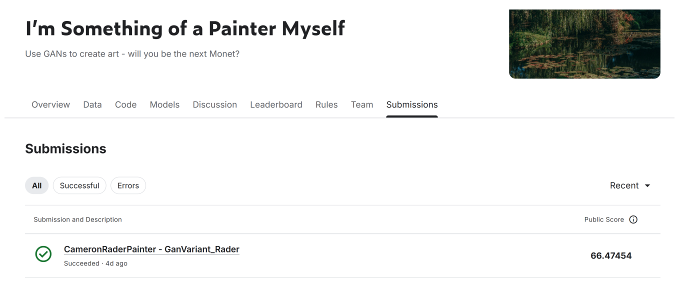
Appendix


Search

I'm Something of a Painter Myself

Submission and Description		Public Score
Kaggle_Competition_Submission - Version 6 Succeeded · 28m ago · Notebook Kaggle_Competition_Submission | Version 6	Baseline	62.31132
Kaggle_Competition_Submission - Version 5 Succeeded · 4d ago · Notebook Kaggle_Competition_Submission | Version 5		85.67459
Kaggle_Competition_Submission - Version 4 Succeeded · 5d ago · Notebook Kaggle_Competition_Submission | Version 4	ResNet-based Generator Enhancement +MSE	93.63928
notebookfdbb3ee323 - Version 1 Succeeded · 5d ago · CycleGAN_50epoch | Version 1	ResNet-based Generator Enhancement	85.63797

Fig. 9: Submission results for the baseline and its enhancements in "I'm Something of a Painter Myself" Kaggle competition.


I'm Something of a Painter Myself

Use GANs to create art - will you be the next Monet?

Submissions

Submission and Description	Public Score
CameronRaderPainter - GanVariant_Rader Succeeded · 4d ago	66.47454

Fig. 10: Submission result for the CUT variant in "I'm Something of a Painter Myself" Kaggle competition.

TABLE A.I: Key hyperparameters and their descriptions for CycleGAN baseline training.

Hyperparameter	Description
IMG_SIZE = 256	Input image size (pixels)
BATCH_SIZE = 1	Batch size (small for CycleGAN stability)
BUFFER_SIZE = 1024	Shuffle buffer for dataset loading
EPOCHS = 50	Number of training epochs
LR_G = 2e-4	Generator learning rate
LR_D = 2e-4	Discriminator learning rate
LAMBDA_CYCLE = 10.0	Cycle consistency loss weight
LAMBDA_ID = 0.5	Identity loss weight ($\leq 0.5 \times$ cycle weight)

TABLE A.II: Layer-wise structure of the ResNet-based generator used to enhance the baseline.

Layer	Output Size	Channels
Input	256×256	3
Conv2D (7×7, 64)	256×256	64
Downsample (128)	128×128	128
Downsample (256)	64×64	256
Residual Blocks ×9	64×64	256
Upsample (128)	128×128	128
Upsample (64)	256×256	64
Conv2D (7×7, 3)	256×256	3

TABLE A.III: Key hyperparameters and their descriptions for CUT variant training.

Hyperparameter	Description
IMG_SIZE = 256	Input image size (pixels)
BATCH_SIZE = 12	Batch size
EPOCHS = 70	Number of training epochs
LR_G = 2e-4	Generator learning rate
LR_D = 2e-4	Discriminator learning rate
LAMBDA_ADV = 1.0	Adversarial (hinge) loss weight
LAMBDA_PATCHNCE = 1.0	PatchNCE contrastive loss weight
LAMBDA_ID_WARM = 0.1	Identity loss weight during warmup
R1_GAMMA = 10.0	R1 regularization penalty weight
R1_EVERY = 16	R1 regularization frequency
PATCHNCE_TEMPERATURE = 0.07	Temperature scaling for PatchNCE
EMA_DECAY = 0.999	Exponential moving average decay
GRAD_CLIP = 10.0	Maximum gradient norm for clipping
WARMUP_STEPS = 20000	Steps before disabling identity loss