



PPO-Guided Adaptive Loss Balancing for Object Inpainting in Mobile Mapping System

Luepol Pipanmekaporn¹, Suwatchai Kamonsantiroj²

¹Department of Computer and Information Science, King Mongkut's University of Technology North Bangkok, Bangkok, 10800, Thailand.
E-mail: luepol.p@sci.kmutnb.ac.th

²Department of Computer and Information Science, King Mongkut's University of Technology North Bangkok, Bangkok, 10800, Thailand.
E-mail: suwatchai.k@sci.kmutnb.ac.th

Abstract

Mobile Mapping Systems (MMS) capture spherical panoramic images that often contain privacy-sensitive objects, such as vehicles. Removing these objects and plausibly background reconstruction remains a challenging inpainting task due to the complex and highly structured nature of street-level scenes. Although Generative Adversarial Networks (GANs) such as Pix2Pix have shown potential for object inpainting, their performance is highly sensitive to the weighting of competing loss terms, which are typically tuned manually and kept fixed throughout training. To address this limitation, we propose an adaptive loss balancing framework that leverages Proximal Policy Optimization (PPO), a reinforcement learning algorithm, to dynamically adjust the loss weight coefficients of Pix2Pix during training. At each training epoch, the PPO agent observes a state vector composed of running averages of generator and discriminator losses alongside image quality metrics, and outputs updated loss weights to guide the generator toward perceptually realistic reconstructions. The agent is trained using six reward signals derived from pixel-level metrics, including PSNR and SSIM and perceptual quality metrics, LPIPS and FID, to jointly capture structural fidelity and visual realism. Experiments on real-world MMS datasets demonstrate that the PPO-based framework consistently outperforms baselines across all evaluated metrics, achieving gains of up to +16.86% over the fixed-weight baseline and up to +10.2% over the adaptive-weight baseline.

Keywords: Object Inpainting, Adaptive Loss Balancing, PPO, Reinforcement Learning, GAN

1. Introduction

Mobile Mapping Systems (MMS) are vehicle-mounted platforms equipped with panoramic cameras and Global Navigation Satellite System (GNSS) receivers that enable efficient acquisition of large-scale street-level geospatial imagery (Elhashash et al., 2022; Petrie and Toth, 2009). The resulting 360-degree equirectangular images are widely used in digital map services, virtual reality applications, and urban planning. However, the practical use of MMS imagery is hindered by a persistent issue: captured scenes often include unwanted objects, especially the survey vehicle, which typically appears as a distorted region in lower part of the equirectangular image, as shown in Figure 1. In addition, MMS images often include privacy-sensitive objects such as license plates, faces, and other personally identifiable information, which must be removed before the images are publicly released. This requirement makes object inpainting a critical preprocessing step in any MMS data pipeline.



Figure 1. Mobile Mapping Panoramas with the survey vehicle occlusion.

Object inpainting, defined as the task of reconstructing missing or masked regions to be visually coherent with their surrounding context, has advanced considerably over the past decade. Early patch-based methods (Mysore Badarinarayana et al., 2025; Huo et al., 2019; Darabi et al., 2012) address this problem by copying textures from surrounding known regions into the missing area. Although these methods are effective for simple and repetitive backgrounds, they often struggle when the target region contains semantically complex structure, such as road surfaces, road marking and lane lines, which are commonly found in street-level imagery. Recently, deep learning-based object inpainting approaches (Yu et al., 2024; Suvorov et al., 2022; Yu et al., 2019; Yu et al., 2018) have been developed to synthesize missing background regions by learning to predict realistic content from the global image context. Among these methods, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have shown particularly effective. For example, Pix2Pix (Isola et al., 2017), a conditional GAN (cGAN) framework that uses a U-Net generator and a PatchGAN discriminator for paired image-to-image translation, has become a widely used method for inpainting tasks (Mansourifar & Simske, 2023; Christovam et al., 2022). This is because Pix2Pix can learn a direct mapping from an input image to its corresponding ground-truth output using paired training data, enabling the model to consistently reconstruct missing regions based on the surrounding contextual structure.

Despite its success, standard Pix2Pix, which mainly relies on adversarial loss and L_1 loss, often produces smooth or blurry results because it focuses on pixel-level accuracy rather than realistic details. To improve visual quality, perceptual loss and style loss have been widely used to capture high-level structures and texture patterns. However, adding more loss terms creates a new challenge: multiple loss terms must be properly balanced. In practice, these loss weights are usually set manually through trial and error, which is time-consuming, dataset-dependent, and results in fixed loss weights that do not adapt during training. Conversely, adaptive loss weighting strategies have been proposed. For instance, SoftAdapt (Heydari et al., 2019) dynamically reassigns loss weights based on the relative rate of change of each loss component, improving training stability without exhaustive grid search. Similarly, GradNorm (Chen et al., 2018) adaptively balances multiple objectives by regulating their gradient magnitudes and relative training rates. However, both methods primarily respond to optimization dynamics rather than directly evaluating the perceptual quality of generated images. Consequently, their weighting strategies may be suboptimal for image restoration tasks in which structural fidelity and perceptual realism are critical.

To overcome these limitations, we formulate loss weight adjustment as a decision-making problem and use Proximal Policy Optimization (PPO) (Schulman et al., 2017), a reinforcement learning algorithm. During training, the PPO agent observes the current training state, including generator and discriminator losses as well as image quality scores, and updates the loss weights at each epoch. To guide learning, the agent is optimized using reward signals derived from image quality metrics, including pixel-based measures such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) (Hore & Ziou, 2010), and perceptual measures such as Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018) and Fréchet Inception Distance (FID) (Heusel et al., 2017). This strategy directly aligns loss weight selection with image quality objectives, enabling fully data-driven and adaptive optimization of Pix2Pix training process. Experiments on MMS datasets demonstrate that the PPO-based framework consistently outperforms baselines across all evaluated metrics, achieving gains of up to +16.86% over the fixed-weight baseline and up to +10.2% over the adaptive-weight baseline.

The main contributions of this paper are as follows:

- We introduce a PPO-based reinforcement learning framework that automatically and dynamically adjusts the loss coefficients to enhance object inpainting performance in MMS imagery.
- We design and evaluate six reward formulations based on image quality signals, including pixel-level measures and perceptual measures and analyze how different reward signals influence the learned policy and the resulting inpainting quality.
- Experiments on real-world MMS data show that the proposed PPO-based framework consistently outperforms fixed-weight and adaptive weight baselines across all evaluation metrics, demonstrating that combining pixel-level and perceptual reward signals leads to better quality inpainting results.

2. Related Works

2.1 Image inpainting in street-view imagery

Object inpainting in street-level imagery presents unique challenges that distinguish it from standard inpainting settings. First, street-level images are typically captured at very high resolutions (e.g. 4096×2048). Second, they are stored in an equirectangular format with a 2:1 aspect ratio, introducing severe geometric distortions. Third, street-level scenes further compound this difficulty, as they contain highly structured

elements, such as road surfaces, lane markings, and building facades, where even minor geometric inconsistencies produce perceptually salient artifacts.

Early inpainting methods addressed missing-region reconstruction through patch-based and exemplar-driven strategies. The foundational approach of Huo et al., 2019) propagates texture from surrounding known regions into the target area, proving effective for repetitive patterns but inadequate when the missing region is large or semantically complex. The advent of deep learning shifted the dominant paradigm toward encoder-decoder architectures and Generative Adversarial Networks (GANs) (Yu et al., 2019; Yu et al. 2018), enabling models to learn semantic priors and synthesize contextually coherent content rather than relying solely on texture replication. Subsequent CNN-based methods extended this capability to large-masked regions; notably, LaMa (Suvorov et al., 2022) employed fast Fourier convolutions to enlarge the effective receptive field and substantially improve reconstruction quality under challenging large-mask conditions. More recently, transformer-based architectures have attracted considerable attention for inpainting due to their capacity to model long-range spatial dependencies through self-attention (Li et al., 2022; Lugmayr et al., 2022). In contrast to convolutional approaches, transformers can capture global scene continuity, which is particularly advantageous for reconstructing extended structures such as lane boundaries or building edges. However, their high computational demands make direct application to the high-resolution panoramic images typical of MMS datasets practically challenging.

Despite these advances, directly adapting existing inpainting frameworks to MMS imagery remains non-trivial. Unlike standard benchmarks such as COCO or Places2, MMS panoramas exhibit strong equirectangular distortions that violate the planar assumptions underlying conventional convolution operations. Beyond geometric fidelity, MMS applications impose strict privacy constraints, requiring the reliable removal of sensitive objects, including vehicles, license plates, and pedestrians, while preserving the photorealism and metric usability of the reconstructed background. These requirements call for inpainting frameworks that can jointly address high-resolution panoramic distortion, structural regularity in street scenes, and perceptual quality, demands that existing general-purpose methods do not fully satisfy.

2.2. Multi-Loss balancing in GAN

Training GANs for image synthesis often requires optimizing multiple loss terms, where the relative weighting significantly affects reconstruction quality. For instance, in the standard Pix2Pix framework, the generator is trained using adversarial loss and L_1 loss. While the L_1 term helps maintain structural consistency, it often produces smooth or blurry outputs because pixel-wise optimization tends to favor averaged predictions rather than sharp textures.

To enhance perceptual realism, perceptual loss (Johnson et al., 2016) and style loss (Gatys et al., 2015), derived from intermediate feature representations of a pretrained VGG network, have been widely adopted to improve texture detail and visual sharpness. However, combining adversarial, L_1 , perceptual, and style losses introduces a multi-objective optimization problem, as these loss terms may conflict and produce competing gradients. Consequently, inpainting performance becomes highly sensitive to the selected loss weights. In practice, these coefficients are typically determined through expensive manual tuning and remain fixed throughout training, even though GAN training dynamics evolve over time.

To reduce the burden of manual weight selection, several adaptive loss weighting methods have been proposed. SoftAdapt (Heydari et al., 2019), for example, adjusts loss weights based on the relative rate of change of each loss component, improving training stability without exhaustive hyperparameter search. Similarly, gradient-based strategies such as GradNorm (Chen et al., 2018) balance the gradient magnitudes of different loss terms to prevent any single objective from dominating the optimization. While these approaches improve stability, they remain largely heuristic and focus on loss dynamics rather than directly optimizing perceptual image quality.

Overall, most existing adaptive loss weighting strategies focus on balancing loss magnitudes, convergence speed, or gradient contributions, rather than explicitly optimizing perceptual image quality. As a result, the learned loss weights may improve training stability, but do not necessarily lead to visually realistic reconstructions. This limitation motivates reinforcement learning-based approaches, where loss weight adjustment is formulated as a decision-making process and guided directly by image-quality reward signals, enabling the training objective to better align with perceptual evaluation criteria.

3. Preliminaries

3.1 Pix2Pix

Pix2Pix (Isola et al., 2017) is a conditional GAN (cGAN) framework designed for paired image-to-image translation tasks, in which the generator learns a mapping from an input x to an output image y conditioned on x itself, rather than on random noise alone. Unlike standard GANs, the conditioning signal ensures that the

generated output is spatially coherent with the input, making Pix2Pix particularly well-suited for structured synthesis tasks such as object inpainting, where the reconstructed region must remain consistent with its surrounding context. The Pix2Pix generator adopts a U-Net architecture, consisting of a contracting encoder path and a symmetric expanding decoder path connected by skip connections. The encoder progressively downsamples the input to capture high-level semantic context, while the decoder upsamples back to the original resolution. The skip connections concatenate feature maps from each encoder layer directly to the corresponding decoder layer, preserving fine-grained spatial details, such as edges and textures that would otherwise be lost during downsampling. This multi-scale feature aggregation enables the generator to produce outputs that are both semantically plausible and structurally faithful to the ground truth.

Pix2Pix introduces a specialized discriminator called PatchGAN, which evaluates individual $N \times N$ patches of the image to find whether each is real or generated. This patch-level evaluation enables the model to focus on local textures and fine-grained details, significantly improving the perceptual quality of generated images. The total generator loss in Pix2Pix combines the adversarial loss with a L_1 loss (or reconstruction loss), encouraging the generated output to be structurally similar to the target image:

$$L_{Pix2Pix} = L_{adv}(G, D) + \alpha_{L1} L_{L1}(G) \quad (1)$$

where $L_{L1}(G) = \mathbb{E}_{x,y} \|y - G(x)\|_1$ is the L_1 loss between the generated image $G(x)$ and the target image y , α_{L1} is a hyperparameter controlling the weight of the L_1 loss (typically $\alpha_{L1} = 100$). With PatchGAN, the discriminator outputs a grid of real/fake decisions. Let N_p be the number of patches and $D_i(\cdot)$ the i th patch score. The conditional adversarial loss is defined as:

$$L_{adv}(G, D) = \mathbb{E}_{x,y} \left[\sum_{i=1}^{N_p} \log D(x_i, y_i) \right] + \mathbb{E}_x \left[\sum_{i=1}^{N_p} \log \left(1 - D(x, G(x_i, G(x_i))) \right) \right] \quad (2)$$

where N_p is the total number of patches in the image, and (x_i, y_i) , $(x_i, G(x_i))$ refer to corresponding patches from the input-target pair and input-generated pair. The discriminator is trained to maximize L_{adv} over patch outputs.

3.2. Perceptual and Style losses

The success of GAN-based inpainting depends on the careful design of loss functions, guiding the model generate realistic and coherent outputs. Among these losses, perceptual loss (Johnson et al., 2016) plays an important role by ensuring visual consistency and aligning high-level features between the ground truth I_{gt} and the generated output I_{out} . This loss is computed a pre-trained VGG-19 network, which extracts deep feature representations from multiple network layers. The perceptual loss is defined as:

$$L_{percept} = \sum_{p=0}^{P-1} \frac{\|f_p(I_{out}) - f_p(I_{gt})\|_1}{N(f_p(I_{gt}))} + \sum_{p=0}^{P-1} \frac{\|f_p(I_{ov}) - f_p(I_{gt})\|_1}{N(f_p(I_{gt}))} \quad (3)$$

where $f_p(I)$ denotes the feature map extracted from the p -th layer of a pretrained VGG network for input image I , and $N(f_p(I_{gt}))$ is a normalization factor based on the number of elements in that feature map. I_{ov} denotes the overlay output, which is the model's prediction projected onto the original input image I_{in} . By minimizing this loss, the network learns to align the perceptual difference between generated and real images.

Style loss (Gatys et al., 2015) encourages the texture and visual patterns in the generated inpainted regions to harmonize with the overall style of the ground truth image. Like perceptual loss, style loss is computed using a pre-trained VGG network, which extracts feature maps corresponding to textures and color patterns from both the generated images I_{out} and the ground truth I_{gt} . The style loss can be defined as follow:

$$L_{style} = \sum_{p=0}^{P-1} \frac{1}{C_p^2} \left\| K_p \left(f_p(I_{out})^T f_p(I_{out}) - f_p(I_{gt})^T f_p(I_{gt}) \right) \right\|_1 + \sum_{p=0}^{P-1} \frac{1}{C_p^2} \left\| K_p \left(f_p(I_{ov})^T f_p(I_{ov}) - f_p(I_{gt})^T f_p(I_{gt}) \right) \right\|_1 \quad (4)$$

where $f_p(I)^T f_p(I)$ denotes the Gram matrix derived from the feature map from the p -th layer of VGG-19 network for an input image I . The normalization factor $K_p(\cdot)$ is calculated as $1 / C_p H_p W_p$, where C_p is the number of channels of feature maps from the p th layer.

3.3. Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) (Schulman et al., 2017; Sutton & Barto, 2018) is an on-policy, actor-critic reinforcement learning algorithm that learns a policy $\pi_\theta(\mathbf{a}_t | \mathbf{s}_t)$ to maximize the expected discounted cumulative reward: $J(\theta) = \mathbb{E}_{\pi_\theta} [\sum_{t=0}^T \gamma^t r_t]$, where $\gamma \in (0, 1)$ is the discount factor. At each time step t , the agent observes state $\mathbf{s}_t$, selects action $\mathbf{a}_t$ according to π_θ and receives a scalar reward r_t from the environment.

PPO consists of two networks: an Actor $\pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t)$ that outputs a distribution over actions, and a Critic $V_{\phi}(\mathbf{s}_t)$ that estimates the expected return from the current state. The advantage of each action is estimated using the Generalized Advantage Estimate (GAE), $\hat{A}_t = \sum_{k=0}^{T-t-1} (\gamma \xi)^k \delta_{t+k}$, where $\delta_t = r_t + \gamma V_{\phi}(\mathbf{s}_{t+1}) - V_{\phi}(\mathbf{s}_t)$ is the temporal difference residual and $\xi \in [0, 1]$ controls the bias-variance trade-off (Schulman et al., 2016). The core contribution of PPO is a clipped surrogate objective that consists the policy update ratio $\rho_t(\theta) = \pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t) / \pi_{\theta_{\text{old}}}(\mathbf{a}_t | \mathbf{s}_t)$ within $[1 - \epsilon, 1 + \epsilon]$:

$$L^{\text{PPO}}(\theta, \phi) = E_t [\min(\rho_t(\theta) \hat{A}_t, \text{clip}(\rho_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t)] - c_1 E_t [(V_{\phi}(\mathbf{s}_t) - \hat{A}_t)^2] + c_2 H[\pi_{\theta}(\cdot | \mathbf{s}_t)] \quad (5)$$

where ϵ is the clipping threshold, $\hat{R}_t$ is the observed discounted return, and c_1, c_2 are coefficients weighting the Critic regression loss and entropy bonus, respectively. The clipping mechanism ensures that each policy update remains conservative, making PPO well-suited to continuous hyperparameter control tasks where abrupt parameter shifts could destabilize the underlying training process.

3.4. Problem formulation

Training Pix2Pix with multiple loss terms requires assigning a set of scale weight coefficients $\lambda = \{\lambda_k\}_{k=1}^K$, which regulate the relative contribution of each loss component L_k to the overall generator objective:

$$L_G = L_{\text{adv}}(G, D) + \sum_{k=1}^K \lambda_k L_k \quad (6)$$

where K denotes the number of auxiliary loss terms, including the reconstruction loss L_1 , perceptual loss L_{percept} and style loss L_{style} . In this work, we formulate the assignment of loss weights as a sequential decision-making problem and address it using PPO. The Pix2Pix training procedure is treated as an environment, while the PPO agent serves as a controller that observes the current training state and dynamically outputs an updated weight vector λ at each epoch, guided by reward signals. The agent aims to learn a policy $\pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t)$ that maximizes the expected discounted cumulative reward $J(\theta)$.

4. PPO-Guided Adaptive Loss Balancing

Figure 2 illustrates the proposed RL-guided loss balancing framework, consisting of training and inference phases. During training, MMS panoramic images are projected into viewport images, which are perspective views extracted from the panorama to reduce geometric distortion. These viewport images are then masked the target object to form inputs and binary masks for a Pix2Pix-based inpainting network with multiple losses. At each epoch, the outputs are evaluated against the ground truth images on a validation set to compute quality metrics as well as loss values. These quantities are encoded as the state $\mathbf{s}_t$ and a reward r_t for a PPO agent $\pi_{\theta}(\mathbf{a}_t | \mathbf{s}_t)$. The agent predicts updated loss weight coefficients $\{\lambda_{L_1}, \lambda_{\text{perc}}, \lambda_{\text{style}}\}$, forming a closed-loop that improves reconstruction quality. During inference, the trained generator operates as a fixed feedforward model, processing each panorama through viewport projection, region selection, masking, inpainting, and inverse projection to produce the final output.

4.1. Target area selection

Street-view images are commonly represented in equirectangular projection, which captures a full 360-degree panoramic view. However, these images pose challenges for GAN-based inpainting. First, processing the entire panorama is computationally expensive. Second, conventional inpainting models assume planar geometry, which is violated in equirectangular images, leading to distortions and artifacts. Although some methods such as cube mapping (Han & Suh, 2021) can reduce distortion by splitting the panorama into multiple views, they require complexity and are not well-suited for localized object removal.

To address these issues, we introduce a viewport projection module that extracts a localized perspective view focused directly on the target object, avoiding the need to process the entire panoramic scene. This method also reduces geometric distortion and preserves object scales, and enables the use of standard planar inpainting models such as Pix2Pix without architectural changes. Formally, the equirectangular image is mapped onto a unit sphere. For a pixel (x, y) in an image of width W and height H , the spherical coordinates (θ, ϕ) are computed as:

$$\theta = \frac{2\pi x}{W} - \pi, \quad \phi = \frac{\pi y}{H} - \frac{\pi}{2} \quad (7)$$

where $\theta \in [-\pi, \pi]$ is the azimuth angle, representing horizontal position around the sphere, and $\phi \in [-\pi/2, \pi/2]$ is the polar angle representing vertical elevation. Using these spherical coordinates, a target viewport is extracted by defining a perspective projection centered at a predefined heading θ_c and elevation

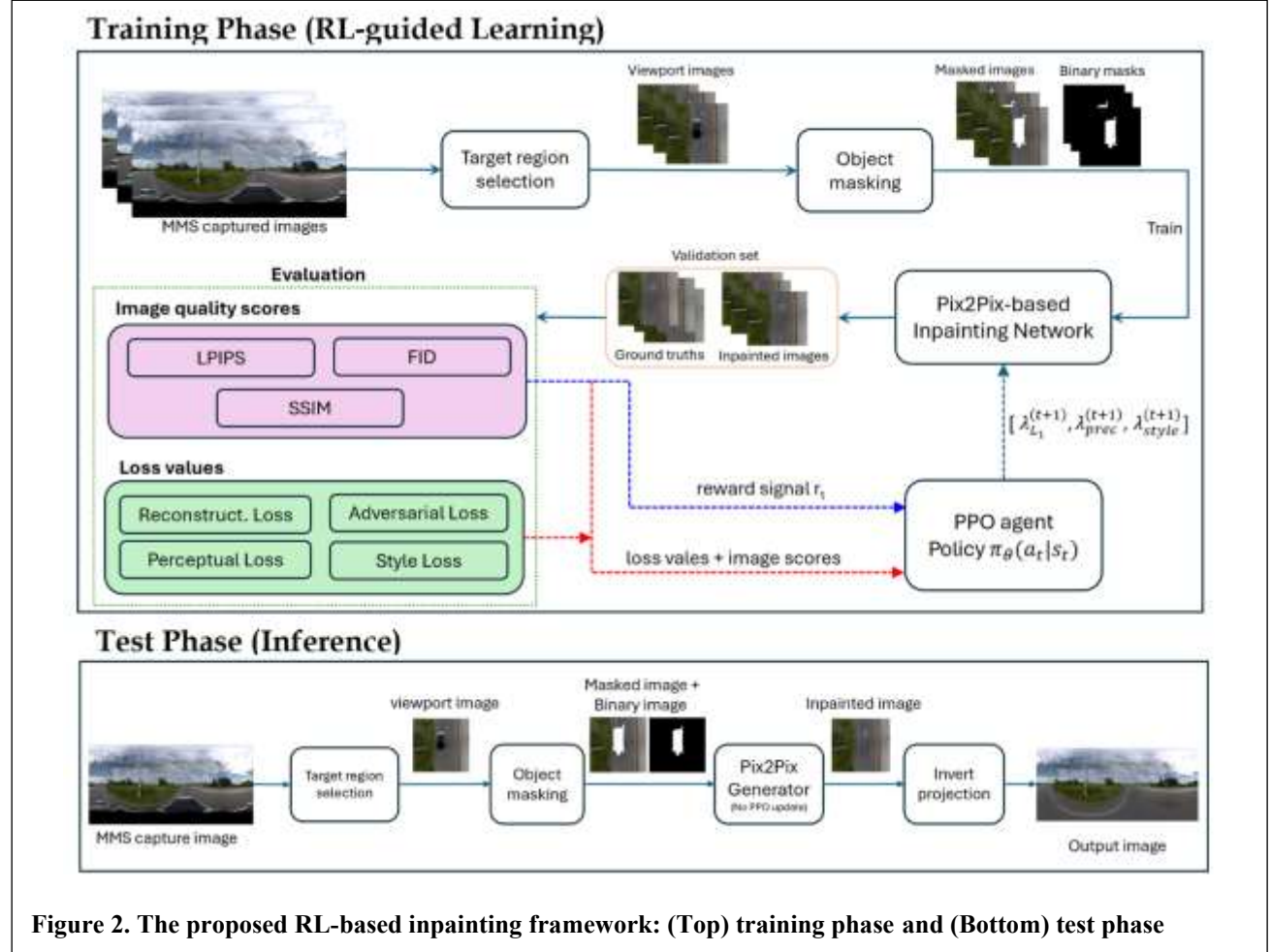


Figure 2. The proposed RL-based inpainting framework: (Top) training phase and (Bottom) test phase

ϕ_c within a specified field of view (FOV), ensuring the target object is centrally positioned. The resulting perspective projection operator maps the relevant region of the equirectangular panorama to a planar image patch of size $H_p \times W_p$, yielding the perspective-projected viewport image used as input to the inpainting network. In this study, the viewport is centered at $\theta_c = 0$ and $\phi_c = -90$ to focus the projection directly below the camera where the survey vehicle is located, with a field of view of 167° and an output resolution of 512×512 pixels. A tight crop is applied around the target object region to isolate it and its relevant local background, discarding irrelevant peripheral content that are not required for inpainting. Furthermore, it ensures that the vehicle occupies a consistent and centered position within the cropped patch across all frames. The cropped region is then resized to a fixed resolution of 256×256 pixels, yielding the final viewport patch $I_p \in \mathbb{R}^{256 \times 256 \times 3}$, as illustrated in Figure 3.

4.2. Object Masking

Given the viewport image I_p , the masking step generates a masked context image $I_c \in \mathbb{R}^{H_p \times W_p \times 3}$ using a binary mask $M \in \{0, 1\}^{H_p \times W_p}$. The mask is defined such that $M_{ij} = 1$ for pixels belonging to the target object and $M_{ij} = 0$ elsewhere. In this work, M is a fixed predefined mask of the target object, eliminating the need for per-frame

segmentation. The masked image is generated by $I_c = I_p \odot (1 - M) + 1 \odot M$ where $\odot$ denotes element-wise region with a uniform value, providing a clear signal for the inpainting network. The pair (I_c, M) is then used as inputs to the generator network.

C. PPO-guided Loss Balancing

Given an input pair (I_c, M) , the Pix2Pix inpainting generator G_θ reconstructs the masked region with a plausible background, producing the inpainted patch $\hat{y} = G_\theta(I_c, M)$. The generator is trained under a combined multi-loss objective comprising four components, including an adversarial loss L_{adv} , a reconstruction loss L_1 , a perceptual loss L_{perc} , and a style loss L_{style} . The total generator loss is defined as

$$L_G = L_{adv}(G, D) + \lambda_{L_1} L_1(G) + \lambda_{perc} L_{perc}(G) + \lambda_{style} L_{style}(G) \quad (8)$$

where λ_{L_1} , λ_{perc} , and λ_{style} , are scalar weight coefficients controlling the relative contribution of the L1, perceptual, and style loss terms, respectively. In standard Pix2Pix training, loss weights are fixed and do not adapt to training dynamics. However, the optimal balance changes over time, where early stages often require favor stronger L_1 supervision for structure, while later stages benefit from increased perceptual and style weighting for texture and visual refinement.

To overcome this limitation, Proximal Policy Optimization (PPO) is used to dynamically adjust $\lambda = \{\lambda_{L_1}, \lambda_{perc}, \lambda_{style}\}$ at each training epoch t based on observed training state and image quality reward signals. The PPO agent treats the Pix2Pix training process as an environment: at each epoch, it observes the current state s_t , output an action a_t , the generator is trained for one epoch under these weights, and a reward r_t , reflecting validation image quality is returned. The objective is to learn a policy $\pi_\theta(a_t|s_t)$ that maximizes the expected discount reward. In this work, a state $s_t \in \mathbb{R}^{10}$ is defined as:

$$s_t = [\Delta L_{adv}^{(t)}, \Delta L_1^{(t)}, \Delta L_{perc}^{(t)}, \Delta L_{style}^{(t)}, \lambda_{L_1}^{(t)}, \lambda_{perc}^{(t)}, \lambda_{style}^{(t)}, Q_{SSIM}^{(t)}, Q_{FID}^{(t)}, Q_{LPIPS}^{(t)}] \quad (9)$$

where $\Delta L_k^{(t)}$ denotes epoch-averaged loss values for loss component k , $\lambda_k^{(t)}$ are the current loss weights and $Q_k^{(t)}$ are image-quality metrics, including Structural Similarity Index Measure (SSIM) [], Fréchet Inception Distance (FID) [], and Learned Perceptual Image Patch Similarity (LPIPS) [], evaluated on the validation set at the end of epoch t . State distributions are normalized online to stabilize learning.

Action space: The action $a_t \in \mathbb{R}^3$ corresponds to the updated loss weight assignment applied to the generator objective for the subsequent epoch:

$$a_t = [\lambda_{L_1}^{(t+1)}, \lambda_{perc}^{(t+1)}, \lambda_{style}^{(t+1)}] \quad (10)$$

where $a_t \in [10^{-4}, 100]$. This bounded continuous space prevents vanishing or dominating loss terms.

Reward function: The reward r_t is computed from validation quality metrics at the end of each epoch to align loss weight adaptation. It is defined as:

$$r_t = 10 \cdot Q_{SSIM}^{(t)} - Q_{FID}^{(t)} - 100 \cdot Q_{LPIPS}^{(t)} \quad (11)$$

The r_t is clipped to $[-20, +10]$ to mitigate instability caused by transient generator failures during early training. The scaling factor of 100 normalizes the ranges of SSIM and LPIPS to be comparable with FID, ensuring balanced contributions across the metrics.

The PPO policy is used every 16 steps using the clipped surrogate objective in Eq. (5), with the critic network

Algorithm 2 Adaptive Multi-Loss Enhanced Pix2Pix

Require: Dataset (x, y)
Require: Generator G , Discriminator D
Require: PPO policy π_θ , value network V_ϕ
Require: Initial loss weights $\lambda = (\lambda_{L1}, \lambda_{perc}, \lambda_{style})$
Require: Update interval $N = 16$

```

1: for training step do
2:   Observe state  $s_t = (L_{L1}, L_{perc}, L_{style}, \lambda_{L1}, \lambda_{perc}, \lambda_{style}, FID, LPIPS)$ 
3:   Sample action  $a_t \sim \pi_\theta(s_t)$ 
4:   Update loss weights:  $\lambda_{t+1} = \lambda_t + a_t$ 
5:   for each mini-batch  $(x, y)$  do
6:     Generate image:  $\hat{y} = G(x)$ 
7:     // Train Discriminator
8:      $L_D^{real} = -\log D(x, y)$ 
9:      $L_D^{fake} = -\log(1 - D(x, \hat{y}))$ 
10:     $L_D = L_D^{real} + L_D^{fake}$ 
11:    Update  $D$  using  $\nabla_D L_D$ 
12:    // Compute Generator Losses
13:     $L_{GAN} = -\log D(x, \hat{y})$ 
14:     $L_{L1} = \|y - \hat{y}\|_1$ 
15:     $L_{perc} = \|\phi(y) - \phi(\hat{y})\|_2$ 
16:     $L_{style} = \|Gram(\phi(y)) - Gram(\phi(\hat{y}))\|_F$ 
17:    Total generator loss:
        
$$L_G = L_{GAN} + \lambda_{L1} L_{L1} + \lambda_{perc} L_{perc} + \lambda_{style} L_{style}$$

18:    Update  $G$  using  $\nabla_G L_G$ 
19:  end for
20:  Compute reward  $r_t$  (based on FID / LPIPS / SSIM / PSNR)
21:  Store transition  $(s_t, a_t, r_t, s_{t+1})$ 
22:  if  $t \pmod N == 0$  then
23:    Update PPO policy using clipped objective:
        
$$L^{PPO}(\theta) = E[\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t)]$$

24:  end if
25: end for

```

V_ϕ updated simultaneously to provide accurate advantage estimates. This closed-loop formulation enables the agent to progressively learn a weight-selection policy that adapts the loss balance in response to the evolving state of the generator, steering it toward perceptually realistic inpainting reconstructions throughout training as summarized in Algorithm 1.

5. Experiments

5.1 Datasets

Two MMS image datasets are collected by using a Ladybug5+ panoramic camera integrated with a GNSS receiver for geo-referencing. The first dataset contains 13,252 panoramic images captured along a 14.3 kilometer route in Bangkok, Thailand, named “BKK” dataset. The second dataset comprises 26,742 images collected along a 9.7 kilometer route in Nakhonsawan city, Thailand, named “NKS” dataset. Each image in both datasets has a resolution of 4,096 x 8,192 pixels. The data were captured under varying illumination conditions, including overcast, partially cloudy, and clear-sky scenarios, introducing realistic appearance variability. Table 3 illustrates statistics of both datasets. Sample images from both datasets are shown in Figure 4.

No data augmentation is applied to evaluate performance under realistic MMS acquisition conditions. The primary goal of this experiment is to remove the survey vehicle that persistently appears at the bottom of each

Table 1. MMS panoramic datasets

Dataset	#Train	#Validation	#Test	Total
BKK	10,252	1,500	1,500	13,252
NKS	20,742	2,000	2,000	24,742



Figure 4. Image samples of (top) BKK dataset and (bottom) NKS dataset

image. As the datasets do not include explicit ground truth annotations, we use GPS coordinates to identify the background region behind the target object in each frame. Let (x_t, y_t) denote the center coordinates of the target object in image t and let θ_t represent the vehicle’s heading direction. A reference point (x_q, y_q) is computed at a distance q_t behind the object using the following equation: $x_q = q_t * \cos(\theta_t + 180) + x_t$ and $y_q = q_t * \sin(\theta_t + 180) + y_t$. After determining this reference point, the dataset is searched for an image covering the corresponding location. The retrieved image, containing the background behind the target, is then used as the ground truth for the current image. Figure 5 illustrates an example of the constructed training data.

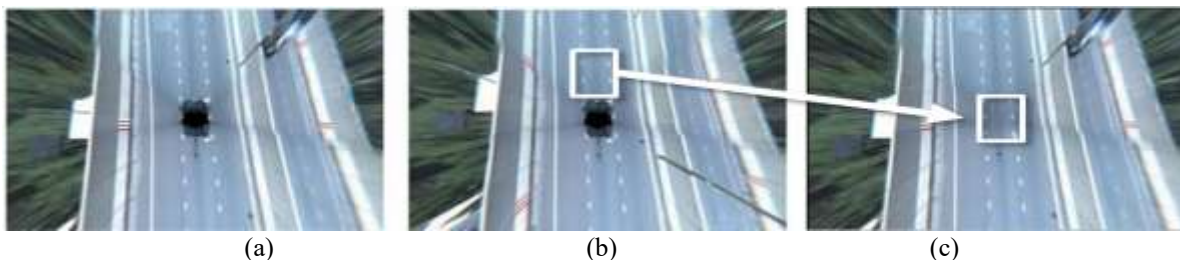


Figure 5. Ground truth image generation with GPS-based matching (a) input image at time step t, (b) Retrieved reference image based on GPS location matching, and (c) Ground truth background generation corresponding to image t.

5.2 Loss-Weighting Strategies

The proposed PPO-based loss balancing framework is evaluated against three baseline methods, all of which use the same Pix2Pix generator and discriminator architecture.

- **Fixed-Weight** This baseline uses constant loss weights selected through a preliminary grid search. For the Bangkok dataset, the coefficients are set to $\lambda_{L_1} = 30$, $\lambda_{prec} = 12.0$, $\lambda_{style} = 10.5$. For the Nakhon dataset, the weights are $\lambda_{L_1} = 20$, $\lambda_{prec} = 25.0$, $\lambda_{style} = 3.0$ through training process.
- **SoftAdapt (Heydari et al., 2019)** is a heuristic adaptive loss weighting method that dynamically rescales each loss component based on its relative rate of change between consecutive training epochs. At epoch t , the weight λ_k for the k -th loss term is computed using a softmax over normalized loss-change rates. In this experiment, the normalized loss rate variant is applied, and weights are updated every five epoch.
- **GradNorm (Chen et al., 2018)** is a gradient-based adaptive loss weighting method that balances multiple loss terms by equalizing their gradient magnitudes with respect to a shared network layer. During training, GradNorm dynamically adjusts each weight coefficient by increasing the weights of slower-learning losses and decreasing those of faster-learning ones, ensuring balanced optimization across loss terms. In this experiment, GradNorm is used to balance the weight coefficients of the Pix2Pix generator, using the generator's last layer as the shared layer for gradient computation. The weights are updated by minimizing a gradient norm alignment loss with respect to w_k only, leaving the generator parameters unchanged, and are renormalized after each update to maintain a constant sum. Weight updates are performed every five epoch, with the balancing hyperparameter $\alpha = 0.5$.

5.3 Implementation details

Pix2Pix backbone. The generator adopts a U-Net architecture consisting of seven encoder blocks (Conv2d with stride 2, Instance Normalization, LeakyReLU) and seven corresponding decoder blocks (ConvTranspose2d with skip connections, Tanh output activation). The bottleneck feature dimension is $512 \times 2 \times 2$. The discriminator is a PatchGAN that evaluates 70×70 image patches. The perceptual loss L_{prec} is computed using feature maps from a pretrained VGG-19 network at layer relu2_2, relu3_3, relu4_3, relu5_1, and relu5_3 while the style loss L_{style} is derived from Gram matrices at layers relu1_2, relu2_2, relu3_3, and relu4_3. Both the generator and discriminator are trained using the Adam optimizer with learning rate $\alpha = 0.0004$ and momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$. All models are trained for 5,000 epochs with a batch size of 16 on NVIDIA A100 GPU.

PPO controller. The PPO controller is implemented using Stable-Baselines3 [15]. Both the Actor and Critic networks are two-layer MLPs with 64 hidden units and Tanh activations. The action space is a three-dimensional continuous vector $\lambda \in [10^{-4}, 100]^3$, clipped element-wise after sampling. The rollout length before each policy update is $n_steps = 16$ epochs. The entropy coefficient is $ent_coef = 0.01$, and State-Dependent Exploration ($use_sde = True$) is enabled to encourage richer exploration in the continuous action space. The discount factor is $\gamma = 0.99$, and the GAE parameter is $\lambda = 0.95$. Rewards are additionally clipped to ± 30 to suppress instability from transient generator failures during early training. The initial loss weights are $\lambda_{L_1} = \lambda_{prec} = \lambda_{style} = 0.100$ for all runs.

5.3 Evaluation metrics

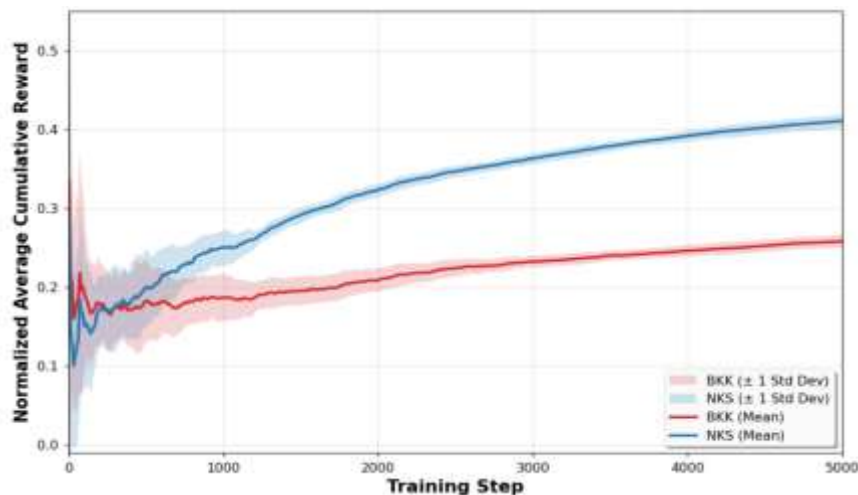


Figure 6. Normalized average cumulative reward over 5,000 training epochs on MMS datasets

All models are evaluated on the held-out test set using four complementary metrics: Structural Similarity Index Measure (SSIM), Learned Perceptual Image Patch Similarity (LPIPS), and Fréchet Inception Distance (FID). Higher SSIM indicates better pixel-level fidelity, while lower LPIPS and FID indicate better perceptual quality and distributional realism, respectively.

To validate that the PPO agent learns a progressively improving weight selection policy throughout training, Figure 6 presents the normalized average cumulative reward of the PPO agent over 5,000 training epochs on the BKK (red line) and NKS (blue line) datasets, reported as the mean across $N=5$ independent runs. The reward at each epoch is computed by Equation (5) and normalized by the total number of training epochs to yield the average per-epoch reward, removing the confounding effect of dataset size and making the trajectories directly comparable across both datasets. Both curves exhibit monotonically increasing with narrowing variance after epoch 1,000, confirming that the PPO agent learns a stable and reproducible weight-selection policy on both datasets. The consistently higher cumulative reward on NKS reflects the greater learning benefit derived from the larger and more statistically diverse dataset, which provides richer training signal for a more effective loss weight configuration.

6. Results and discussion

Table 2 reports the average and standard deviation (SD) across evaluation metrics over five independent runs with different random seeds. As shown in this table, the proposed PPO-based framework consistently achieves the best performance on both datasets (LPIPS = 0.087 ± 0.005 , FID = 6.633 ± 0.22 , SSIM = 0.968 ± 0.002 , and PSNR = 31.24 ± 0.38 dB on the BKK dataset, and LPIPS = 0.085 ± 0.004 , FID = 6.322 ± 0.19 , SSIM = 0.971 ± 0.001 , and PSNR = 31.58 ± 0.35 dB on the NKS dataset). Compared to the fixed-weight baseline, PPO yields substantial improvements, particularly in LPIPS and FID, while also achieving higher SSIM. The fixed-weight method exhibits weaker performance (LPIPS = 0.152 ± 0.008 and FID = 8.870 ± 0.316

on BKK), along with relatively larger variance, indicating sensitivity to initialization and limited robustness. For both adaptive baselines, including SoftAdapt and GradNorm, significantly outperform the fixed-weight approach, confirming the benefit of dynamic loss weighting. However, neither matches the PPO framework. On BKK, SoftAdapt slightly outperforms GradNorm (LPIPS: 0.096 vs. 0.104; FID: 7.082 vs. 7.411), whereas on NKS, GradNorm achieves better LPIPS and FID (0.093 and 6.727) than SoftAdapt (0.101 and 7.040), suggesting dataset-dependent behavior. Overall, PPO not only delivers the best quantitative results but also maintains consistently low variance across runs, demonstrating stable and reliable learning. These findings highlight the advantage of formulating loss balancing as a reward-driven optimization problem, leading to superior image quality compared to both fixed and heuristic adaptive strategies.

Figure 7 illustrates the validation convergence of four loss-weighting strategies, including Fixed-weight (black-dashed), Soft-Adapt (yellow dotted), GradNorm (purple dotted), and the proposed PPO-based method (red),

Table 2 Reconstruction comparisons across loss-weighting strategies over the datasets

Method	Category	Dataset	Reconstruction quality		
			LPIPS (↓)	FID (↓)	SSIM (↑)
Fixed-Weight	Static	BKK	0.152 ± 0.008	8.870 ± 0.316	0.901 ± 0.003
		NKS	0.127 ± 0.007	8.476 ± 0.280	0.904 ± 0.002
SoftAdapt	Loss-convergence	BKK	0.096 ± 0.006	7.082 ± 0.250	0.965 ± 0.003
		NKS	0.101 ± 0.003	7.040 ± 0.220	0.968 ± 0.002
GradNorm	Gradient-balance	BKK	0.104 ± 0.007	7.411 ± 0.295	0.963 ± 0.004
		NKS	0.093 ± 0.006	6.727 ± 0.240	0.966 ± 0.003
PPO*	Quality-reward	BKK	0.087 ± 0.005	6.633 ± 0.225	0.968 ± 0.004
		NKS	0.085 ± 0.006	6.322 ± 0.198	0.971 ± 0.002

over 5,000 training epochs on two MMS datasets. As shown in this figure, all the methods exhibit a similar trend: rapid error reduction in early epochs (0-1,000), gradual improvement in mid epochs (1000 – 2500) and stable convergence thereafter (2,500-5,000). The PPO-based method consistently outperforms all the baselines across LPIPS, SSIM, and FID. On BKK, it achieves significant gains over the fixed-weight baseline (-42.8% LPIPS, +7.4% SSIM, and -25% FID), with comparable improvements on NKS (-33.1% LPIPS, -25.4% FID, +7.5% SSIM). It also demonstrates more stable convergence than SoftAdapt and GradNorm. These results indicate that PPO effectively learns to balance loss terms by progressively shifting from reconstruction to perceptual and style objectives, leading to superior inpainting performance without manual tuning.

Figure 8 shows distinct loss-weight adaptation patterns across BKK and NKS. The Fixed-Weight baseline remains constant, whereas the adaptive methods progressively redistribute emphasis among reconstruction, perceptual, and style objectives. SoftAdapt exhibits the largest short-term

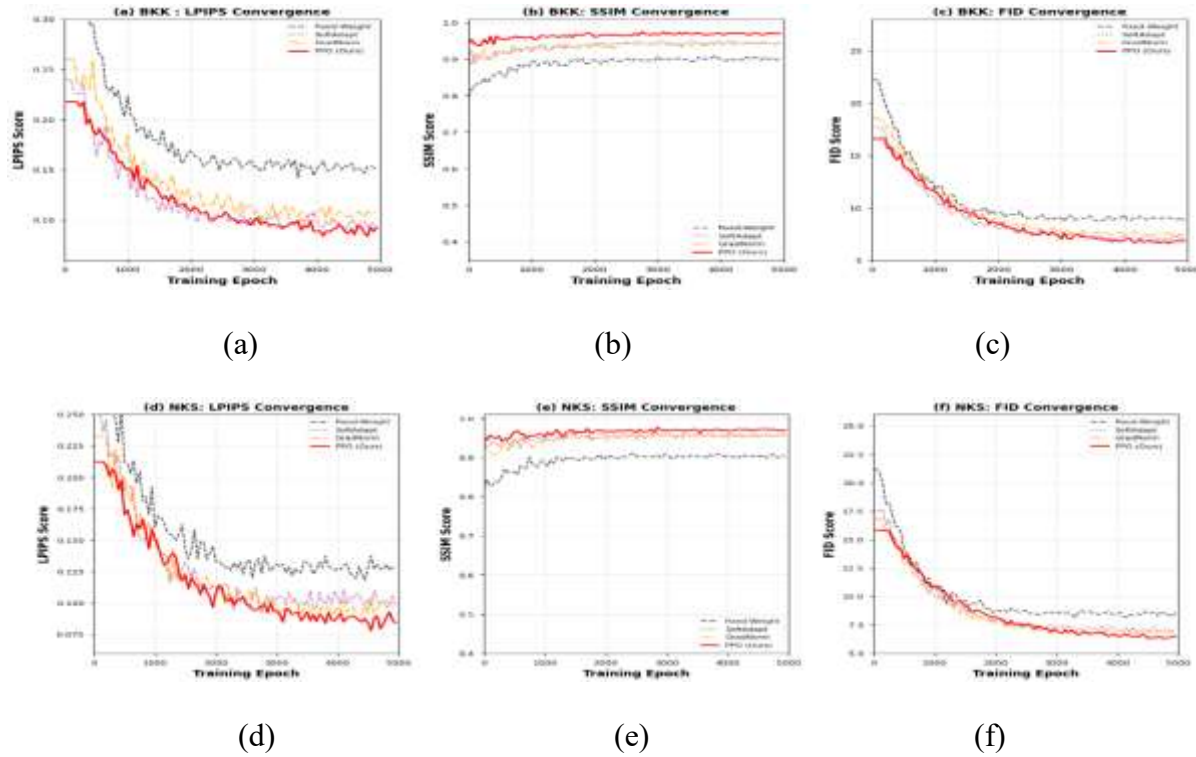


Figure. 7 Validation metric convergence comparison across training epochs (a)-(c) results on BKK dataset and (d)-(f) results on NKS dataset

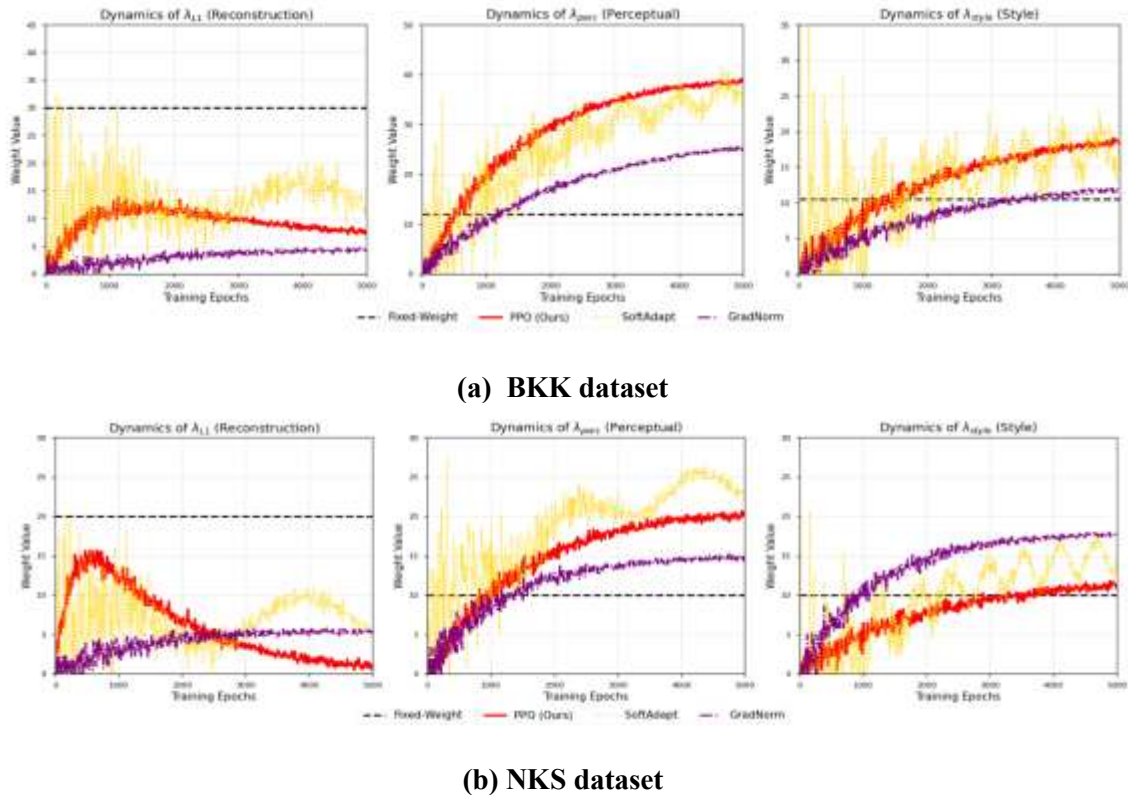


Figure 8. Weight adaptations of over 5,000 training epochs on (a) the BKK dataset and (b) NKS dataset

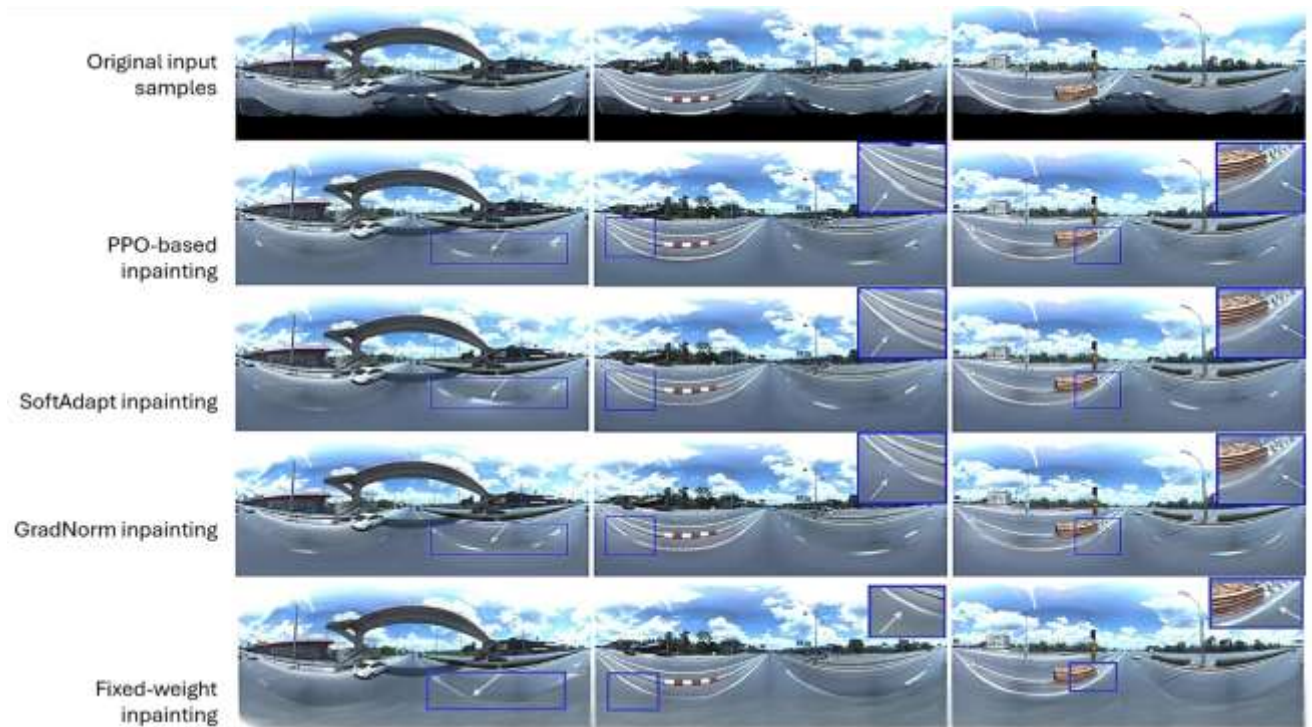


Figure 9. visual result comparison across the models on BKK dataset

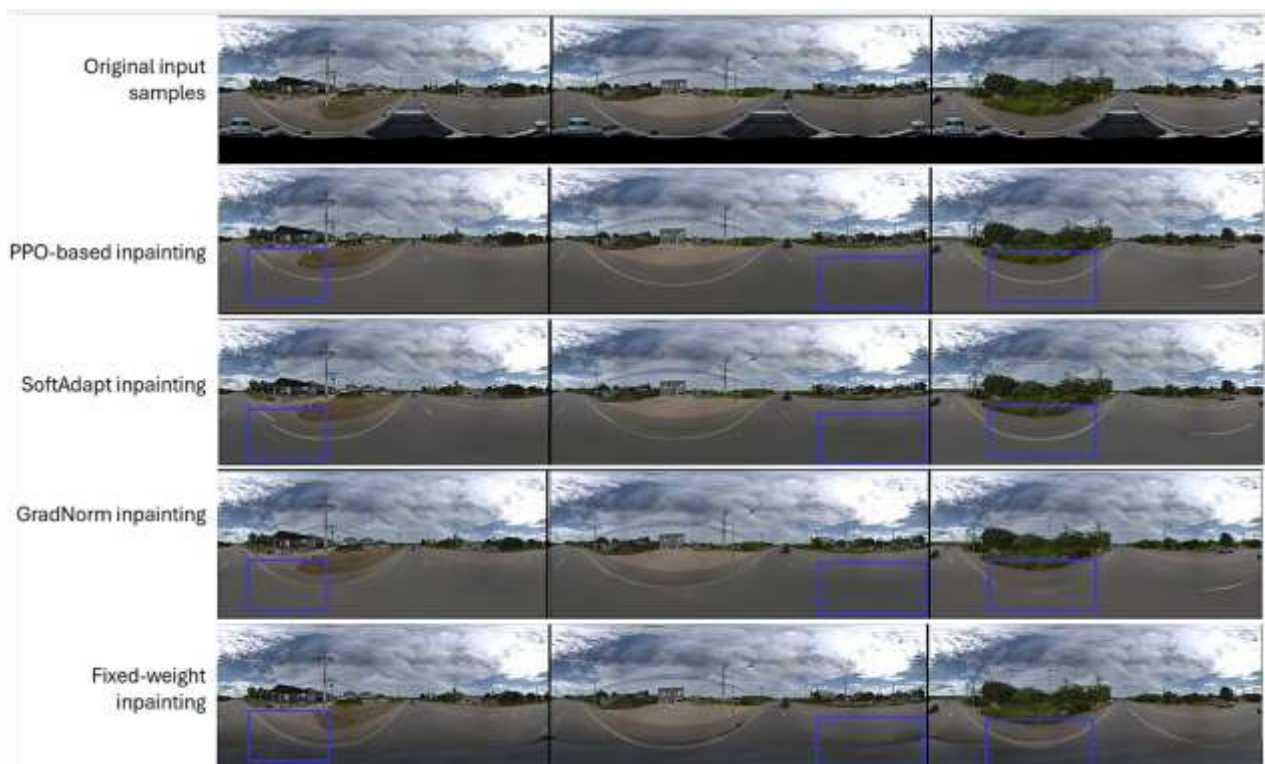


Figure 10. visual result comparison across the models on NKS dataset

gradually.
fluctuations, particularly during early training, while GradNorm changes more smoothly and gradually.

For BKK, PPO initially increases the reconstruction weight before gradually reducing it, while the perceptual and style weights increase consistently throughout training. This indicates a shift from early pixel-level reconstruction toward stronger perceptual and texture-related objectives at later stages. For NKS, the transition is more pronounced: PPO rapidly increases the reconstruction weight early in training but subsequently suppresses it to a very low level, while the perceptual weight rises steadily and the style weight increases moderately. GradNorm follows smoother monotonic trends, with particularly strong emphasis on the style term for NKS.

PPO exhibits structured, dataset-specific adaptation, progressively reallocating weight from reconstruction toward perceptual and style objectives rather than following a fixed weighting pattern. In contrast, SoftAdapt is substantially more oscillatory, whereas GradNorm provides smoother but more gradual adjustments. These trends suggest that PPO learns different weighting trajectories according to the optimization dynamics of each dataset.

7. Ablation study

Impact of reward signal: To investigate how the choice of reward formulation influences the PPO agent's learned policy and the final inpainting quality, we evaluate six distinct rewards (r_1 to r_6) as detailed in Table 3. This is important because the reward function acts as the sole objective guiding the agent's weight selection strategy, leading to improving perceptual image quality. As shown in this table, the structural-only reward r_1 achieves the highest SSIM on BKK (0.970 ± 0.002) but the worst LPIPS and FID among all variants. This indicates that relying solely on structural supervision encourages the generator toward smooth but perceptually unrealistic reconstructions. In contrast, the distributional reward r_2 ($-Q_{FID}$) significantly improves perceptual quality metrics but sacrifices SSIM. the perceptual reward r_3 ($-100 \cdot Q_{LPIPS}$) emerges as the most effective single-component signal, achieving LPIPS = 0.089 ± 0.004 and FID = 6.680 ± 0.220 on BKK with competitive SSIM (0.968 ± 0.002). This suggests that patch-level perceptual similarity provides the most informative single learning signal for this task.

For two-component combinations, the distributional + perceptual reward r_4 ($-Q_{FID} - 100 \cdot Q_{LPIPS}$) outperforms both r_2 and r_3 individually in terms of LPIPS and FID, suggesting that that these two perceptual objectives are complementary. In contrast, the structural + perceptual reward r_5 ($10 \cdot Q_{SSIM} - 100 \cdot Q_{LPIPS}$) further improves performance by incorporating SSIM as a stabilizing structural constraint, consistently outperforming all single-component and the Distributional + Perceptual variants on LPIPS and FID. Finally, the proposed reward r_6 ($10 \cdot Q_{SSIM} - Q_{FID} - 100 \cdot Q_{LPIPS}$) achieves the best LPIPS and FID scores across both datasets (0.087 ± 0.005 and 6.630 ± 0.200 on BKK, and 0.085 ± 0.004 and 6.321 ± 0.200 on NKS), while maintaining SSIM statistically indistinguishable from the structural only maximum. This demonstrates that combining structural, perceptual, and distributional rewards provide complementary and mutually reinforcing supervision, enabling the PPO agent to learn loss weight configurations that jointly optimize structural fidelity, perceptual realism, and distributional consistency.

PPO update intervals: The PPO update interval is a critical hyperparameter that directly influences the stability of the closed-loop training process. Intuitively, this parameter introduces a trade-off: shorter intervals enable the agent to response quickly to changing loss dynamics, but

Table 3 Effects of different reward components on the BKK and NKS datasets over five different runs.

Reward Component	Description	Dataset	Reconstruction Quality		
			LPIPS	FID	SSIM
$r_1 = 10 \cdot Q_{SSIM}$	Structural Only	BKK	0.118±0.008	7.850±0.350	0.970±0.002
		NKS	0.108±0.007	7.400±0.300	0.971±0.001
$r_2 = -Q_{FID}$	Distributional Only	BKK	0.098±0.006	6.890±0.204	0.962±0.003
		NKS	0.091±0.005	6.520±0.210	0.965±0.002
$r_3 = -100 \cdot Q_{LPIPS}$	Perceptual Only	BKK	0.089±0.004	6.680±0.220	0.968±0.002
		NKS	0.088±0.004	6.380±0.190	0.971±0.001
$r_4 = -Q_{FID} - 100 \cdot Q_{LPIPS}$	Distributional + Perceptual	BKK	0.092±0.006	6.850±0.280	0.969±0.002
		NKS	0.089±0.005	6.500±0.230	0.972±0.002
$r_5 = 10 \cdot Q_{SSIM} - 100 \cdot Q_{LPIPS}$	Structural + Perceptual	BKK	0.088±0.005	6.650±0.230	0.968±0.002
		NKS	0.087±0.004	6.340±0.200	0.971±0.002
$r_6 = 10 \cdot Q_{SSIM} - Q_{FID} - 100 \cdot Q_{LPIPS}$	Full	BKK	0.087±0.005	6.630±0.200	0.968±0.002
		NKS	0.085±0.004	6.321±0.200	0.972±0.001

can destabilize the Pix2Pix optimization, while longer intervals provide a more stable training signal for the generator but limit the PPO agent's ability to adapt effectively due to delayed feedback. To identify the optimal balance, we evaluate update intervals of 4, 8, 16, and 32 epochs on both the BKK and NKS datasets using the best-performing reward signal r_6 with all other hyperparameters fixed. As summarized in Table 4, results are averaged over five independent runs. The 4-epoch interval obtains the weakest performance and the highest variance across all metrics, suggesting that too frequent updates introduce the agent to noisy and unreliable reward signals before the generator has stabilized.

Table 4 Performance evaluation of policy update intervals on both datasets

Update interval	Dataset	Reconstruction Quality		
		LPIPS	FID	SSIM
4 epochs	BKK	0.103±0.008	7.250±0.350	0.967±0.003
	NKS	0.099±0.006	6.900±0.310	0.970±0.002
8 epochs	BKK	0.093±0.006	6.850±0.260	0.968±0.002
	NKS	0.089±0.005	6.500±0.220	0.970±0.001
16 epochs	BKK	0.089±0.005	6.680±0.230	0.969±0.002
	NKS	0.086±0.004	6.380±0.200	0.972±0.001
32 epochs	BKK	0.090±0.004	6.750±0.190	0.970±0.002
	NKS	0.087±0.004	6.450±0.160	0.972±0.001

As shown in Table 4, performance improves consistently as the interval increases to 8 and 16 epochs, with the 16-epoch configuration achieving the best overall results. This confirms that updating interval of 16 epochs provides an effective balance between policy responsiveness and reward stability. However, further increasing the interval to 32 epochs leads to slight decrease in LPIPS and FID, while SSIM remain unchanged. This suggests that infrequent updates hinder the agent's ability to adapt to evolving generator dynamics, particularly during critical transition phases of GAN training. Figure 12 compares qualitative inpainting results obtained from the three highest reward signals, r_4 , r_5 , and r_6 , across representative viewport samples. In all cases, the survey vehicle is effectively removed and the missing road surface is reconstructed with visually consistent texture,

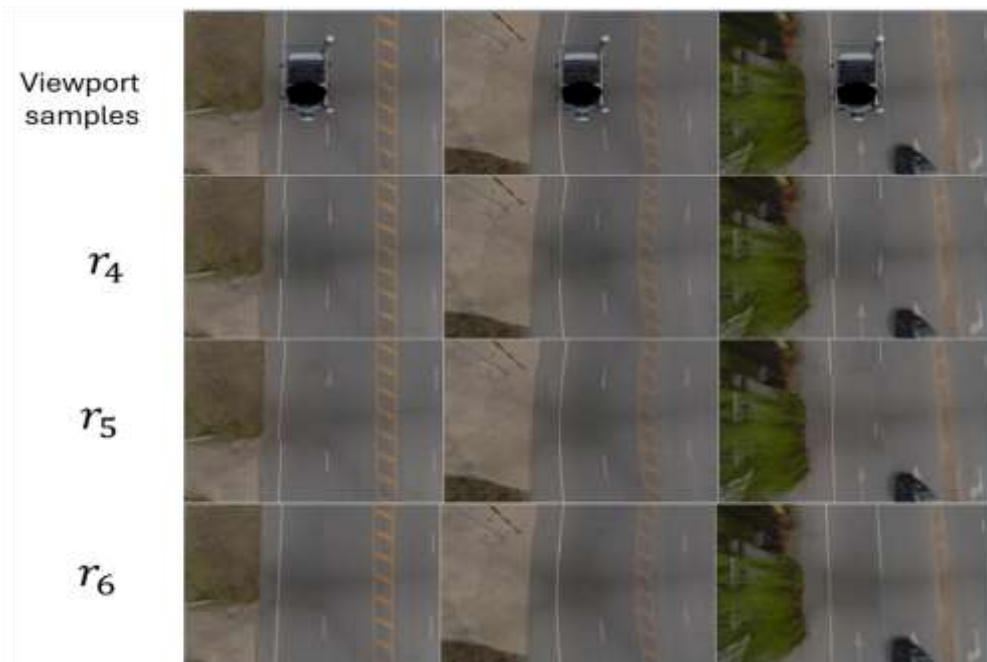


Figure. 12 Visual results based on top-3 reward signals

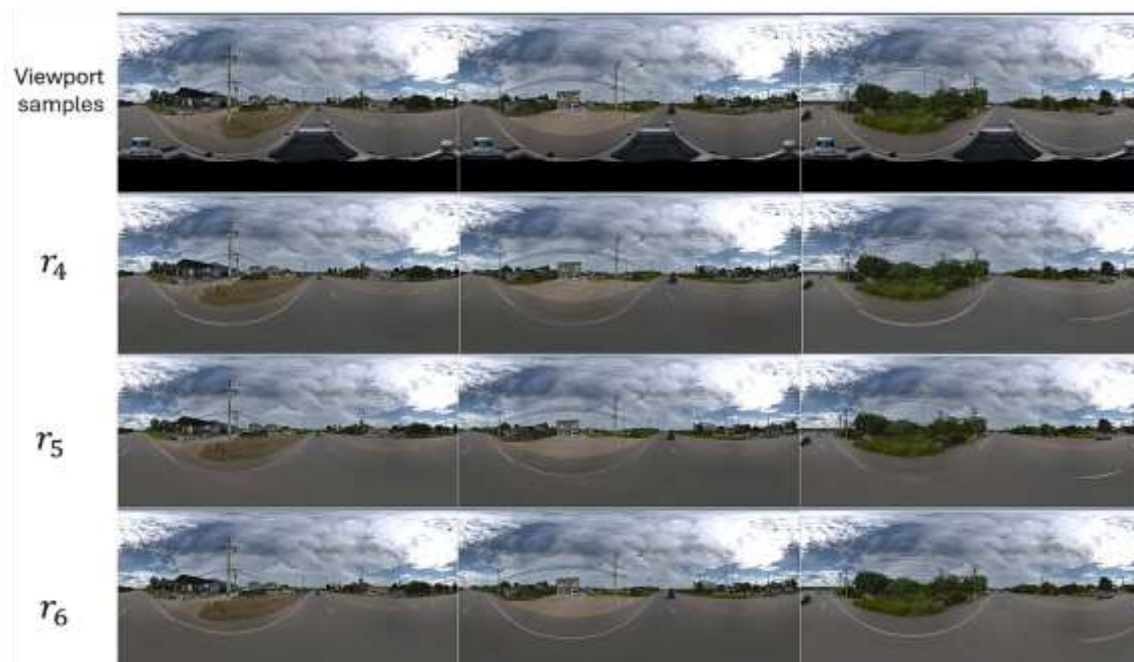


Figure 13 Visual results based on top-3 reward signals

lane markings, and surrounding structure. Across the three reward signals, the outputs remain broadly stable, indicating that the learned policy produces consistent reconstructions among the top-performing reward states. Minor differences are mainly visible in local texture smoothness and lane-marking continuity, while the overall road geometry and scene appearance are preserved. This suggests that higher-reward solutions converge toward similar visually plausible reconstructions rather than producing substantially different outputs. Figure 13 presents the reconstructed results for panoramic images.

Input resolution: Different viewport resolutions influence the level of spatial information available to the model and can affect the visual quality produced by the generator. To assess the resolution effect, we evaluate the PPO framework at three different viewport resolutions, including 256×256 , 512×512 , and 1024×1024 , across five different runs, using the proposed reward signal while keeping all other hyperparameters fixed.

Table 5 presents the impact of viewport resolution on the visual performance of the proposed PPO framework across both the BKK and NKS datasets. As illustrated in this table, the lowest resolution (256×256) obtains the

Table 5 Visual Quality across different input resolutions

Resolution	Dataset	Visual Quality Metrics		
		LPIPS ($\downarrow$)	FID ($\downarrow$)	SSIM ($\uparrow$)
256×256	BKK	0.115 ± 0.009	8.850 ± 0.420	0.962 ± 0.003
	NKS	0.108 ± 0.008	8.400 ± 0.380	0.965 ± 0.002
512×512	BKK	0.087 ± 0.005	6.633 ± 0.220	0.968 ± 0.002
	NKS	0.086 ± 0.005	6.322 ± 0.190	0.972 ± 0.001
1024×1024	BKK	0.091 ± 0.006	6.950 ± 0.280	0.966 ± 0.003
	NKS	0.089 ± 0.006	6.650 ± 0.250	0.970 ± 0.002

weakest results across all metrics on both datasets. This indicates that limited spatial resolution restricts the generator's ability to capture the fine-grained structural and textural detail required for perceptual background reconstruction. Furthermore, limited resolution can reduce the reliability of the PPO reward signal by producing less informative validation metric estimates. The 512×512 resolution achieves the best overall inpainting performance across all metrics and both datasets, confirming that it provides an optimal resolution to preserve spatial detail. In contrast, increasing the resolution to 1024×1024 leads to consistent degradation compared to 512×512 . This suggests that higher resolution inputs introduce complexity and variance in the training process, making it more difficult for the PPO agent to reliably detect and exploit improvements in the reward signal.

8. Conclusion

This paper introduces a reinforcement learning-guided framework for adaptive loss balancing in object inpainting for panoramic Mobile Mapping System imagery. A Proximal Policy Optimization (PPO) agent dynamically adjusts the Pix2Pix loss coefficients at each epoch using reward signals derived from SSIM, FID, and LPIPS. By formulating loss weighting as a sequential decision-making problem, the method eliminates manual tuning and enables automatic discovery of dataset-specific optimal configurations. Experiments on the BKK and NKS datasets (over five runs) show that the proposed approach consistently outperforms baseline methods, achieving strong improvements in perceptual quality (e.g., LPIPS = 0.087 and FID = 6.633 on BKK). Analysis of the learned policy reveals a three-phase adaptation strategy—initial structural learning, perceptual refinement, and stable convergence, demonstrating the agent's ability to adjust supervision dynamically during training. Ablation studies confirm that the combined reward formulation effectively balances structural, perceptual, and distributional objectives, while optimal hyperparameters are identified as a 16-epoch PPO update interval and 512×512 input resolution. Deviations from these settings lead to reduced performance, highlighting their importance for stable training. Reward trajectory analysis further shows that the PPO agent learns a consistent and stable policy across datasets.

Future work will extend the proposed framework in several directions. First, its robustness will be evaluated under more diverse environmental and imaging conditions, including variations in illumination, road texture, traffic density, weather, occlusion size, and scene geometry. Second, richer state representations will be investigated by incorporating not only the current loss values and training progress but also gradient statistics, temporal loss trends, and validation-quality indicators, which may provide the controller with a more informative description of the training dynamics. Third, the framework will also be integrated with newer inpainting architectures, such as transformer- and diffusion-based models, to examine whether the learned weighting strategy generalizes beyond the current backbones. Finally, cross-dataset and cross-city evaluations

will be conducted to determine whether the learned loss-weighting policy can transfer to unseen mobile-mapping environments without extensive retuning.

Funding Statement: The research was funded by King Mongkut's University of Technology North Bangkok, Contract no. **KMUTNB-68-BASIC-42**.

Supplementary Materials: The source code, experimental configurations, and supporting datasets used in this study are provided as supplementary materials accompanying this article. <https://github.com/BrzKung/enhanced-pix2pix-adaptive-weight-loss>

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Chen, Z., Badrinarayanan, V., Lee, C.-Y., & Rabinovich, A. (2018). GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning* (Vol. 80, pp. 794–803). <https://proceedings.mlr.press/v80/chen18a.html>
2. Christovam, L. E., Shimabukuro, M. H., Galo, M. de L. B. T., & Honkavaara, E. (2022). Pix2pix conditional generative adversarial network with MLP loss function for cloud removal in a cropland time series. *Remote Sensing*, 14(1), 144. <https://doi.org/10.3390/rs14010144>
3. Darabi, S., Shechtman, E., Barnes, C., Goldman, D. B., & Sen, P. (2012). Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics*, 31(4), Article 82. <https://doi.org/10.1145/2185520.2185578>.
4. Elhashash, M., Albanwan, H., & Qin, R. (2022). A review of mobile mapping systems: From sensors to applications. *Sensors*, 22(11), 4262. <https://doi.org/10.3390/s22114262>
5. Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.1508.06576>
6. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (Vol. 27, pp. 2672–2680). https://proceedings.neurips.cc/paper_files/paper/2014/hash/f033ed80deb0234979a61f95710dbe25-Abstract.html
7. Han, S. W., & Suh, D. Y. (2021). A 360-degree panoramic image inpainting network using a cube map. *Computers, Materials & Continua*, 66(1), 213–228. <https://doi.org/10.32604/cmc.2020.012223>
8. Heydari, A. A., Thompson, C. A., & Mehmood, A. (2019). SoftAdapt: Techniques for adaptive loss weighting of neural networks with multi-part loss functions [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.1912.12355>
9. Hore, A., & Ziou, D. (2010). Image quality metrics: PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition* (pp. 2366–2369). <https://doi.org/10.1109/ICPR.2010.579>
10. Ruzic, T., & Pizurica, A. (2015). Context-aware patch-based image inpainting using Markov random field modeling. *IEEE Transactions on Image Processing*, 24(1), 444–456. <https://doi.org/10.1109/TIP.2014.2372479>
11. Isola, P., Zhu, J.-Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1125–1134). <https://doi.org/10.1109/CVPR.2017.632>
12. Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *Computer vision – ECCV 2016* (pp. 694–711). Springer. https://doi.org/10.1007/978-3-319-46475-6_43
13. Li, W., Lin, Z., Zhou, K., Qi, L., Wang, Y., & Jia, J. (2022). MAT: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 10758–10768). <https://doi.org/10.1109/CVPR52688.2022.01049>
14. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., & Van Gool, L. (2022). RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11461–11471). <https://doi.org/10.1109/CVPR52688.2022.01117>

15. Mansourifar, H., & Simske, S. J. (2023). GAN-based object removal in high-resolution satellite images [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2301.11726>
16. Mysore Badarinarayana, A. K., Pratsch, C., Lunkenbein, T., & Jug, F. (2025). Pattern matching-based denoising for images with repeated sub-structures. *Machine Learning and Knowledge Extraction*, 7(2), Article 34. <https://doi.org/10.3390/make7020034>
17. Petrie, G., & Toth, C. K. (2009). Introduction to laser ranging, profiling, and scanning. In J. Shan & C. K. Toth (Eds.), *Topographic laser ranging and scanning: Principles and processing* (pp. 1–28). CRC Press. <https://doi.org/10.1201/9781420051438.ch1>
- Schulman, J., Moritz, P., Levine, S., Jordan, M. I., & Abbeel, P. (2016). High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*. <https://arxiv.org/abs/1506.02438>
18. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.1707.06347>
19. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press.
20. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., & Lempitsky, V. (2022). Resolution-robust large mask inpainting with Fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (pp. 2149–2159). <https://doi.org/10.1109/WACV51458.2022.00323>
21. Uittenbogaard, R., Sebastian, C., Vijverberg, J., Boom, B., Gavrilu, D. M., & de With, P. H. N. (2019). Privacy protection in street-view panoramas using depth and multi-view imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 10581–10590) <https://doi.org/10.1109/CVPR.2019.01083>
22. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5505–5514). <https://doi.org/10.1109/CVPR.2018.00577>
23. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2019). Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4470–4479). <https://doi.org/10.1109/ICCV.2019.00457>
24. Yu, L., Gao, Y., Pakdaman, F., & Gabbouj, M. (2024). Panoramic image inpainting with gated convolution and contextual reconstruction loss. In *ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4255–4259). IEEE. <https://doi.org/10.1109/ICASSP48485.2024.10446469>
25. Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 586–595). <https://doi.org/10.1109/CVPR.2018.00068>
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2818–2826). <https://doi.org/10.1109/CVPR.2016.308>