Improved Land Cover Classification of Aerial Imagery and Satellite Image Time Series using Diffusion-based Super-Resolution

Hubert Kanyamahanga, Mareike Dorozynski, Franz Rottensteiner

Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover – Germany
(kanyamahanga, dorozynski, rottensteiner)@ipi.uni-hannover.de

Keywords: Land Cover Classification, Diffusion Models, Transformers, Fully Convolutional Networks, Multi-scale Data Fusion

Abstract

Accurate land cover classification requires both spatial details and temporal information of remote sensing data. While publicly available satellite image time series (SITS) offer short revisit times, they suffer from limited spatial resolution. In contrast, aerial imagery provides fine-grained spatial details, but its temporal coverage is limited. Thus, combining data from those sensors is of interest, because their properties are complementary w.r.t. the problem domain. However, the large gap in spatial resolution between these two sensors makes their integration challenging. Generating super-resolution-SITS (SR-SITS) before fusion can help to reduce this gap. In this work, we propose a new approach that integrates diffusion models for generating SR-SITS into a method for the joint pixel-wise classification of aerial and SITS data. Specifically, we employ a diffusion model to generate SR-SITS at an intermediate resolution from the raw SITS and aerial imagery of the same observed area. The SR-SITS are temporally encoded and fused with the aerial features using a cross attention module to produce pixel-wise classification at the geometrical resolution of the aerial image. Experimental results on the existing FLAIR benchmark dataset indicate that our approach achieves state-of-the-art results, with a mean Intersection over Union score of 64.0% and an overall accuracy of 76.6%.

1. Introduction

Remote sensing (RS) imagery plays a vital role in numerous applications, including the classification of land cover (LC). Satellite image time series (SITS), such as those from Sentinel-2, provide a good temporal coverage for capturing seasonal changes, but their limited spatial resolution (e.g. 10 m Ground Sampling Distance (GSD) for Sentinel-2) makes it difficult to detect small structures. Conversely, aerial imagery offers much higher spatial resolution (e.g., 20 cm GSD), but typically with much lower revisit times. Therefore, there is a need for designing approaches that can effectively integrate SITS and aerial imagery to fully leverage their respective advantages.

The integration of multi-scale RS data is typically achieved by using separate networks for each data source before fusing their outputs. For aerial and satellite images acquired at a single point in time, Fully Convolutional Networks (FCNs) such as U-Net (Ronneberger et al., 2015) or transformer-based architectures such as the Vision Transformer (ViT) (Dosovitskiy et al., 2021) are commonly used. For SITS, temporal dependencies are additionally modeled by attention-based modules, e.g. (Garioud et al., 2024; Kanyamahanga and Rottensteiner, 2024; Heidarianbaei et al., 2024). In all cases, the SITS must be upsampled by a large factor for the fusion (e.g., 50 for Sentinel-2 images and aerial data with 20 cm GSD). Moreover, fusion is often performed at the lowest-resolution to reduce computational cost, limiting the integration of multi-scale information and exploitation of fine spatial details in the fusion (Garioud et al., 2024).

A potential approach to reduce the scale gap between satellite and aerial imagery is to use super-resolution (SR) techniques (Donike et al., 2025), which aim to reconstruct higher resolution (HR) images from images having a coarser resolution. SR techniques have evolved from simple interpolation methods to advanced deep learning frameworks such as convolutional neural networks (CNNs) and generative adversarial networks (GANs) (Kapilaratne et al., 2022). GANs have been widely used due to their ability to generate HR images that closely resemble real data. However, they are difficult to train and frequently introduce artifacts that can negatively affect downstream tasks. More recently, diffusion-based generative models have emerged as a promising alternative, achieving good results in image synthesis and SR across various domains including RS (Okabayashi et al., 2024; Donike et al., 2025). Diffusion models aim to reconstruct missing high-frequency texture details in lower-resolution (LR) data using HR images. This is achieved by constraining the generation process on other data, e.g. images or reference labels, guiding the model to produce HR images that are consistent with the input. Existing diffusion approaches for SR are typically designed to generate a single HR image, either from a single acquisition (Donike et al., 2025; Wang and Sun, 2025) or from an image time series (Okabayashi et al., 2024). Consequently, information about temporal variations in appearance is not preserved. Moreover, their reliance on RGB channels limits their ability to exploit the rich spectral information present in multi-spectral data. To our knowledge, no existing work has employed diffusion models to generate a *time series of satellite images*, nor has explored their use for the integration of aerial and Sentinel-2 data for LC classification.

Another key challenge lies in the fusion of multi-scale data. A common approach involves element-wise addition of features, which is simple and computationally efficient (Garioud et al., 2024; Heidarianbaei et al., 2024), but often fails to account for the varying relevance of each data source across different spatial and temporal contexts, leading to suboptimal results. Attention-based approaches, which learn to weight features based on their contextual importance, have been shown to achieve a better performance (Kanyamahanga et al., 2025). In these cases, however, attention is computed using only features at the lowest resolution, limiting the ability of the model to capture fine spatial details and integrate multi-scale contextual information.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-517-2026

To address these limitations, we present a new method for the integration of aerial and SITS data to predict pixel-wise LC at the GSD of the aerial image. Our method introduces a Temporal Super-Resolution Diffusion (TSRDiff) block that generates a SR-SITS from a LR-SITS in order to bridge the resolution gap between SITS and aerial imagery. Each image of the LR time series is upsampled to an intermediate resolution GSD^{SR}. The HR aerial images are used in training so that the network can learn to generate these SITS at a higher resolution, while auxiliary losses help to maintain both spatial alignment and temporal consistency of the SITS. The SR and classification components are trained jointly, ensuring that the generated images are not only visually realistic but also semantically meaningful. The features learned from aerial and SITS data are combined on the basis of a new cross-attention fusion module, allowing one modality to complement the other across multiple scales.

The scientific contributions of this paper are fourfold. (1) We propose a new method for combining aerial imagery and SITS for LC classification, using generative diffusion models for SR to reduce the resolution gap between the two data sources. (2) We introduce auxiliary losses to support the diffusion model in capturing high-frequency details and maintaining radiometric consistency, thereby improving the quality of generated SR-SITS. (3) We introduce a fusion module based on cross-attention to facilitate the integration of complementary information from aerial and SITS data at multiple scales. (4) We demonstrate through extensive experiments that our approach improves classification accuracy across various LC types, performing competitively with existing approaches.

2. Related Work

Diffusion models for super-resolution of SITS: Diffusion models (Ho et al., 2020) are probabilistic generative models that progressively transform random noise drawn from a simple distribution (e.g., a Gaussian) into complex data distributions such as HR images. Compared to GANs (Kapilaratne et al., 2022), which are difficult to train and often produce images with limited diversity (Romero et al., 2020), diffusion models offer a more stable training procedure and have been shown to generate high-quality and diverse images (Li et al., 2022). Diffusion models have been used for single-image SR of satellite data, often requiring paired LR–HR images for training (Wang and Sun, 2025). These approaches primarily rely on RGB bands, often neglecting valuable spectral information from other bands.

Wang and Sun (2025) employed latent diffusion models (Rombach et al., 2022) conditioned on a combination of a single-date Sentinel-2 image (10 m GSD) and semantic maps from Open Street Map to reconstruct a HR image of a single epoch. The HR images used in training were obtained from World-View with a GSD of 1 m. Similarly, Donike et al. (2025) employed latent diffusion models conditioned on multi-spectral Sentinel-2 data, with SPOT-6/7 imagery (2.5 m GSD) serving as HR data. Unlike Wang and Sun (2025), they additionally use the NIR band, thus providing spectral information beyond RGB. However, none of the approaches cited so far generates SR-SITS.

Okabayashi et al. (2024) adapted the SRDiff model of Li et al. (2022) by introducing a temporal-attention encoder (Garnot and Landrieu, 2020) to incorporate a Sentinel-2 SITS for generating a single HR image of 2.5 m GSD, using a single-date SPOT-6 image (2.5 m GSD) in training and considering RGB channels only. Although this approach produces high-quality images, it does not recover temporal patterns in the output that could help to separate classes with different variations of appearance over time. Taking advantage of this information at a higher resolution would require an approach that can generate a SR-SITS data from LR-SITS instead of a single image. In addition, the data used for training by Okabayashi et al. (2024) are limited to agriculture areas, raising questions about the model's generalization to more heterogeneous or urban landscapes.

Multi-scale fusion of aerial and SITS data: Garioud et al. (2024) present an approach to exploit the complementary strengths of aerial images and SITS based on a two-branch U-Net architecture: A U-Net with a Temporal self-Attention Encoder (U-TAE) is used to encode temporal features from the SITS, which are fused with aerial features from a U-Net encoder to predict pixel-wise class scores. Fusion is performed via element-wise addition of aerial and SITS features in the skip connections, with SITS features being upsampled to higher resolutions. Straka and Gruber (2024) improved the classification performance by training multiple models with different encoders in the aerial branch, while keeping the SITS encoder and the fusion from (Garioud et al., 2024) unchanged. However, this comes at the cost of increased computation and training complexity, limiting scalability in resource-constrained scenarios. Despite these advances, a major challenge remains in effectively aligning information across modalities. The resolutions of aerial and SITS data may very well differ by a factor of 50. This may cause feature-level inconsistencies, as upsampling may blur the spatial information contained in the features derived from the SITS. Moreover, fusion by summation or concatenation treats features from both modalities as equally informative, potentially limiting the model's ability to learn their relative contributions to the classification of LC.

Unlike Garioud et al. (2024), Kanyamahanga et al. (2025) employed a cross-attention fusion module to enable information exchange between aerial and SITS features. A hybrid temporal encoder combining convolutions and attentions is used to extract spatio-temporal features from SITS, while a SegFormer (Xie et al., 2021) encodes aerial images for pixel-wise predictions. Fusion is applied to the SITS features and the corresponding aerial features, which are those of the lowest resolution, and the fused features are integrated into the skip connections. Although the method showed improved performance compared to Garioud et al. (2024), it does not fully take advantage of the temporal variations at higher spatial resolutions.

Discussion: The method proposed in this paper aims to address the challenges of complex LC classification by reducing the resolution gap between SITS and aerial imagery, thereby facilitating their joint integration. To the best of our knowledge, none of the existing approaches have used diffusion models to generate a sequence of SITS data to further combine it with HR information from aerial imagery, with the goal to predict LC at the GSD of the aerial image. The work most closely related to ours is (Okabayashi et al., 2024), which, however, neither generates SR-SITS data nor addresses the task of LC classification. Thus, we extend (Okabayashi et al., 2024) by introducing a temporal diffusion module to generate a SR-SITS, considering the NIR band in addition to RGB channels. We also introduce auxiliary losses to support the diffusion model to generate more realistic SR-SITS. To fuse features from aerial and SITS data, we develop a new cross-attention module that enables multi-scale interaction, allowing SITS features to complement aerial features across different spatial resolutions.

3. Background: Super-Resolution with Diffusion Models

Denoising Diffusion Probabilistic Models generate data by learning to reverse a process that gradually adds noise to training samples (Ho et al., 2020). The image generation process is modeled as consisting of a forward and a reverse process. The forward process progressively corrupts a HR image $\boldsymbol{x}_0$ over a sequence of diffusion steps $k = 1, ..., K$, with Gaussian noise according to variance schedule $\beta_1, ..., \beta_K$, producing a sequence of noisy samples $\boldsymbol{x}_1, ..., \boldsymbol{x}_k, ..., \boldsymbol{x}_K$:

$$q(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) = \mathcal{N}\Big(\boldsymbol{x}_k;\ \sqrt{1-\beta_k}\cdot\boldsymbol{x}_{k-1},\ \beta_k\cdot\mathbf{I}\Big), \quad (1)$$

where $q(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})$ is the transition probability between $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k-1}$, $\mathcal{N}$ is the normal distribution, the variance $\beta_k \in (0, 1)$ controls the amount of noise added at step k, and $\mathbf{I}$ is a unit matrix. Using $\alpha_k = 1 - \beta_k, \quad \bar{\alpha}_k = \prod_{s=1}^{k} \alpha_s$, this model of the forward process allows $\boldsymbol{x}_k$ to be sampled at any timestep k based on a random noise vector $\boldsymbol{\varepsilon}$ sampled from a Gaussian:

$$\boldsymbol{x}_k = \sqrt{\bar{\alpha}_k}\cdot\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_k}\cdot\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}). \quad (2)$$

Image generation requires the probability $p_{\boldsymbol{w}}(\boldsymbol{x}_{k-1} \mid \boldsymbol{x}_k)$ of the reverse process, parameterized by $\boldsymbol{w}$. For small β_k, $p_{\boldsymbol{w}}$ can also be modeled by a Gaussian, i.e. $p_{\boldsymbol{w}}(\boldsymbol{x}_{k-1} \mid \boldsymbol{x}_k, \boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}_{k-1};\ \boldsymbol{\mu}_{\boldsymbol{w}},\ \boldsymbol{\Sigma}_{\boldsymbol{w}})$, with mean $\boldsymbol{\mu}_{\boldsymbol{w}} = \boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{x}_k, k)$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{w}}$. Ho et al. (2020) propose the following parameterization for $\boldsymbol{\mu}_{\boldsymbol{w}}$:

$$\boldsymbol{\mu}_{\boldsymbol{w}}(\boldsymbol{x}_k, k) = \frac{1}{\sqrt{\alpha_t}}\cdot\left(\boldsymbol{x}_k - \frac{\beta_k}{\sqrt{1-\bar{\alpha}_k}}\cdot\hat{\varepsilon}_{\boldsymbol{w}}\right), \quad (3)$$

where $\hat{\varepsilon}_{\boldsymbol{w}}$ is a noise component predicted by a neural network $\varepsilon_{\boldsymbol{w}}(\boldsymbol{x}_k, k)$ parameterized by $\boldsymbol{w}$. For the covariance, Ho et al. (2020) propose to use $\boldsymbol{\Sigma}_{\boldsymbol{w}} = \tilde{\beta}_k \cdot \boldsymbol{I}$ with $\tilde{\beta}_k = \beta_1$ for $k = 1$ and $\tilde{\beta}_k = \frac{1-\bar{\alpha}_{k-1}}{1-\bar{\alpha}_k}$ for $k > 1$, which is constant. In training, for a random training image $\boldsymbol{x}_0$ and a random step k, eq. 2 is used to determine $\boldsymbol{x}_k$ based on a random vector $\boldsymbol{\varepsilon}$. Then, a vector $\boldsymbol{\varepsilon}_{\boldsymbol{w}} = \varepsilon_{\boldsymbol{w}}(\boldsymbol{x}_k, k)$ is predicted by the network, and the loss becomes $||\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{\boldsymbol{w}}||$; the parameters are updated based on the gradient of this loss. Thus, the network is trained to predict the noise contaminating the image $\boldsymbol{x}_0$. In inference, $\boldsymbol{x}_K$ is sampled from a standard Gaussian, $\boldsymbol{x}_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and then for $k = K - 1, \ldots, 1$, $\boldsymbol{x}_{k-1}$ is iteratively sampled from $p_{\boldsymbol{w}}(\boldsymbol{x}_{k-1} \mid \boldsymbol{x}_k)$ (Ho et al., 2020). Using $\mu_{\boldsymbol{w}}(\boldsymbol{x}_k, k)$ according to eq. 3 and $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ drawn from a Gaussian, this leads to:

$$\boldsymbol{x}_{k-1} = \mu_{\boldsymbol{w}}(\boldsymbol{x}_k, k) + \sqrt{\tilde{\beta}_k}\cdot\boldsymbol{\eta}, \quad (4)$$

Li et al. (2022) use the method just described to generate a HR image $\boldsymbol{x}^{HR}$ from a LR image $\boldsymbol{x}^{LR}$. For inference, the input $\boldsymbol{x}^{LR}$ is upsampled to a higher resolution by resampling. The output becomes $\boldsymbol{x}^{HR} = up\left(\boldsymbol{x}^{LR}\right) + \boldsymbol{x}_0$, where $\boldsymbol{x}_0$ is generated by a diffusion model. The main difference is that in every generation step, the network $\boldsymbol{\varepsilon}_{\boldsymbol{w}}$ in eq. 4 and eq. 3 takes features $\boldsymbol{z} = En\left(\boldsymbol{x}^{LR}\right)$ obtained from the LR image via an encoder $En(\cdot)$ as an additional input: $\boldsymbol{\varepsilon}_{\boldsymbol{w}} = \varepsilon_{\boldsymbol{w}}(\boldsymbol{x}_k, \boldsymbol{z}, k)$. Consequently, the mean in eq. 3 and eq. 4 becomes $\mu_{\boldsymbol{w}}(\boldsymbol{x}_k, \boldsymbol{z}, k)$, i.e. it also depends on $\boldsymbol{z}$. Training requires pairs of corresponding LR and HR images. For each such pair, the $\boldsymbol{x}_0 = \boldsymbol{x}^{HR} - up\left(\boldsymbol{x}^{LR}\right)$ forms the basis for an update step in a way similar to the one explained earlier, the difference yet again being that the network to be trained now also depends on the encoded LR features $\boldsymbol{z}$ (Li et al., 2022).

4. Methodology

We start the description of our method by presenting the network architecture in Section 4.1. Section 4.2 discusses training.

4.1 Network Architecture

4.1.1 Overview: Our proposed network architecture is shown in Figure 1. The input consists of a single HR image $\boldsymbol{x}^{HR}$ of shape $C^{HR} \times H^{HR} \times W^{HR}$, having a spatial extent of $H^{HR} \times W^{HR}$ and comprising C^{HR} channels, and a co-registered LR-SITS $\boldsymbol{x}^{LR}$ with dimensions $T \times C^{LR} \times H^{LR} \times W^{LR}$, encoding T temporal acquisitions, each with C^{LR} channels and a spatial extent $H^{LR} \times W^{LR}$. The GSDs of the two input datasets are GSD^{HR} and GSD^{LR}, respectively. We assume that the spectral bands in $\boldsymbol{x}^{HR}$ are also those of $\boldsymbol{x}^{LR}$; if the original SITS has more bands, those not having a correspondence in the HR image are discarded. The HR image may have additional bands, e.g. a normalized digital surface model (nDSM). The output of the network consists of a LC map of dimension $H^{HR} \times W^{HR}$, i.e. at the GSD of the HR image.

The network architecture consists of two main branches: the *LR branch*, which processes the LR-SITS, and the *HR branch*, which processes the HR image. They are connected by a *fusion module*. The *LR branch* consists of two main components: the *Temporal Super-Resolution Diffusion (TSRDiff) block* and the *SR Feature Extractor*. The TSRDiff block applies a diffusion model to the LR-SITS to predict a SR-SITS $\hat{\boldsymbol{x}}^{SR}$ at a resolution $GSD^{SR} = 2^{S_f} \cdot GSD^{HR}$ that it matches the GSD of the features extracted at an appropriate stage S_f of the HR encoder (cf. Section 4.1.2); this corresponds to an increase of the resolution by a factor $f_u = \frac{GSD^{LR}}{2^{S_f} \cdot GSD^{HR}}$. The SR Feature Extractor is an encoder-decoder network and processes the SR-SITS; its decoder is only used in training (cf. Section 4.2).

The *HR branch*, described in Section 4.1.4, is an encoder-decoder network with skip connections that processes the HR image, predicting the final output of the network, a label map at GSD^{HR}. Its decoder does not only process the features generated by its encoder: the *fusion module* applies a cross-attention mechanism to determine a joint representation of both input modalities from the outputs of the encoders of both the HR branch and the LR feature extractor. These fused features form an additional input of the decoder of the HR branch at multiple scales. The fusion module is described in Section 4.1.5.

4.1.2 TSRDiff Block: This block is based on (Okabayashi et al., 2024), which encodes a SITS into a single feature map using L-TAE (Garnot and Landrieu, 2020) and integrates it into the SRDiff model (Li et al., 2022) for predicting a single HR image. We extend this model such that it generates a SITS. The encoder E_C generates a latent representation $\boldsymbol{z}^{LR}$ from the LR-SITS. We model E_C using the T-ConvFormer of Kanyamahanga et al. (2025) with three stages. In each stage, it uses several blocks combining convolutions in the spatial dimensions and transformers in the temporal one, before downsampling the features by a factor of 2 (Voelsen et al., 2024). Unlike Kanyamahanga et al. (2025) we do not merge the features of different epochs at the end of the decoder, so that the output $\boldsymbol{z}^{LR}$ contains a multi-scale feature map $\boldsymbol{z}^{\tau,LR}$ for every epoch $\tau \in \{1, \ldots, T\}$.

The encoded features along with the HR input image are processed by a diffusion model for SR similar to the one of Li et al.

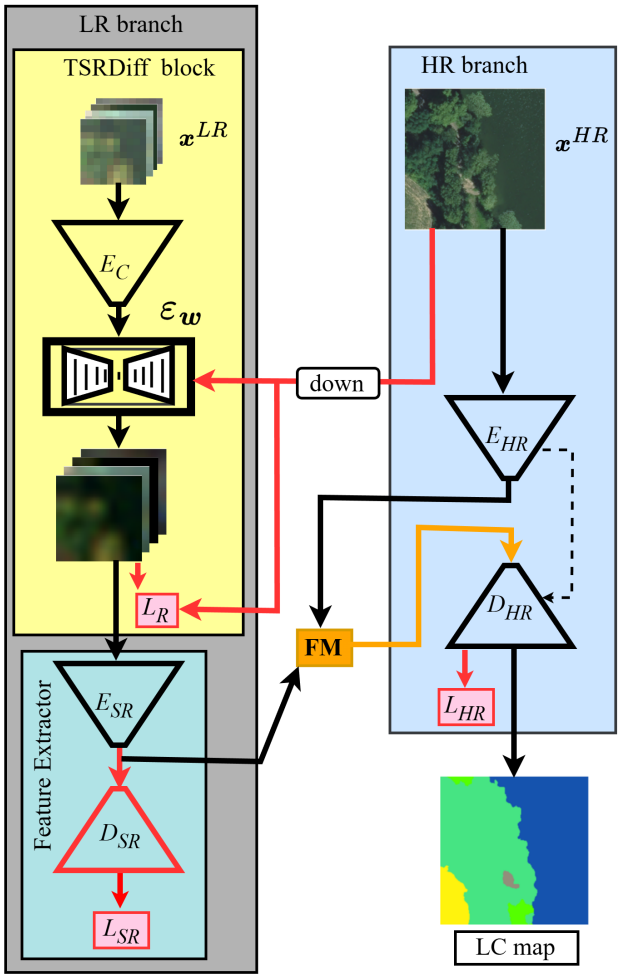

Figure 1. Network architecture of the proposed method. The LR branch (gray) processes a LR-SITS $\boldsymbol{x}^{LR}$. It is upsampled to the resolution GSD^{SR} using the TSRDiff module (yellow) consisting of an encoder E_C and a diffusion network $\boldsymbol{\varepsilon_w}$. It results in a SR-SITS $\hat{\boldsymbol{x}}^{SR}$ that is processed by the SR Feature Extractor (cyan). The HR branch (light blue), consisting of an encoder E_{HR} and a decoder D_{HR} with skip connections (broken arrow), processes the HR image $\boldsymbol{x}^{HR}$. The features generated by E_{SR} and E_{HR} are combined in the fusion module FM (orange), and the fused features form an additional input to D_{HR}. Finally, the HR decoder D_{HR} predicts the class labels. Red components are only active during training. In training, the decoder D_{SR} of the LR Feature Extractor predicts class labels at the resolution of the SR-SITS. The losses L_R, L_{SR} and L_{HR} are described in Section 4.2. Training the diffusion module also requires a downsampled version of the HR image (red arrow).

(2022) outlined in Section 3, but generating a time series in the reverse process. First, the input SITS is upsampled to GSD^{SR} using bicubic interpolation, yielding $\boldsymbol{x}_u$; $\boldsymbol{z}^{LR}$ is upsampled in the same way, yielding a feature map $\boldsymbol{z}$ containing a multi-scale feature map $\boldsymbol{z}^\tau$ for every epoch τ. Similarly to the procedure outlined in Section 3, the diffusion model has to predict the difference between the desired output $\hat{\boldsymbol{x}}^{SR}$ and $\boldsymbol{x}_u$, but in our case, this is another time series $\boldsymbol{x}_0$ consisting of T images $\boldsymbol{x}_0^\tau$. The output $\hat{\boldsymbol{x}}^{SR}$ thus becomes $\hat{\boldsymbol{x}}^{SR} = \boldsymbol{x}_u + \boldsymbol{x}_0$.

To generate $\boldsymbol{x}_0$, we apply the inference procedure based on the reverse process described in Section 3 to every epoch τ independently. Given the number K of diffusion steps, for every epoch τ, we sample a random image $\boldsymbol{x}_K^\tau$ and iteratively adapt it using the reverse process, using the update rule according to eq. 3 in every step k, but using the data of epoch τ for computing the mean. Thus, the mean becomes $\mu_{\boldsymbol{w}} = \mu_{\boldsymbol{w}}(\boldsymbol{x}_k^\tau, \boldsymbol{z}^\tau, k)$ and is defined according to eq. 3, but additionally depends on $\boldsymbol{z}^\tau$. In the last iteration of the reverse process, the noise $\boldsymbol{\eta}$ is set to zero. The output SR-SITS $\hat{\boldsymbol{x}}^{SR}$ has the dimensionality $T \times C^{LR} \times H^{SR} \times W^{SR}$, where the spatial dimensions (H^{SR}, W^{SR}) depend on the upsampling factor f_u, and T as well as C^{LR} are identical to those of the input.

Inference requires a network module $\varepsilon_{\boldsymbol{w}}(\boldsymbol{x}^\tau, \boldsymbol{z}^\tau)$ to predict $\hat{\varepsilon}_{\boldsymbol{w}}$ in eq. 3. We adapt the network architecture of Li et al. (2022) for that purpose. At step k of inference, its input consists of the current image $\boldsymbol{x}_k^\tau$, the encoded LR features $\boldsymbol{z}^\tau$, and k. Li et al. (2022) first use a convolutional block to adapt the feature dimension of $\boldsymbol{x}^\tau$ to the one of $\boldsymbol{z}^\tau$ (which, in their case, only consists of features at a single scale) and then apply a U-Net to the sum $\boldsymbol{x}^\tau + \boldsymbol{z}^\tau$. The U-Net consists of several corresponding encoder and decoder layers with skip connections. A final 1×1 convolution maps the features to the predicted noise $\hat{\varepsilon}_{\boldsymbol{w}}$ required for the update step (eq. 3 and eq. 4). We use the same network design, but consider the fact that $\boldsymbol{z}^\tau$ contains features at multiple scales. For defining the input of the U-Net, we use the features at the highest resolution in the same way as Li et al. (2022), but additionally, in every stage of the decoder we add the feature map in $\boldsymbol{z}^\tau$ at the corresponding resolution to the output of the previous encoder step of the U-Net, thus integrating the encoded features at multiple encoder levels.

4.1.3 LR Feature Extractor: This block consists of an encoder E_{SR} and a decoder D_{SR}, the latter only being used in training. E_{SR} has the same structure as E_C, but it depends on different parameters. Also, the feature maps of individual epochs are averaged at each scale, because only one label map is to be predicted (Kanyamahanga et al., 2025). The number of stages in E_{SR} depends on the structure of the encoder of the HR branch, because only feature maps at corresponding resolutions can be fused (cf. Section 4.1.5); we use three stages in our experiments. The output of E_{SR} consists of a feature map $\boldsymbol{F}_e^{SR}$ for each encoding stage e, with $e \in \{1, 2, 3\}$ in our setting. The dimensionality of $\boldsymbol{F}_e^{SR}$ is $C_e^{SR} \times H^{SR}/2^{e-1} \times W^{SR}/2^{e-1}$. In inference, these features form the input of the fusion module (cf. Section 4.1.5). In training, they are also processed by a UPerNet decoder D_{SR} (Xiao et al., 2018) to generate pixel-wise class scores at GSD^{SR}, which are used for auxiliary supervision (cf. Section 4.2.1).

4.1.4 HR branch: This branch is an encoder-decoder network that processes a HR image $\boldsymbol{x}^{HR}$ to produce a pixel-wise LC map, considering input from the fusion module. We use MaxViT (Tu et al., 2022) with five stages as the encoder E_{HR}. In the first stage, $\boldsymbol{x}^{HR}$ is processed by a convolution block, resulting in a feature map $\boldsymbol{F}_1^{HR}$ at half the spatial resolution of the input. All subsequent stages $s \in \{2, \ldots, 5\}$ process the output of the previous stage by a sequence of local and global transformer blocks, followed by downsampling by a factor of 2 (Tu et al., 2022). This results in 5 feature maps $\boldsymbol{F}_s^{HR}$ of dimension $C_s^{HR} \times H^{HR}/2^s \times W^{HR}/2^s$, where s denotes the stage. They are fused with those generated by the SR encoder E_{SR} at corresponding spatial resolutions (cf. Section 4.1.5). Given the definition of GSD^{SR} (cf. Section 4.1.1, the feature map corresponding to $\boldsymbol{F}_s^{HR}$ is $\boldsymbol{F}_e^{SR}$ with stage index $e = s - (S_f - 1)$. Thus, fusion is applied to the HR features of all stages $s \geq S_f$,

while the features of the other stages are introduced into the decoder via normal skip connections. We denote the fused features corresponding to stage s of E_{HR} by $\boldsymbol{F}_s^f$.

The decoder D_{HR} is similar to a U-Net decoder, but adapted to allow for the integration of the results of the fusion module. It consists of 4 stages. In the stages $s \geq S_f$, the feature map from the previous stage $s+1$ (in the first stage ($s=4$) we use the feature map $\boldsymbol{F}_5^f$ generated by the fusion module instead) is upsampled by a factor of 2 using bilinear resampling before being concatenated with the feature map $\boldsymbol{F}_s^f$ generated by the fusion of $\boldsymbol{F}_s^{HR}$ and $\boldsymbol{F}_{s-(S_f-1)}^{SR}$. The output of that stage is generated by processing the concatenated features by two convolution layers, reducing the feature dimension to the one of the input. In the last $S_f - 1$ decoder stages, no fusion is applied; instead, the upsampled feature map from the previous decoder stage $s-1$ is concatenated with the output of the corresponding encoder stage s before processing the concatenated features by convolutions. The output of the last decoder stage ($s = 1$) is upsampled by a factor of 2, yielding a feature representation at the resolution of the input image, GSD^{HR}. A 1×1 convolution generates raw class scores that are normalized by a softmax layer. The LC map is derived from the normalized class scores.

4.1.5 Fusion Module: At stage s of the HR branch, this module fuses the HR feature maps $\boldsymbol{F}_s^{HR}$ generated by E_{HR} at stage s and $\boldsymbol{F}_{s-(S_f-1)}^{SR}$ generated by E_{SR} at stage $e = s - (S_f - 1)$. Thus, the set of stages at which features are fused depends on S_f (cf. Section 4.1.1). We choose S_f so that $\frac{GSD^{SR}}{GSD^{HR}} = 2^{S_f} \approx f_u$. In our experiments (cf. Section 5), we have $\frac{GSD^{LR}}{GSD^{HR}} = 50$. This leads to $S_f = 3$, corresponding to $GSD^{SR} = 1.6\ m$ and $f_u = 6.25$. Thus, fusion is applied to the features at stages 5, 4 and 3 of E_{HR} and the results will be used in stages 4 and 3 of D_{HR}, while normal skip connections are applied in stages 1 and 2 of D_{HR} (cf. Section 4.1.4).

Fusion is based on multi-head cross-attention, allowing HR features to selectively attend to relevant temporal information. First, the feature maps are reshaped to form sequences. Then, a linear layer is used to map the sequence of HR features to the query matrix $\boldsymbol{Q}^{HR}$, while the key and value matrices $\boldsymbol{K}^{SR}$ and $\boldsymbol{V}^{SR}$, are generated by linearly projecting the sequence of SR features. These matrices form the input to a multi-head cross-attention block MHA according to (Vaswani et al., 2017):

$$\boldsymbol{W}\boldsymbol{F}_s = MHA\left(\boldsymbol{Q}^{HR}, \boldsymbol{K}^{SR}, \boldsymbol{V}^{SR}\right). \tag{5}$$

Using the HR features as queries allows the model to selectively extract temporal information from the SR data that is most relevant for each spatial location. As in common transformer blocks (Vaswani et al., 2017), a residual connection is applied, and after a layer normalization (LN), the resultant intermediate features $\boldsymbol{F}_{int}$ are processed by an MLP consisting of two linear layers with a ReLU activation in between. A second residual connection followed by another layer normalization yields the final fused features $\boldsymbol{F}_s^f$:

$$\begin{aligned} \boldsymbol{F}_{int} &= LN\left(\boldsymbol{W}\boldsymbol{F}_s + \boldsymbol{F}_s^{HR}\right) \\ \boldsymbol{F}_s^f &= LN\left(MLP\left(\boldsymbol{F}_{int}\right) + \boldsymbol{F}_{int}\right), \end{aligned} \tag{6}$$

which form input of the decoder D_{HR}.

4.2 Training

We start the training procedure by pre-training the encoder E_C of the TSRDiff block for N_{PT} epochs (we use $N_{PT} = 30$). For that purpose it is extended by a U-Net decoder similar to D_{SR}, and the network is trained by minimizing a classification loss defined by eq. 8 using reference labels downsampled by majority vote. The denoising U-Net $\varepsilon_{\boldsymbol{w}}$ is initialized following He et al. (2015), while E_{SR} is initialized using the parameters of E_C and the HR encoder E_{HR} is initialized with weights pre-trained on ImageNet-1k (Tu et al., 2022). The remaining components, i.e. the decoders (D_{SR}, D_{HR}) and the fusion module, are initialized randomly according to He et al. (2015). The total number of diffusion steps K is a hyperparameter (set to $K = \mathbf{500}$ in our experiments). Starting from this initialization, all components of the network except E_C (which is frozen) are trained by minimizing a loss function L consisting of three components: two classification loss terms L_{HR} and L_{SR} for the outputs of the HR branch and the SR Feature Extractor, respectively, and a SR reconstruction loss L_R for the TSRDiff block. The total loss L is a weighted sum of these three components, with the weights $\lambda_{HR}, \lambda_{SR}, \lambda_R$ in $[0, 1]$:

$$L = \lambda_{HR} \cdot L_{HR} + \lambda_{SR} \cdot L_{SR} + \lambda_R \cdot L_R. \tag{7}$$

The classification losses L_{HR} and L_{SR} are described in Section 4.2.1, while L_R is presented in Section 4.2.2.

The loss L is minimized using Adam (Kingma and Ba, 2015). In each iteration, a minibatch of N_{MB} pairs ($\boldsymbol{x}^{HR}$, $\boldsymbol{x}^{LR}$), each consisting of one HR image $\boldsymbol{x}^{HR}$ and one SITS $\boldsymbol{x}^{LR}$, is processed by the network before the gradient of the loss is used to update the network parameters. For every pair, the forward pass starts by encoding $\boldsymbol{x}^{LR}$ by E_C, yielding features $\boldsymbol{z}^{LR}$. The initial upsampling steps of $\boldsymbol{x}^{LR}$ and $\boldsymbol{z}^{LR}$ described in Section 4.1.2 is applied, yielding upsampled versions $\boldsymbol{x}_u$ and $\boldsymbol{z}$ of these data. However, the diffusion process is performed in way that differs from inference. First, the HR image is downsampled to GSD^{SR}, yielding $\boldsymbol{x}_0^D$, and a random diffusion step k is selected. Also, the noise component $\boldsymbol{\varepsilon}$ (eq. 2) is randomly drawn. Then, for each timestep τ, a residual image $\boldsymbol{x}_{res}^\tau = \boldsymbol{x}_0^D - \boldsymbol{x}_u^\tau$ is determined. This image is contaminated by $\boldsymbol{\varepsilon}$ according to eq. 2, yielding the contaminated image $\boldsymbol{x}_{res,k}^\tau$ corresponding to the diffusion step k. Using a convolutional layer, this image is mapped to a tensor $\bar{\boldsymbol{x}}_{res,k}^\tau$ having the same feature dimensionality as $\boldsymbol{z}$, and the noise image $\hat{\boldsymbol{\varepsilon}}_{\boldsymbol{w}}^\tau = \hat{\boldsymbol{\varepsilon}}_{\boldsymbol{w}}\left(\bar{\boldsymbol{x}}_{res,k}^\tau, \boldsymbol{z}^\tau\right)$ is predicted by the diffusion network. This noise image and $\boldsymbol{x}_{res,k}^\tau$ are used to predict the residual image $\boldsymbol{x}_0^{SR,\tau}$ according to Li et al. (2022), which is added to the upsampled LR image of that timestep to yield the corresponding SR image: $\hat{\boldsymbol{x}}^{SR,\tau} = \boldsymbol{x}_0^{SR,\tau} + \boldsymbol{x}_u^\tau$. In this step of the training procedure, some additional intermediate results required to compute the SR reconstruction loss L_R will be determined as well, as will be described in Section 4.2.2.

The SR-SITS $\hat{\boldsymbol{x}}^{SR}$ predicted in the way just described is processed by the encoder E_{SR}, yielding a feature map $\boldsymbol{F}_e^{SR}$ for every stage e (cf. Section 4.1.3). These features are used by the decoder D_{SR} to predict class scores at GSD^{SR}, which form the input to the SR classification loss L_{SR}. The HR image $\boldsymbol{x}^{HR}$ is processed by the encoder E_{HR}, which computes a feature map $\boldsymbol{F}_s^{SR}$ for every stage s (cf. Section 4.1.4). These feature maps are fused with corresponding maps $\boldsymbol{F}_e^{SR}$ from the LR branch and then processed by the HR decoder, as described in Sections 4.1.4 and 4.1.5. Finally, the output of the HR decoder is used to evaluate the classification loss L_{HR} (cf. Section 4.2.1).

4.2.1 Classification losses (L_{HR}, L_{SR}): We use the focal loss (Lin et al., 2017) to model both classification losses in eq. 7, L_{HR} and L_{SR}, because it is supposed to mitigate problems in scenarios with a high class imbalance:

$$L_{Focal} = -\frac{1}{N_P}\sum_{i=1}^{N_P}\sum_{c=1}^{N_C} t_{i,c}\cdot(1-p_{i,c})^{\gamma}\cdot\log(p_{i,c}). \quad (8)$$

In eq. 8, N_P denotes the number of pixels in the mini-batch for which the loss is computed, i is the index of a pixel, N_C is the number of classes, and c is the index of a specific class. The variable $t_{i,c}$ indicates whether pixel i corresponds to class c in the reference ($t_{i,c}=1$) or not ($t_{i,c}=0$). The term $p_{i,c}$ is the softmax score for pixel i to correspond to class c. The term $(1-p_{i,c})^{\gamma}$, controlled by the focusing parameter γ, will increase the weight of the loss for pixels with large softmax scores for a wrong class, which often occurs for underrepresented classes. L_{HR} corresponds to L_{Focal} applied to the softmax scores determined at the end of the HR Decoder, D_{HR}, while for L_{SR}, it is applied to the output of the Decoder of the SR Feature Extractor, requiring a label map that is downsampled to GSD^{SR} by majority vote.

4.2.2 SR reconstruction loss: The SR reconstruction loss L_R combines multiple loss terms:

$$L_R = \alpha_D\cdot L_D + \alpha_C\cdot L_C + \alpha_\nabla\cdot L_\nabla + \alpha_T\cdot L_T + \alpha_S\cdot L_S, \quad (9)$$

each loss term L_* associated with a weight α_*. All of them compare a grid $\hat{\boldsymbol{y}}$ based on the results of the prediction of the network to another grid $\boldsymbol{y}$ of the same size derived from a reference. Denoting individual elements of these grids by $\hat{y}_i$ and y_i, respectively, and using $||\boldsymbol{y}||$ to denote the number of elements of $\boldsymbol{y}$, all of these loss terms are based on the L_1 loss which evaluates the mean absolute deviation of $\hat{\boldsymbol{y}}$ from $\boldsymbol{y}$:

$$L_1(\hat{\boldsymbol{y}},\boldsymbol{y},) = \frac{1}{||\boldsymbol{y}||}\sum_{i=1}^{||\boldsymbol{y}||}|\hat{y}_i - y_i|. \quad (10)$$

The individual terms are explained in the following paragraphs.

Diffusion loss L_D: This is the main loss term for learning the parameters of the diffusion network. For every pair in a mini-batch, we determine the time step τ_{HR} of the image in the SITS which was acquired most closely to the HR image. As in (Li et al., 2022), the loss is computed by comparing the noise image $\boldsymbol{\varepsilon}$ to the predicted one, in our case the one predicted for the time τ_{HR}. This results in $L_D = L_1(\hat{\boldsymbol{\varepsilon}}_{\boldsymbol{w}}^{\tau_{HR}}, \boldsymbol{\varepsilon})$, with $L_1(\cdot)$ according to eq. 10. The remaining loss terms in eq. 9 are considered to be auxiliary losses.

Radiometric consistency loss L_C: This loss is designed to minimize the pixel-wise difference between the SR image predicted for the timestep τ_{HR} (see earlier) and its HR reference counterpart. It is computed for all common spectral bands $b \in B$, where B denotes the set of spectral bands of the SITS and the downsampled HR image $\boldsymbol{x}_0^D$. The loss becomes

$$L_C = \frac{1}{||B||}\cdot\sum_{b\in B} L_1\left(\hat{\boldsymbol{x}}_b^{SR,\tau_{HR}}, \boldsymbol{x}_{0,b}^D\right), \quad (11)$$

where the subscript b indicates an image band. This loss is expected to help the model retain local texture information.

Gradient consistency loss L_∇: This loss is designed to emphasize a similarity of the local image structure of the image $\hat{\boldsymbol{x}}^{SR,\tau_{HR}}$ to the downsampled HR image $\boldsymbol{x}_0^D$. We compute the gradient norms $||\nabla\hat{\boldsymbol{x}}_b^{SR,\tau_{HR}}||$ and $||\nabla\boldsymbol{x}_{0,b}^D||$ for every band b. The loss becomes:

$$L_\nabla = \frac{1}{||B||}\cdot\sum_{b\in B} L_1\left(||\nabla\hat{\boldsymbol{x}}_b^{SR,\tau_{HR}}||, ||\nabla\boldsymbol{x}_{0,b}^D||\right). \quad (12)$$

By emphasizing on changes rather than on absolute pixel values, this loss allows the model to reproduce sharp edges and fine structural details without contaminating the spectral properties.

Temporal consistency loss L_T: This loss penalizes differences in spatial gradient magnitude between consecutive epochs τ and $\tau+1$ of the SR-SITS $\hat{\boldsymbol{x}}^{SR}$, computed for every spectral band b. Using $M=(T-1)\cdot||B||$, the loss becomes

$$L_T = \frac{1}{M}\cdot\sum_{\tau\in T}\sum_{b\in B} L_1\left(||\nabla\hat{\boldsymbol{x}}_b^{SR,\tau}||, ||\nabla\hat{\boldsymbol{x}}_b^{SR,(\tau+1)}||\right). \quad (13)$$

This constraint ensures that the generated SR-SITS sequence maintains coherent transitions over time, penalizing abrupt or unrealistic changes in spatial structure.

Spectral loss L_S: This loss minimizes the discrepancy in spectral values between a SR-SITS $\hat{\boldsymbol{x}}^{D,SR}$ that is a downsampled version of the SR output and the LR-SITS $\boldsymbol{x}^{LR}$. It is computed for all spectral bands b for each epoch τ. The loss becomes:

$$L_S = \frac{1}{T\cdot||B||}\cdot\sum_{\tau=0}^{T}\sum_{b\in B} L_1\left(\hat{\boldsymbol{x}}_b^{D,SR,\tau}, \boldsymbol{x}_b^{LR,\tau}\right). \quad (14)$$

This loss ensures that the SR images maintain the radiometric properties of the LR input when being downsampled to the original GSD.

5. Experiments

5.1 Dataset

Our experiments are based on the French Land Cover from Aerospace ImageRy (FLAIR) #2 Challenge dataset (Garioud et al., 2024). It includes co-registered mono-temporal aerial imagery, nDSMs, and Sentinel-2A SITS for 916 areas across France. The aerial images consist of four spectral channels (RGB and NIR) with $GSD^{HR} = 0.2\ m$. For the TSRDiff block, only the RGB and NIR channels are used. For the HR branch, the spectral channels are combined with a nDSM at the same resolution, this results in $C^{HR}=5$ channels for the HR input $\mathbf{x}^{HR}$. For the SITS with $GSD^{LR} = 10\ m$, we use the $C^{LR}=4$ spectral channels available in the aerial images. Satellite images with more than 5% cloud cover were discarded, leaving between 20 and 110 valid time steps per area. Similar to (Garioud et al., 2024), we apply temporal monthly averaging after cloud filtering to reduce input redundancy and mitigate short-term acquisition artifacts, resulting in a SITS with a varying number of timesteps (T); the maximum is 12. These SITS constitute our LR input $\mathbf{x}^{LR}$. The land cover reference is provided at the pixel level of the aerial image and comprises 13 LC categories: *building (bld.)*, *pervious surface (pvs.)*, *impervious surface (ips.)*, *bare soil (bs.)*, *water (wt.)*, *coniferous (cfs.)*, *decidous (dcs.)*, *brushwood (bsd.)*, *vineyard (vyd.)*, *herbaceous*

vegetation (hvg.), *agricultural land (agr.)*, *plowed land (pld.)*, and a class *other* corresponding to unknown LC. The dataset is highly imbalanced, with class frequencies ranging from 1.1% (*other*) to 19.8% (*hvg*). Each area is divided into tiles covering 512 x 512 pixels at GSD^{HR}. For each of the 77,762 tiles, the corresponding SITS is cropped to 40×40 pixels at GSD^{LR}, centered on the aerial tile. The dataset is split into training, validation and testing subsets according to a 63% / 17% / 20% split. More details can be found in (Garioud et al., 2024).

5.2 Experimental Setup

The goal of the experiments is to evaluate the performance of the proposed classifier taking a HR image and LR-SITS as input. In particular, we want to validate the hypothesis that the new modules improve the classification performance compared to other fusion approaches as well as methods only using unimodal data. For that purpose, the method described in Section 4 is applied to the data presented in 5.1. The spatial dimensions of the input aerial and SITS data are $H^{HR} = W^{HR} = 512$ and $H^{LR} = W^{LR} = 40$, respectively. Data augmentation is applied in training using random rotations by 90°, 180°, 270°, horizontal and vertical flipping. We train the entire network for up to 400k iterations, using $K = 500$ diffusion steps and a batch size of $N_{MB} = 8$. We use the Adam optimizer (Kingma and Ba, 2015) with the parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for minimizing the loss in eq. 7. The initial learning rate is $1e^{-3}$, which is reduced using learning rate decay. The weights for the loss terms in equations 7 and 9 are set to $\lambda_{HR} = 0.7$, $\lambda_{SR} = 0.3$, $\lambda_R = 1.0$ and $\alpha_D = 0.6$, $\alpha_C = 0.5$, $\alpha_\nabla = 0.7$, $\alpha_T = 0.3$ and $\alpha_S = 0.5$, respectively. These values were selected based on the best performance on the validation dataset. All the models are implemented using PyTorch Lightning and trained on a cluster with two NVIDIA A100 80GB GPUs.

All the experiments follow the same experimental protocol. We compare our approach with seven baseline methods and perform an ablation study with respect to the introduced key components. Our baselines are summarized in Table 1. The first one, U-Net (Garioud et al., 2022), predicts LC only based on the aerial image. This is also true for MaxViT, which is a standalone version of our HR branch. U-T&T (Garioud et al., 2024), T-CF-S (Kanyamahanga et al., 2025) and Ensemble (Straka and Gruber, 2024) are three baseline methods that integrate aerial and SITS data. The model U-TAE-D extends U-T&T by our TSRDiff block, fusing the SR and HR features at corresponding stages. Finally, the baseline T-CF-M applies our method without the TSRDiff block, using raw SITS as input. In this case, all encoded features are upsampled to a GSD of 6.4 m before being fused with the aerial features.

Our ablation study is based on our proposed network and consists of two sets of experiments: The first set compares our cross-attention fusion approach with alternative fusion techniques; all of these experiments are conducted with L_D, i.e. without the auxiliary losses in eq. 9. The first experiment (Base) employs element-wise addition of features. The second variant (Base+CWC), uses channel-wise concatenation. The third one (Base+CA) uses cross-attention. The second set of experiments aims to evaluate the impact of the individual auxiliary losses by adding them one after the other in the order in which they occur in eq. 9, thus extending the setting of the experiment Base+CA.

To evaluate the results, we report the intersection over union (IoU_c) for each class c, the mean intersection over union (*mIoU*) and the overall accuracy (*OA*). Each experiment was repeated three times, using different random initializations and stochastic batch shuffling. We present the mean evaluation metrics and standard deviations across the three test runs.

Name	Aerial	SITS
U-Net	U-Net (RsN)	✘
MaxViT	MaxViT	✘
U-T&T	U-Net (RsN)	U-TAE
T-CF-S	SegFormer	T-CF
Ensemble	U-Net (RNX & MiT)	U-TAE
U-TAE-D	U-Net (RsN)	TSRDiff & U-TAE
T-CF-M	MaxViT	T-CF
Ours	MaxViT	TSRDiff & T-CF

Table 1. Overview of the experiments conducted to compare our approach to baseline methods. Name: name of the experiment. Aerial / SITS: network architectures used for processing the aerial image and the SITS, respectively. We abbreviate the names of the models: ResNet-34 as RsN, ResNeXt as RNX, MiT-B2 as MiT and T-ConvFormer as T-CF.

5.3 Results

Table 2 presents the *mIoU* and OA of the results of our method and the baselines presented in Table 1. Overall, our method achieves a *mIoU* of 64.0% and an OA of 76.6%. These are the second best metrics, but the difference to the best metrics, achieved by the Ensemble method of Straka and Gruber (2024), is only 0.1% and 0.2% in *mIoU* and OA, respectively, which is in the order of magnitude of the standard deviation of these metrics and thus not significant. The Ensemble method combines predictions from four separate models by averaging their outputs, which increases robustness through model diversity but comes at the cost of significantly higher computational complexity. Despite using a single model, our method achieves the same level of performance without sacrificing accuracy.

Method	**mIoU [%]**	**OA [%]**	**Params [M]**
U-Net	55.2 ± 0.32	71.3 ± 0.14	24.4
MaxViT	61.6 ± 0.26	74.8 ± 0.29	40.5
U-T&T	56.8 ± 2.76	71.7 ± 1.55	27.3
T-CF-S	60.1 ± 1.50	74.3 ± 1.38	135.0
Ensemble	**64.1** ± 0.11	**76.8** ± 0.30	140.2
U-TAE-D	57.1 ± 0.12	71.8 ± 0.14	70.2
T-CF-M	61.8 ± 0.60	75.2 ± 0.48	71.0
Ours	64.0 ± 0.02	76.6 ± 0.17	127.0

Table 2. Mean IoU ($mIoU$) and Overall Accuracy (OA) [%] achieved by the methods in Table 1 on the FLAIR #2 test set. Params [M]: number of parameters of the model in millions. Values represent the average and standard deviation over three independent runs. Best results are shown in **bold** font.

As far as the other methods are concerned, our approach outperforms the baselines only using aerial imagery. In *mIoU*, the improvement is +8.8% compared to U-Net. Our MaxViT-based HR branch achieves a value that is already 6.4% better than U-Net in *mIoU*, which shows the benefits of using a state-of-the-art encoder; in particular, MaxViT outperforms three methods integrating SITS, namely U-T&T, the multi-modal SegFormer-based model T-CF-S and the U-TAE model trained on SR-SITS (U-TAE-D). The U-TAE-D achieves a marginal gain of +0.3% compared to the model trained on the original SITS upsampled with bilinear interpolation (U-T&T) while performing significantly worse than our full model. The variant of our method without the TSRDiff block, T-CF-M, achieves insignificantly better results than MaxViT. However, the integration of the TSRDiff block yields a significant improvement of 2.4% in

$mIoU$ and 1.8% in OA compared to MaxViT-based HR branch alone. This underscores the impact of the diffusion component in leveraging temporal information from SITS to compensate for the limited temporal resolution of aerial imagery.

Table 3 presents the class-wise IoU scores obtained by our method and the baselines on the FLAIR #2 test set. Our proposed method achieves the highest *IoU* scores for six classes, outperforming models trained exclusively on aerial images (e.g., U-Net, MaxViT) and SITS-based models (e.g., U-T&T, T-CF-S). Notably, our methods brings consistent gains across classes with temporal dynamics such as *agr*, *vyd*, and *hvg*. In these cases, the margin by which our approach surpasses aerial image-based approaches is typically in the range of 2–7% in *IoU*. Compared to an ensemble model of (Straka and Gruber, 2024), our model performs competitively on the same level or even better in most vegetation classes except for two classes (*dcs* and *cfs*), where our model is outperformed by the former. While the difference for *dcs* is relatively small 1%, the gap is more noticeable for *cfs*, where our approach lags behind by 3.7%. This behaviour can be explained by the tendency of diffusion models to oversmooth fine textural details which are crucial for distinguishing coniferous canopies from other vegetation types. Nevertheless, these results indicate that the temporal information from SITS can help to reduce the confusion between classes that change over time.

Another important aspect influencing model performance is class imbalance. As shown in Table 3, some classes are easier to differentiate than others. In the FLAIR #2 dataset, some land-cover types, particularly *bare soil*, *coniferous* and *vineyards* are underrepresented, whereas *herbaceous vegetation* and *deciduous* are comparatively dominant. Notably, our approach outperforms the second best model on underrepresented classes such as *bs* and *vyd*, with modest *mIoU* gains of 1.7% and 0.2%, respectively. However, this advantage does not extend to *cfs*, where our method is outperformed by a large margin, the reasons having been discussed earlier. This indicates that further refinement is needed to improve the model’s ability to generalize to highly variable or complex classes.

Qualitative results for several compared models are presented in Figures 2 and 3, representing regions with different class distribution (rural and urban, respectively). The results show that models incorporating SITS data are more effective in identifying vegetation classes (*agr*, *hvg* and *dcs*). This is further confirmed by a visual analysis of Figure 2, showing how vegetation classes are better classified by our approach (see regions indicated in black squares). In contrast, the results presented in Figure 3 indicate that, although there are notable differences in the *mIoU* between the baselines (U-Net, and U-T&T) and the best-performing approaches (Ensemble and ours) for non-vegetation classes such as *bld*, *ips*, and *pvs*, the visual distinctions across those methods are relatively small. Nevertheless, it can be observed that all the objects types are clearly identified; buildings and roads connecting them and green spaces near residential areas. Figure 2 also indicates that fully delineating vegetation classes remains challenging. All compared approaches encountered difficulties in differentiating *herbaceous vegetation* from *agricultural lands*, most likely because of their high similarity in spectral and textural characteristics, leading to classification inaccuracies (cf. areas in black circles). Overall, despite some misclassification problems among vegetation classes where class mixing is observed, our results indicate the benefits of our approach in classifying LC.

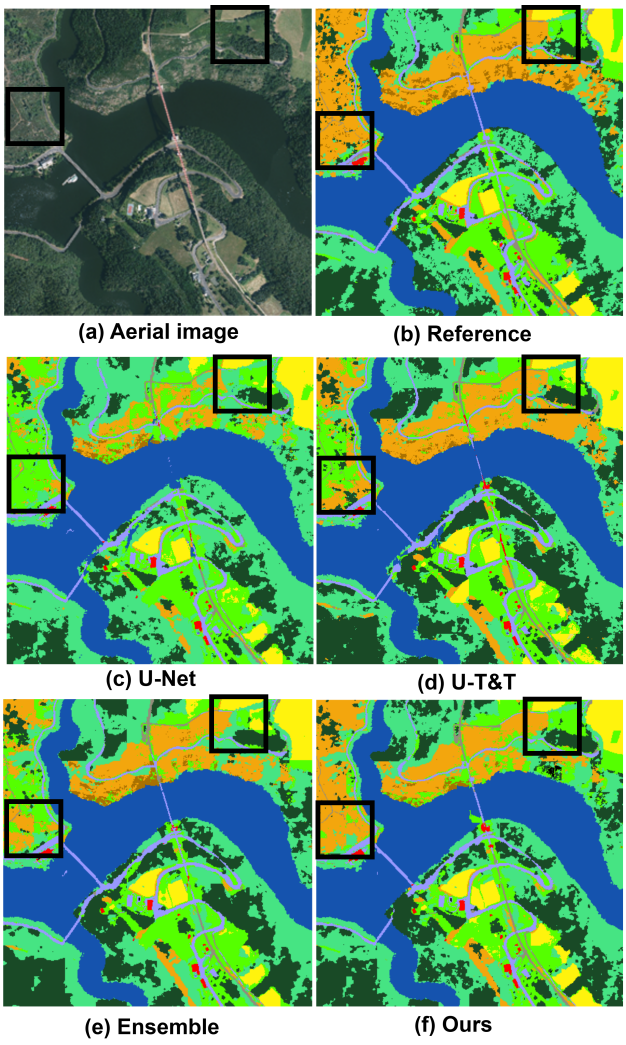

Figure 2. Aerial image of a rural test area, the reference and the LC maps predicted by four selected methods. The depicted area consists of several tiles that were classified independently. Compared methods are denoted by the acronyms in (c) – (f). Colours: red - *bld*, grey - *pvs*, light blue - *ips*, brown - *bs*, blue - *wt*, dark green - *cfs*, acquamarine - *dcs*, orange - *bsd*, purple - *vyd*, bright green - *hvg*, yellow - *agr*, dark yellow - *pld*.

Table 4 presents the results of our ablation study, comparing our full method to several variants. The table shows that the integration of cross-attention fusion and auxiliary supervision improve the performance. The Base model, which uses element-wise addition for fusing aerial and SITS features, achieves the lowest *mIoU* of 62.2%. Introducing channel-wise concatenation (Base+CWC) leads to a minor improvement of 0.3%. When a cross-attention fusion (Base+CA) is applied, the *mIoU* increases by approximately 0.8%, indicating that the cross-attention approach is to be preferred over other fusion strategies. This improvement can be attributed to the ability of cross-attention approaches to dynamically learn the relationships between features from different modalities, allowing the model to focus on the most relevant information during fusion.

As far as auxiliary supervision is concerned, we found that introducing individual auxiliary losses generally leads to performance improvements. Adding L_C provides a modest gain of +0.2% in the *mIoU* over the baseline Base+CA. The gradient-

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-XI-2-2026-517-2026

Models	IoU [%]											
	bld.	*pvs.*	*ips.*	*bs.*	*wt*	*cfs.*	*dcs.*	*bsd.*	*vyd.*	*hvg.*	*agr.*	*pld.*
U-Net	81.8	49.2	72.8	40.5	85.0	41.1	68.7	23.9	62.2	48.4	53.0	35.5
MaxViT	84.9	57.3	73.9	59.7	88.7	59.6	73.0	27.5	68.3	48.7	55.5	41.9
U-T&T	81.9	48.6	71.9	43.4	83.2	56.9	69.8	25.6	65.1	46.0	53.3	36.6
T-CF-S	81.3	50.6	73.0	42.4	80.5	55.4	71.2	23.9	65.2	45.5	54.1	38.9
Ensemble	85.9	**60.0**	**76.3**	62.7	91.0	**68.4**	**75.7**	**29.6**	69.5	49.0	57.8	**43.0**
U-TAE-D	83.9	52.3	73.5	53.4	86.4	50.7	70.3	25.1	56.6	44.5	50.6	37.4
T-CF-M	85.1	58.3	74.6	64.0	91.1	66.9	74.3	26.8	67.6	50.9	56.6	39.9
Ours	**86.4**	59.5	76.2	**64.4**	**92.4**	64.7	74.7	28.8	**69.7**	**51.0**	**58.8**	41.6

Table 3. Class-wise IoU values [%] on the test set of the FLAIR #2 dataset achieved by our method and the baselines defined in Table 1. The numbers are averages achieved by three independent test runs. The corresponding standard deviations are excluded for the lack of space. Best results are indicated in **bold** font.

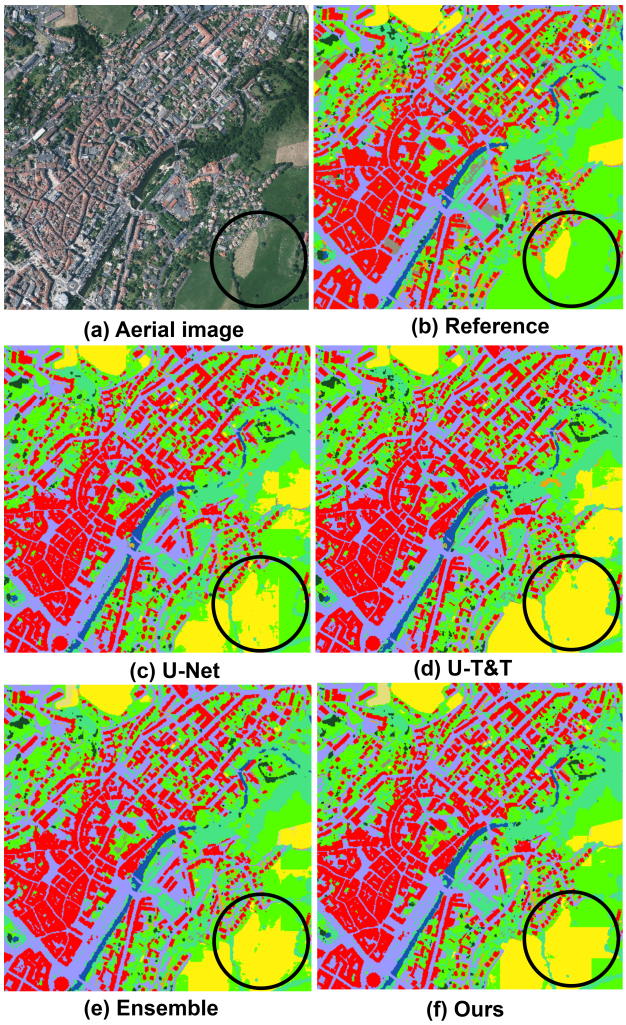

Figure 3. Aerial image of an urban test area, the corresponding reference and the LC maps predicted by four selected methods. The depicted area consists of several tiles that were classified independently. Compared methods are denoted by the acronyms in (c) – (f). Colour code: cf. Figure 2

based loss L_∇ yields the largest performance gain, increasing the *mIoU* to 63.9%. In contrast, incorporating L_T results in a slight decrease of -0.5% in the *mIoU*, indicating that the diffusion model may oversmooth informative temporal signals if that loss is used. This may occur in scenarios where land surfaces change rapidly over short time periods (e.g, crop growth cycles or harvesting events). Finally, adding L_S partially compensates for this performance drop, recovering nearly +0.2%

Method	mIoU [%]	OA [%]
Base	62.2 ± 0.11	75.5 ± 0.22
Base+CWC	62.5 ± 0.23	75.7 ± 0.27
Base+CA	63.0 ± 0.13	76.1 ± 0.19
Base+CA+L_C	63.2 ± 0.51	76.2 ± 0.37
Base+CA+$L_C + L_\nabla$	63.9 ± 0.16	76.5 ± 0.34
Base+CA+$L_C + L_\nabla + L_T$	63.4 ± 0.25	76.3 ± 0.21
Base+CA+$L_C + L_\nabla + L_T + L_S$	63.6 ± 0.06	76.4 ± 0.09
Ours	64.0 ± 0.02	76.6 ± 0.17

Table 4. Evaluation of our ablation studies. The first three experiments are related to three different fusion methods, all using L_D but no auxiliary losses for supervising the TSRDiff block. The subsequent experiments add one auxiliary loss after the other to Base+CA.

in the *mIoU*. In the overall, our ablation study indicates that the use of diffusion models to enhance SITS together with our proposed cross-attention and auxiliary supervision components leads to a significant performance gain of about 1.8% compared to the method Base. The auxiliary losses improve the *mIoU* by 1% compared to the method only using L_D for supervising the TSRDiff branch.

6. Conclusion

In this work, we presented a new method that uses diffusion models to enhance the spatial resolution of SITS, thereby facilitating the joint integration of aerial and SITS data for LC classification. Our results show that using complementary information from aerial and SITS data leads to state-of-the-art result, with a *mIoU* of 64.0% and an OA of 76.6%. We see the most significant impact of our proposed method on classes such as *bare soil*, *coniferous*, *vineyard*, *agricultural land* that are especially affected by seasonal variations. The block for increasing the spatial resolution of SITS data leads to an improvement of 2.2% in $mIoU$, partly to be attributed to the auxiliary losses. The new fusion module also improves the results slightly.

Future research could investigate combining this approach with other HR satellite imagery such as SPOT-6 (Garioud et al., 2025), offering more detailed information about the LC up to 1.6 m GSD. Such integration can potentially improve the delineation of complex LC classes, addressing the misclassification problems present in the current approaches. Alternatively, exploring other diffusion model variants such as latent diffusion models (LDMs) (Wang and Sun, 2025), which operate in a learned latent space while capturing key semantic structures may help preserve high-level features. Leveraging pretrained diffusion models on large-scale datasets may further enhance performance by providing rich transferable knowledge and im-

proving model generalization. Another aspect worth investigating is the use of the full temporal spectrum of SITS to represent the phenological cycle of vegetation in a better way.

References

Donike, S., Aybar, C., Gómez-Chova, L., Kalaitzis, F., 2025. Trustworthy super-resolution of multispectral sentinel-2 imagery with latent diffusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 18, 6940–6952.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*.

Garioud, A., Giordano, S., David, N., Gonthier, N., 2025. FLAIR-HUB: Large-scale Multimodal Dataset for Land Cover and Crop Mapping. *arXiv preprint arXiv:2506.07080*.

Garioud, A., Gonthier, N., Landrieu, L., De Wit, A., Valette, M., Poupée, M., Giordano, S., 2024. Flair: A country-scale land cover semantic segmentation dataset from multi-source optical imagery. *Advances in Neural Information Processing Systems (NIPS)*, 36, 16456–16482.

Garioud, A., Peillet, S., Bookjans, E. M., Giordano, S., Wattrelos, B., 2022. FLAIR #1: Semantic segmentation and domain adaptation dataset. *ArXiv*, abs/2211.12979.

Garnot, V. S. F., Landrieu, L., 2020. Lightweight temporal self-attention for classifying satellite images time series. *International Workshop on Advanced Analytics and Learning on Temporal Data*, Springer, 171–181.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *IEEE International Conference on Computer Vision (ICCV)*, 1026–1034.

Heidarianbaei, M., Kanyamahanga, H., Dorozynski, M., 2024. Temporal ViT-U-Net Tandem Model: Enhancing Multi-Sensor Land Cover Classification Through Transformer-Based Utilization of Satellite Image Time Series. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10, 169–177.

Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NIPS)*, 33, 6840–6851.

Kanyamahanga, H., Dorozynski, M., Rottensteiner, F., 2025. Classification of Satellite Image Time Series and Aerial Images Based on Multiscale Fusion and Multilevel Supervision. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-G-2025, 477–486.

Kanyamahanga, H., Rottensteiner, F., 2024. Land Cover Classification based on Multiscale Time Series of Satellite and Aerial Images. *Proceedings, 44th Annual Scientific and Technical Conference of the DGPF in Remagen*, 32, 223–235.

Kapilaratne, R. G. C. J., Kakuta, S., Kaneta, S., 2022. Enhanced Super Resolution for Remote Sensing Images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-3-2022, 53–60.

Kingma, D. P., Ba, J., 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.

Li, H., Yang, Y., Chang, M., Chen, S., Feng, H., Xu, Z., Li, Q., Chen, Y., 2022. SRDiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479, 47–59.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. *IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

Okabayashi, A., Audebert, N., Donike, S., Pelletier, C., 2024. Cross-sensor super-resolution of irregularly sampled sentinel-2 time series. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 502–511.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., 2022. High-resolution image synthesis with latent diffusion models. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 10684–10695.

Romero, S. L., Marcello, J., Vilaplana, V., 2020. Super-resolution of sentinel-2 imagery using generative adversarial networks. *Remote Sensing*, 12(15), 2424.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, 234–241.

Straka, J., Gruber, I., 2024. Modernized training of U-Net for aerial semantic segmentation. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 776–784.

Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y., 2022. Maxvit: Multi-axis vision transformer. *European Conference on Computer Vision (ECCV)*, Springer, 459–479.

Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I., 2017. Attention is All you Need. *Advances in Neural Information Processing Systems (NIPS)*, 30, 5998–6008.

Voelsen, M., Rottensteiner, F., Heipke, C., 2024. Transformer models for Land Cover Classification with Satellite Image Time Series. *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 92(5), 547–568.

Wang, C., Sun, W., 2025. Semantic guided large scale factor remote sensing image super-resolution with generative diffusion prior. *ISPRS Journal of Photogrammetry and Remote Sensing*, 220, 125–138.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J., 2018. Unified perceptual parsing for scene understanding. *European Conference on Computer Vision (ECCV)*, 418–434.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., Luo, P., 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems (NIPS)*, 34, 12077–12090.