

IMPERIAL

**Refining Gridded Asset
Exposure Estimates with
Auxiliary Co-variates and
Aggregated Supervision**

MLBD MRes Project Report

Project Code: MLBD_2024_06

Name: Ping-Yen, Chuang

CID: 06002664

Supervisor: Prof. Ralf Toumi

Assessor: Prof. Heather Graven

Word Count: 9906 (not including layperson's summary)

Date of Submission: 22nd, August, 2025

Layperson's summary

Imagine you're trying to figure out how much damage a hurricane might cause. A key first step is knowing exactly what's in its path: where all the buildings are and what they're worth. This is known as "**asset exposure**" data, and it's essential for preparing for natural disasters like earthquakes, floods, and storms.

While some organizations, like the Global Earthquake Model (GEM) Foundation, have created global maps of building values, these maps often have problems when you look at them up close. The data can be questionable based on broad assumptions such as admin-level statistics rather than detailed local information. Structurally, a single dot on a map might represent an entire group of buildings, but we don't know their exact locations or shapes. This can lead to inaccuracies: a valuable building compound might be assigned to one grid cell when it actually spans several. This "local noise" makes the data unreliable at the fine-grained level needed for precise disaster planning.

This project developed a new machine learning system to improve these imperfect maps. The core idea is to create a smarter way to refine and re-estimate asset values by using auxiliary information. Instead of just using the noisy GEM data, our system combines it with other widely available information from satellites, such as nighttime light intensity and census-based population count.

Our method starts by converting all the data into a common grid format with a resolution of 30 arcsec (approximately 1 km^2 per grid cell). Then, we train a deep learning model to look at the data not just for a single interested grid cell, but also the information in its surrounding area. The model's job is to learn the patterns that connect data from all resources, leveraging the local spatial context to support its predictions.

The crucial part of our approach is that we use a technique called "aggregated supervision," which is inspired by Multiple Instance Learning (MIL). We group the fine-grained grid cells into larger/courser "tiles". The model predicts the value for each small grid cell, but we only check if the total value for the entire tile is correct given that we expect the local noises would be averaged out across larger area. By training this way, the model learns to intelligently re-distribute the value from the noisy original GEM asset value map and use more colourful information to guide its estimates.

The model is also designed to resist being overly influenced by the presumably noisy original GEM data. As a part of main attempt, we explicitly discourages the model from becoming too dependent on the GEM input. This makes the predictions more stable and encourages the model to rely more on the auxiliary signals when the GEM data is less reliable. This creates a trust-weighted learning strategy where the model respects original values when they are reliable but doesn't blindly follow them when they're noisy.

Finally, our model is designed to estimate its own uncertainty. It produces not only a new, re-estimated value for each grid but also a measure of how confident it is in that prediction. This is important because it tells us where the output is most reliable and where we need to be cautious. To sum up, this new framework provides a more reliable, flexible, and understandable way to improve global asset exposure maps, particularly in places where good data is hard to find. This re-estimation strategy is expected to be applicable for improving other data especially when numbers are inferred from heavy spatial estimation and a validable ground truth is nearly impossible to access, such as attempts for high-resolution population mapping.

Contents

1	Introduction	3
1.1	Background and Motivation	3
1.2	Research Objective	3
1.3	Technical Approach Overview	4
1.4	Summary of Findings	5
2	Literature Review	6
2.1	Overview of the GEM Exposure Data and Its Role in Global Risk Assessment .	6
2.2	Aggregated Supervision via Multiple Instance Learning (MIR)	8
3	Method	9
3.1	Data collection, enhancement and transformation for skewed data distribution .	9
3.1.1	Gridding, extracting patches, vectorisation	9
3.1.2	Patch random rotation	10
3.1.3	Right-skewed distribution, logarithm transform	10
3.2	Deep learning model, attention mechanism	11
3.2.1	Model structure	11
3.3	Uncertainties	12
3.3.1	Non-deterministic Input Feature and Taylor Expansion Estimate	12
3.3.2	Out-of-distribution (OOD) detector	14
3.4	Loss function	14
3.4.1	Tile-level likelihood	14
3.4.2	Grid-level regularisation	15
3.4.3	GEM sensitivity penalty	15
4	Experiments and Results	17
4.1	Overview of Input Data	17
4.1.1	Night-time Light Data (Lit)	17
4.1.2	Population Data (Pop)	17
4.1.3	Asset Exposure Data (GEM)	17
4.1.4	Training Region and Input Channels	17
4.2	Model Training Setup	18
4.2.1	Training loop skeleton code	19
4.2.2	Hyperparameter Configuration	20
4.3	Training performance	21
4.4	Prediction outputs	22
4.4.1	Re-estimated asset value (mean)	22
4.4.2	Uncertainty prediction (standard deviation)	24
4.5	Sensitivity	24
5	Discussion, Analysis and Conclusion	27
5.1	Effect of hyperparameter α on Re-estimated Predictions	27
5.2	Evaluation of Predictive Uncertainty	28
5.3	Feature importance, sensitivity Analysis: Interpreting Learned Spatial Dependencies	29
5.4	Resilience to noisy GEM data input	30
5.5	Summary of findings	31
5.6	Future Work	32
References		33

Abstract

Quantifying asset exposure on the ground is a critical step in assessing the potential economic impact of natural disasters. While global datasets such as those provided by the Global Earthquake Model (GEM) Foundation offer valuable information on asset locations and values, these estimates are often uncertain at fine spatial scales.

This project proposes a machine learning framework to improve the reliability of asset exposure estimates by learning from GEM data in combination with publicly available auxiliary sources, such as satellite-based nighttime light intensity and population count. By converting all inputs into a gridded spatial format and training the model on a reference region, the approach aims to reduce sensitivity to potentially unreliable inputs and produce more stable, uncertainty-aware asset predictions at a fine spatial resolution by leveraging external auxiliary co-variates.

To account for the challenges of noisy supervision and uncertain ground truth, the model is designed to learn patterns over aggregated spatial tiles, allowing it to capture robust signals while softening the influence of locally inconsistent data. Additionally, the framework includes a mechanism to quantify aleatoric uncertainty and control the level of trust placed in each data source during training.

Beyond its immediate relevance for disaster risk modelling, the methodology developed here contributes to broader efforts in geo-spatial machine learning, particularly in settings where reliable ground truth is limited and robustness to noisy inputs is essential.

1 Introduction

1.1 Background and Motivation

Accurate, high-resolution asset exposure data is essential for understanding and mitigating the economic risks posed by natural disasters. In domains like flood modelling, typhoon damage estimation, and earthquake risk assessment, it is crucial to first quantify "what is on the ground", specifically, the replacement cost of built assets within a given region. In addition to many other efforts being made to achieve the goal [1] [2], the Global Earthquake Model (GEM) Foundation [3] has made significant advances in this area by providing comprehensive global exposure datasets that report asset centroids and corresponding values, generally reported in 2020 USD. These data products offer one of the most standardised resources available for global asset exposure estimation.

While valuable at a global scale, this data becomes suspiciously problematic at fine spatial resolutions due to both structural and statistical sources of local noise.

Structurally, each centroid represents a building compound or group of buildings, yet we lack access to the true building footprints. When rasterising these points into grid cells (30 arcsec), values are assigned based on centroid location, leading to spatial mismatch: some assets span multiple cells, while others may be missed entirely as illustrated in **Figure 1**. Statistically, GEM asset estimation uses a bottom-up modelling approach for total asset exposure at the highest administrative level, relying on the finest accessible statistics, then disaggregate the value with satellite-based proxies to infer exposure values [4]. In some areas, particularly when zoomed in, neighbour data points exhibit identical replacement cost values per square metre, revealing the use of shared proxies rather than locally observed differences. These modelling artefacts, combined with spatial imprecision, result in inconsistent or unreliable signals when interpreting the GEM data at the grid level, which having a decent resolution is especially crucial to estimation economic loss for certain types of disasters [5] [6].

1.2 Research Objective

This project aims to develop a machine learning framework that reduces dependence on the Global Earthquake Model (GEM) dataset for asset exposure estimation, by learning to exploit auxiliary covariates, specifically nighttime light intensity (Lit) and population count (Pop), which are available globally as open, gridded datasets. Although GEM provides centroids of

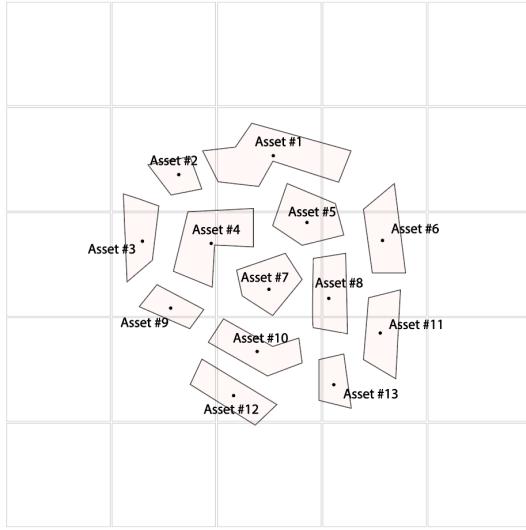


Figure 1: The GEM data came in with asset's representative point each tagged with its estimated value. However the building footprint (those polygons) are not accessible

asset clusters and their corresponding replacement cost values, these inputs are often uncertain or inconsistent at fine spatial resolution due to their modelling origin.

Therefore, The core objective of this project is to develop a robust, uncertainty-aware framework for estimating asset exposure at high spatial resolution (30 arcsec). By robust, we specifically mean that perturbations or inconsistencies in the GEM input data should not cause large fluctuations in the model predictions, especially when alternative data sources (e.g., population or night-time lights) offer more stable signals, we also expect the model to be trained with a more reliable ground truth [7].

To achieve this, we train the model to predict grid-level asset values, but supervise it at a coarser, aggregated tile-level. This strategy, motivated by ideas from Multiple Instance Regression (MIR) [8], serves multiple purposes. First, it helps mitigate the effect of local [9], since aggregated values effectively smooth over inconsistencies caused by building misalignment or identical cost artifacts. Second, it provides the model with flexibility to shift focus away from noisy GEM inputs and learn from auxiliary patterns present in population and light intensity data. Third, it supports a clear benchmark: a model that directly uses the GEM raster (without ML) would exhibit full dependence on the GEM channel, whereas a successful model under our framework will demonstrate reduced GEM sensitivity by intelligently compensating with information extracted from auxiliary covariates.

Ultimately, our goal is to produce asset exposure estimates that are not only more stable but also explainable through learned uncertainty and feature attribution.

1.3 Technical Approach Overview

All inputs are rasterized onto a 30-arcsec global grid and assumed mutually independent. For each input instance, or grid cell, a surrounded 5×5 spatial patch of the three input channels is extracted and tokenized, forming 25 tokens each with three-dimensional features (Lit, Pop, GEM). These input tokens are fed into an attention-based neural architecture inspired by vision transformers, followed by a multi-layer perceptron (MLP) [10] head that produces:

- **Predicted asset value**

The predicted re-estimated asset value in each 30 arcsec grids

- **Predicted uncertainty (variance)**

The predicted variance in each 30 arcsec grids, enabling a probabilistic interpretation for asset value prediction.

To train the model, we use a likelihood-based loss function that operates at the tile level by summing predicted means and variances across grid cells. This formulation allows us to accommodate local noise while capturing disagreement across data sources [11]. To prevent the model from exploiting this flexibility to produce overly smooth or unrealistic fine-grid patterns, we introduce an alpha-regularisation term. This term penalizes grid-level predictions that deviate too far from the original GEM values, using a soft threshold based on a user-defined trust parameter α . This ensures that GEM values still serve as local anchors, especially in high-quality regions, while allowing for some reallocation based on stronger auxiliary signals. To be specific, bigger α yields wider bound, allowing reallocation with higher extend of freedom.

In parallel, we introduce a GEM sensitivity penalty as a separate and complementary regularisation term. This penalty explicitly discourages the model from becoming overly sensitive to the GEM input channel by penalizing the output Jacobian with respect to GEM input [12]. This has the effect of making the model’s predictions more stable in the presence of local GEM noise or estimation errors. Unlike alpha regularisation, which encourages local fidelity, the sensitivity term pushes the model to defer to auxiliary signals (Lit and Pop) when they provide stronger or more consistent information.

Together, these components support a trust-weighted learning strategy: GEM inputs are respected where they appear reliable, but not blindly followed when they are noisy.

1.4 Summary of Findings

In this work, we propose a novel spatial machine learning framework that re-estimates asset values at fine resolution while explicitly modelling uncertainty and mitigating over-reliance on potentially noisy inputs. By leveraging localized patch-based context and auxiliary geospatial covariates, the model learns to produce both predictions and uncertainty estimates that remain robust to inconsistencies in the original GEM asset data. Through a combination of aggregated supervision, attention-based , and targeted regularisation, the method not only improves predictive stability under perturbations but also reveals interpretable spatial sensitivity patterns. These findings suggest that our approach can serve as a flexible and uncertainty-aware building block for broader geospatial applications, such as collaboration with attempts of high-resolution population disaggregation [13] [14], in which ground truth is almost impossibly accessible, and the results are expected to be imperfect.

2 Literature Review

2.1 Overview of the GEM Exposure Data and Its Role in Global Risk Assessment

The Global Earthquake Model (GEM) Foundation has developed one of the most widely used global exposure datasets for seismic risk modelling [15]. GEM also provides spatially detailed estimates of physical asset values (replacement costs) for buildings across the world, which serves as a foundational input for probabilistic seismic risk assessment models. Such model is powered by GEM's own OpenQuake engine [4], and is openly accessible for use by researchers, governments, and humanitarian organizations.

At its core, GEM's exposure model adopts a bottom-up methodology, which aims to aggregate information from the most detailed level available [3]. In contrast to top-down methods [16] [17] [18], which disaggregate national economic totals using proxies like population or nighttime lights for example the Lit-Pop method for regional GDP disaggregation [19], the bottom-up approach begins with local data sources: building inventories, housing censuses, socioeconomic indicators, and where necessary, proxies like gridded population or built-up area. These data are used to estimate **the number of buildings** in each administrative unit or grid cell, their **average floor area**, and an appropriate **replacement cost per square metre**. The total asset value for a given unit is then computed as the product of these three factors.

The final exposure dataset provides values such as: number of buildings per typology, average built-up area per building, urban/rural classification and most importantly replacement cost per square metre.

GEM disaggregates building exposure into a dense set of spatial units, where each unit represents a group of buildings with similar characteristics. Each unit is tagged with aggregated building counts, floor area, and replacement cost, and serves as a representative exposure estimate for that local cluster

The GEM exposure model offers several key advantages. First, it provides consistent global coverage, enabling comparative risk analysis across countries and continents. Second, its bottom-up approach allows for locally tailored exposure estimation where sufficient data exist, providing a more realistic picture of built assets than uniform or top-down global models. A global exposure (v2023) is demonstrated in Figure 2 [3], note that only data at such a coarse resolution (1000 km² hexagon) are freely accessible.

Despite its many strengths, the GEM exposure model also presents challenges when used at fine spatial resolutions. One key limitation lies in the nature of its input data. While the bottom-up strategy is sound in theory, in practice, GEM often relies on indirect proxies, such as population density, built-up area from satellite imagery, or gridded nightlight intensity, to estimate building counts or distributions [20]. These proxies can be highly variable and are prone to local inconsistencies, especially in rapidly developing or informal urban areas.

Moreover, some attributes like average floor area per building or replacement cost per square meter are typically assigned based on national or regional averages. These estimates often ignore within-city variation in wealth, land use, and construction quality. For example, a hexagon covering a mix of informal settlements and modern high-rise developments may receive a uniform building type and cost assumption, thereby misrepresenting the actual asset distribution.

Another source of local error stems from the treatment of building height. GEM does not explicitly model height as a separate multiplier, but rather includes it implicitly in the average floor area per building. As a result, any misrepresentation of the number of stories will distort total replacement cost estimates, especially in high-density urban areas where vertical development is common. These localized inaccuracies can compound at fine spatial scales, making GEM less reliable for high-resolution applications such as intra-urban vulnerability mapping

or grid-level exposure prediction.

This research builds on GEM's global exposure model by investigating and addressing the limitations of using it at fine spatial scales. While GEM offers a valuable global baseline, its exposure estimates are not optimized for local-level accuracy, especially in data-scarce or structurally heterogeneous environments. The goal of this work is to assess, and where possible, reduce the spatial noise and uncertainty that arise from the use of coarse assumptions in fine-gridded models.

Specifically, this project proposes to integrate GEM's exposure data with high-resolution covariates—such as nightlight intensity and gridded population density—to train machine learning models that can produce more spatially detailed and locally adaptive asset value estimates. By incorporating probabilistic uncertainty and testing performance at multiple scales, the research aims to both refine exposure modelling and better understand where GEM's bottom-up methodology holds and where it breaks down.

In doing so, the project contributes to ongoing efforts to improve the spatial granularity and reliability of global exposure datasets, particularly for use in climate and disaster risk modelling.

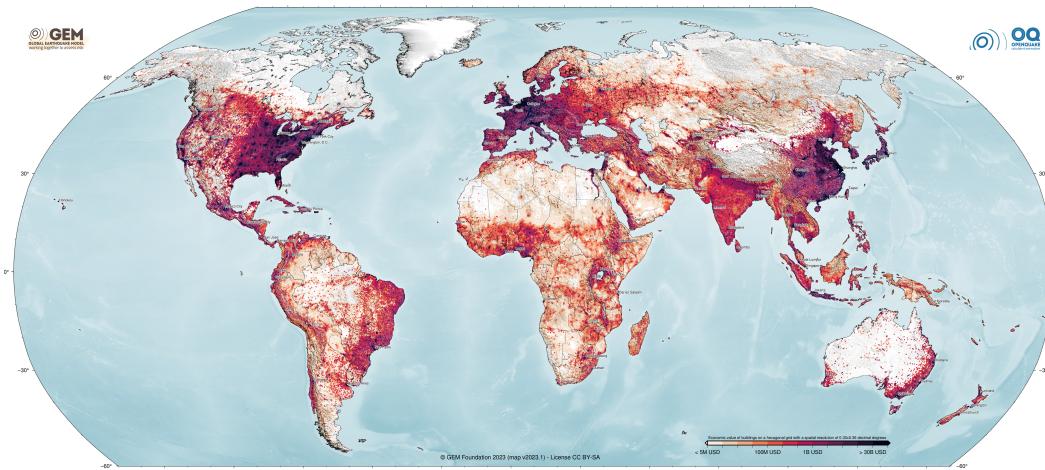


Figure 2: The Global Exposure Map (v2023.1) presents the geographic distribution of residential, commercial and industrial buildings. The total replacement cost is presented on a hexagonal grid, with a spacing of 0.30×0.36 decimal degrees (approximately $1,000 \text{ km}^2$ at the equator).

2.2 Aggregated Supervision via Multiple Instance Learning (MIR)

Multiple Instance Learning (MIR) is a weakly supervised learning framework in which supervision is provided at the group (or bag) level rather than for individual instances. Instead of requiring a label for each data point, MIR associates a label with a set of instances, allowing the model to learn instance-level predictions from aggregated or uncertain ground-truth information [21]. This paradigm is particularly suited for problems in geospatial and remote sensing domains, where high-resolution labels may be noisy, imprecise, or unavailable, but coarser aggregate statistics are more reliable.

MIR has been applied successfully in tasks such as land use classification, crop yield estimation, and poverty mapping, where learning from aggregated supervision leads to improved generalizability and robustness. In these settings, the model generates predictions at the fine resolution but evaluates performance based on an aggregated function—such as the sum, mean, or max—over each bag, matched against the corresponding coarse label.

A prominent example is MIR applied to county-level crop yield prediction from remote sensing data. In this scenario, each county constitutes a bag, labelled with overall yield, while satellite imagery pixels within the county serve as instances. Wagstaff et al. (2008) highlighted this: “A compelling and large-scale problem that requires multiple-instance regression is crop yield prediction from remote sensing data” [22]. Building on this, recent work by Wang et al. [23] employs MIR with an attention-based mechanism to address mixed-pixel noise—achieving high accuracy in U.S. Corn Belt yield prediction ($R^2 \approx 0.84$).

In this work, we adopt an aggregated supervision strategy based on the principles of MIR to mitigate the impact of local noise and positional uncertainty in the GEM asset dataset. We define each 5×5 tile (25 fine-resolution grids inside) as a *bag*. Within a tile, each grid cell serves as an instance, for which we extract a $3 \times 5 \times 5$ local patch (three input channels over a 5×5 spatial neighbourhood) as the model input. The model produces a grid-level prediction \hat{y}_p for each grid, or its corresponding patch p , in the tile. Rather than supervising these grid-level predictions individually, since the GEM values at this resolution are often noisy or spatially imprecise, we supervise only their **sum** at the tile level. The tile-level target y_{tile} is computed by summing the GEM asset values at the corresponding grid (corresponding patch centres) within the tile. To formalize the training data:

$$\mathcal{D}_{\text{train}} = \left\{ \{\mathbf{X}_{\text{patch}}\}_{\text{patch} \in \text{tile}}, y_{\text{tile}} \right\}_{\text{tile} = 1, 2, \dots, N}, \text{ where: } y_{\text{tile}} = \sum_{\text{patch} \in \text{tile}} X_{G, \text{centre}}^{(\text{patch})}$$

The model’s tile-level prediction is similarly the sum of its per-grid predictions:

$$\hat{y}_{\text{tile}} = \sum_{\text{patch} \in \text{tile}} \hat{y}_{\text{patch}}. \quad (1)$$

The MIR-based loss is then defined as:

$$\mathcal{L}_{\text{MIR}} = \mathcal{L}(\hat{y}_{\text{tile}}, y_{\text{tile}}). \quad (2)$$

This formulation preserves fine-resolution feature extraction at the grid level while reducing the influence of noisy GEM estimates at that scale. By aggregating predictions and targets over the 5×5 tile, the model receives a more stable and spatially robust supervision signal, while still learning grid-level representations that are consistent when summed to the coarser scale.

3 Method

This section outlines the methodological framework, including the model architecture, training targets, and treatment of uncertainty. Our approach predicts asset values on each grid using a 75-dimensional input derived from the surrounding extracted 3-channel 5×5 patches (night lights, population and GEM exposure). Since GEM values are also used as part of the input, we account for their uncertainty by modelling prediction variance as stemming both from data noise and input unreliability. A custom likelihood-based loss is introduced to capture this causal link between uncertain inputs and predictive uncertainty.

3.1 Data collection, enhancement and transformation for skewed data distribution

3.1.1 Gridding, extracting patches, vectorisation

To enable learning at high spatial resolution (30 arcsec, approximately 1 km^2), all input data sources are first converted into a common raster grid format. Each asset representative point in the GEM dataset, originally defined by geographic coordinates (longitude, latitude), is indexed and aligned to a 2D array, where each element corresponds to a 30 arcsec grid cell. Asset values are accumulated per grid cell to generate a finely gridded exposure map.

To reduce label noise and reflect the regional scale of validation metrics, we further aggregate this fine-resolution grid into coarser tiles in which each comprises 5×5 grid cells (i.e., 150 arcsec tile-level raster map). This results in two resolutions of the GEM asset map: a fine 30 arcsec grid used as input, and a coarse 150 arcsec tile map used to supervise predictions.

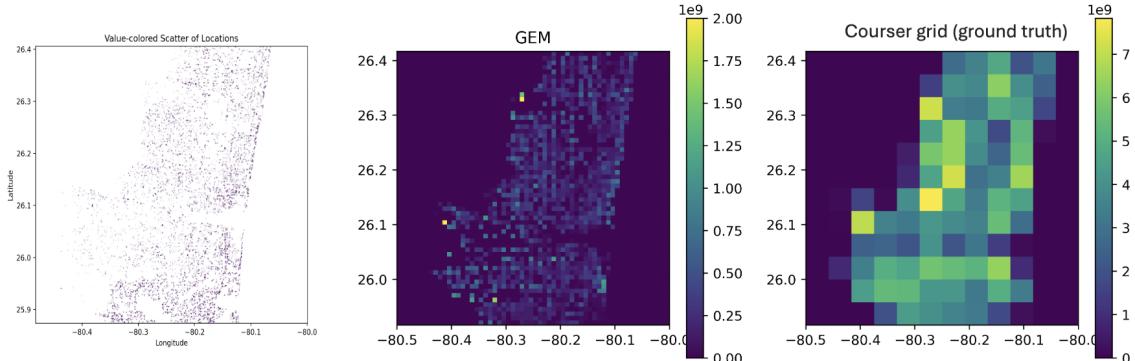


Figure 3: The leftmost image shows the original residential GEM representative points tagged with asset values. These are rasterized to create the fine-resolution grid. The fine grid is then subdivided into 5×5 tiles, with values aggregated within each tile to form the training targets.

Next, we prepare the model input by stacking auxiliary covariates: night-time light intensity and population count onto the fine-resolution GEM grid. This results in a three-channel input tensor where each channel corresponds to one data source (GEM, Lit, Pop). For each of the 25 grid cells within a coarse tile, we extract a 5×5 spatial patch centred on that grid cell across all three channels. This patch serves as the local context for predicting asset value on the interested grid.

After this extraction, the data is vectorised in a 5-dimensional tensor of shape:

$$\mathbf{X}_{\text{input}} \in \mathbb{R}^{N \times 25 \times 3 \times 5 \times 5}$$

where:

- N is the number of coarse tiles

- 25 is the number of fine grid cells per tile
- 3 corresponds to the input channels (GEM, Lit, Pop)
- 5x5 is the spatial patch size centred on each grid cell

3.1.2 Patch random rotation

To enhance generalisation and prevent the model from learning spurious directional biases, we apply rotational data augmentation during training [24]. Specifically, each extracted 5×5 patch is randomly rotated 0° , 90° , 180° , or 270° with equal probability. This is based on the assumption that there is no inherent directional dependence in the spatial distribution of assets relative to the centre of each patch.

Without such augmentation, the model might overfit to regional layout patterns present in the training data. For instance, as shown in 4, the Florida dataset, used for model training, has a concentration of high-value assets along the south-eastern coastline. This could lead the model to learn an unintended heuristic such as: “if the east side of the patch has low or missing values, the centre is likely to have high asset value”. Such a pattern would fail to generalize to other regions, like the west coast, where the spatial context is horizontally mirrored.

Rotational augmentation mitigates this issue by enforcing spatial equivalence in the model’s learned features, helping it focus on relative patterns rather than fixed orientations.

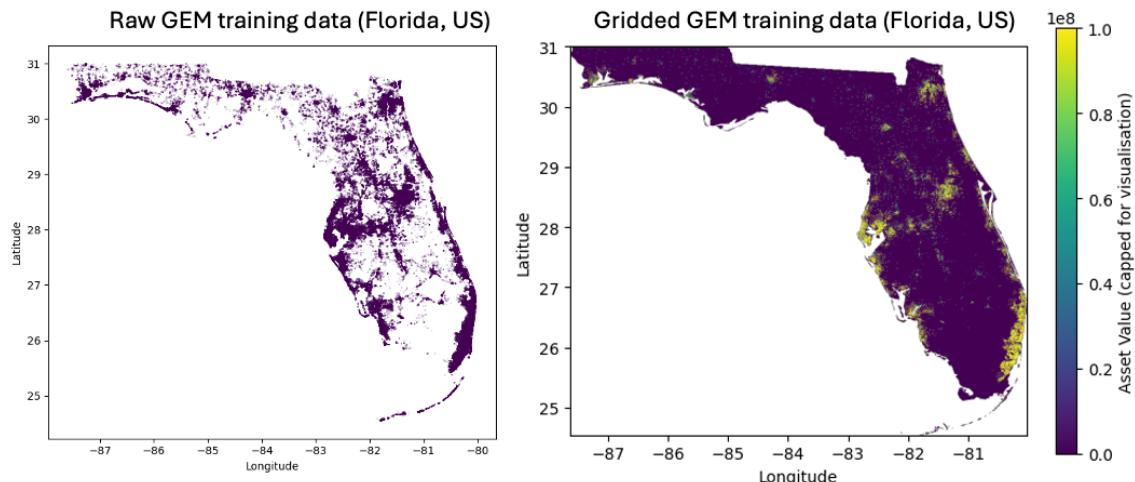


Figure 4: Asset exposure data for Florida used in training. The left panel shows the original GEM data, where each dot represents a representative point tagged with an asset value. The right panel shows the same data after rasterization to a 30 arcsecond grid, where asset values have been spatially aggregated into grid cells. The concentration of high-value regions along the east coast highlights the importance of rotational data augmentation to prevent the model from learning geographically specific directional biases.

3.1.3 Right-skewed distribution, logarithm transform

Since the input covariates, Lit, Pop and GEM, are all heavily right-skewed, we apply a logarithmic transformation [25] to each channel prior to training:

$$X \rightarrow \log(X + 1)$$

This transformation helps compress the extreme upper values while expanding the lower range, resulting in a more balanced and wider distribution. Such normalization is crucial for stable and efficient gradient-based optimization in deep learning [26], as it ensures better numerical conditioning and helps the model learn effectively across all value ranges. Figure 5 below

illustrates the empirical distributions of all three channels before and after applying the log transform. Note that the addition of 1 ensures numerical stability and preserves positivity for all inputs.

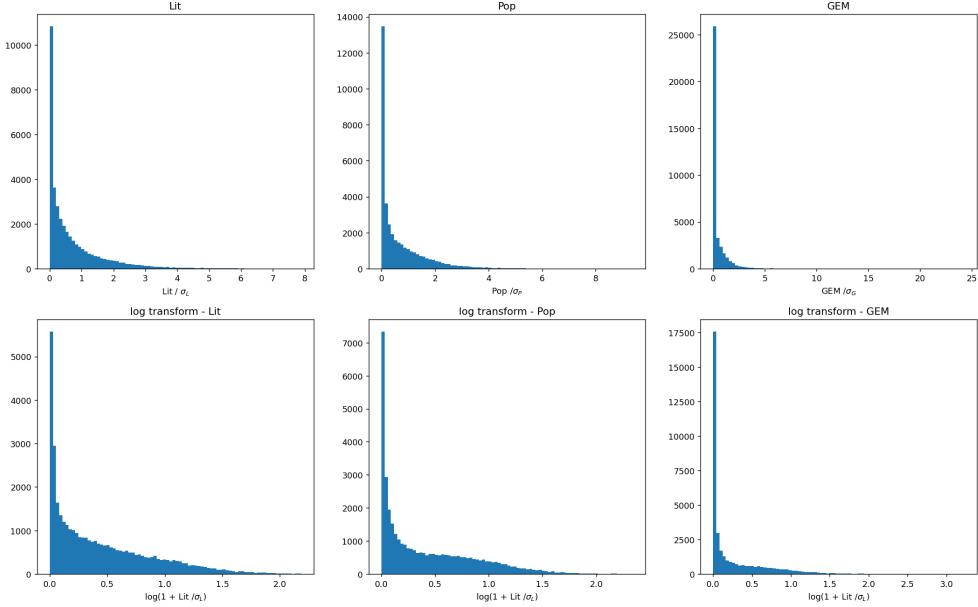


Figure 5: Distribution of raw (top row) and log-transformed (bottom row) input features for the residential Florida training set. Columns represent night-time light intensity (left), population density (middle), and GEM asset value (right). The $\log(x + 1)$ transform reduces skewness and expands lower-value ranges, improving input conditioning for training. Note that all values were initially scaled down by its standard deviation for better numerical stability.

3.2 Deep learning model, attention mechanism

3.2.1 Model structure

To model regional asset exposure while capturing spatial context and uncertainty, we adopt a transformer-inspired deep learning architecture with two outputs: the predicted asset value and its associated variance. The design is motivated by the vision transformer framework proposed in *"An Image is Worth 16×16 Words"* [27], where spatial tokens are treated as sequential inputs to an attention-based model. In our case, each 5×5 patch around a target grid is tokenized into 25 vectors, each corresponding to a spatial location within the patch. This representation allows the model to learn dependencies and patterns from the entire patch.

As Figure 6 shown, the three input channels (log-transformed Lit, Pop, and GEM) for each 5×5 patch are flattened into 25 vectors of length 3. These are projected to a higher-dimensional embedding space via a learnable linear transformation. Positional encodings are added to retain the spatial structure of the patch, ensuring the model can distinguish the position of each token despite permutation invariance in attention. This embedded sequence is passed through a residual self-attention block, where the attention mechanism enables each spatial token to “communicate” with every other, modelling co-occurrence and collaborative patterns across the patch [28].

The resulting attention-enhanced sequence is flattened and passed through a multi-layer perceptron (MLP) head with ReLU activations and dropout for regularisation. The final layer outputs two values per input tile: a predicted asset value \hat{y} , and a predicted variance $\hat{\sigma}^2$, both constrained to be positive via a SoftPlus activation. This setup allows the model not only to fit the central value but also to express uncertainty arising from both data noise and unreliable

input covariates [29] (e.g., GEM values being both input and noisy target), for which we will interpret the prediction together as a Gaussian random variable:

$$\hat{y}_{\text{grid}} \sim \mathcal{N}(\mu(\mathbf{X}_{\text{patch}}), \sigma^2(\mathbf{X}_{\text{patch}}))$$

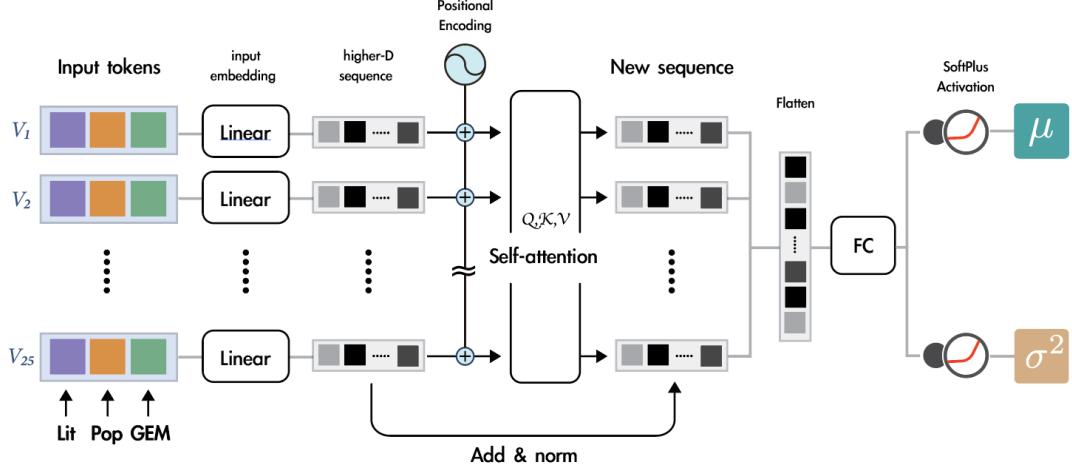


Figure 6: Overview of the ML model architecture.

3.3 Uncertainties

3.3.1 Non-deterministic Input Feature and Taylor Expansion Estimate

We model predictive aleatoric uncertainty by assuming the network outputs both a mean and variance estimate for each $3 \times 5 \times 5$ input patch:

$$\mathbf{X}_{\text{patch}} = [X_L, X_P, X_G] \in \mathbb{R}^{3 \times 5 \times 5},$$

Again, X_L , X_P , and X_G are the log-transformed Lit, Pop and GEM, respectively. Each is treated as a 5×5 spatial patch. The model returns two outputs:

$$\mu_{\theta}(\mathbf{X}_{\text{patch}}), \quad \sigma_{\theta}^2(\mathbf{X}_{\text{patch}}),$$

where $\mu_{\theta}(\mathbf{X}_{\text{patch}})$ is the re-estimated asset value at the centre of the patch with $\sigma_{\theta}^2(\mathbf{X}_{\text{patch}})$ represents the learned aleatoric variance. These are produced by a neural network parametrised by θ .

However, since the GEM asset value is not only part of the prediction target but also used as an input feature, we must consider the fact that the GEM input itself is stochastic. This uncertainty can propagate through the model and influence the predicted mean. To reflect this, we introduce an additional uncertainty component due to the non-deterministic nature of the GEM input.

Thus, the final model prediction is given by:

$$\hat{y}_{\text{grid}} = \mu_{\theta}(\tilde{\mathbf{X}}_{\text{patch}}) + \xi,$$

where:

- \hat{y}_{grid} is the probabilistic re-estimated asset value at the patch's centre grid;

- $\xi \sim \mathcal{N}(0, \sigma_\theta^2(\mathbf{X}_{\text{patch}}))$ represents model-predicted aleatoric uncertainty;
- $\tilde{\mathbf{X}}_{\text{patch}}$ denotes the stochastic input patch, where the GEM channel is modelled as a random variable. The other two channels, X_L and X_P , are treated as deterministic.

Since asset values are used both as input and as part of the supervision target, we approximate the uncertainty in the GEM input using the model's own predicted variance. In addition, we ignore the uncertainty of the uncertainty itself—i.e., we assume ξ is drawn from a Gaussian with fixed variance evaluated at the deterministic original per-grid input $\mathbf{X}_{\text{patch}}$.

To estimate how input uncertainty in the GEM channel affects the prediction, we perform a first-order Taylor expansion of μ_θ around the deterministic input:

$$\mu_\theta(\tilde{\mathbf{X}}_{\text{patch}}) \approx \mu_\theta(\mathbf{X}_{\text{patch}}) + (\tilde{X}_G - X_G)^\top (\nabla_{X_G} \mu_\theta(\mathbf{X}_{\text{patch}})) = \mu_\theta(\mathbf{X}_{\text{patch}}) + \varepsilon^\top J_\mu,$$

where $J_\mu = \nabla_{X_G} \mu_\theta(\mathbf{X}_{\text{patch}})$ denotes the gradient (flattened Jacobian in this context) of the predicted mean with respect to the flattened GEM patch input.

We treat the GEM patch X_G as a 25-dimensional vector, flattened row-wise:

$$X_G = [X_{G_1}, X_{G_2}, \dots, X_{G,\text{cen}}, \dots, X_{G_{25}}]^\top \in \mathbb{R}^{25}.$$

The perturbation $\varepsilon \in \mathbb{R}^{25}$ is drawn from a zero-mean multivariate Gaussian:

$$\varepsilon \sim \mathcal{N}(0, \Sigma_G),$$

where the covariance matrix $\Sigma_G \in \mathbb{R}^{25 \times 25}$ models input uncertainty in the GEM channel. Given that the empirical average absolute correlation (i.e., the off-diagonal entries of the covariance matrix) is only around 0.1, I will treat the features as approximately independent. This simplification reduces computational cost, and it is acceptable in this context since the final objective relies only on a Taylor approximation rather than an exact treatment of the full dependence structure. We further assume that the uncertainty (variance) is proportional to its squared GEM value, normalised by the centre value and the uncertainty of which is approximately the predicted variance:

$$\Sigma_G = \frac{\sigma_\theta^2(\mathbf{X}_{\text{patch}})}{X_{G,\text{cen}}^2} \begin{bmatrix} X_{G_1}^2 & & & \\ & X_{G_2}^2 & & \\ & & \ddots & \\ & & & X_{G_{25}}^2 \end{bmatrix} = \frac{\sigma_\theta^2(\mathbf{X}_{\text{patch}})}{X_{G,\text{cen}}^2} \text{Diag}(X_{G_1}^2, \dots, X_{G_{25}}^2).$$

The variance contribution from this stochastic input shift, $\varepsilon^\top J_\mu$, is given by:

$$\begin{aligned} \text{Var}(\varepsilon^\top J_\mu) &= \mathbb{E}[(\varepsilon^\top J_\mu)^2] \\ &= \mathbb{E}[J_\mu^\top \varepsilon \varepsilon^\top J_\mu] \\ &= J_\mu^\top \mathbb{E}[\varepsilon \varepsilon^\top] J_\mu \\ &= J_\mu^\top \Sigma_G J_\mu \\ &= \frac{\sigma_\theta^2(\mathbf{X}_{\text{patch}})}{X_{G,\text{cen}}^2} J_\mu^\top \text{Diag}(X_{G_1}^2, \dots, X_{G_{25}}^2) J_\mu \\ &= \frac{\sigma_\theta^2(\mathbf{X}_{\text{patch}})}{X_{G,\text{cen}}^2} \sum_{k=1}^{25} X_{G_k}^2 \left(\frac{\partial \mu_\theta(\mathbf{X}_{\text{patch}})}{\partial X_{G_k}} \right)^2. \end{aligned}$$

Finally, treating ξ and $\varepsilon^\top J_\mu$ as independent Gaussian noise terms, the overall predictive distribution becomes:

$$\hat{y}_{\text{grid}} \sim \mathcal{N} \left(\mu_\theta(\mathbf{X}_{\text{patch}}), \sigma_\theta^2(\mathbf{X}_{\text{patch}}) + \frac{\sigma_\theta^2(\mathbf{X}_{\text{patch}})}{X_{G,\text{centre}}^2 + \text{eps}} \sum_{k=1}^{25} X_{G_k}^2 \left(\frac{\partial \mu_\theta(\mathbf{X}_{\text{patch}})}{\partial X_{G_k}} \right)^2 \right)$$

where a small constant eps is added in the denominator for numerical stability.

3.3.2 Out-of-distribution (OOD) detector

While our model explicitly captures **aleatoric uncertainty** through a learned predictive variance, **epistemic uncertainty**, which arises from lack of knowledge about certain inputs, remains inaccessible during inference. To partially recover this missing component, we adopt a **one-class Support Vector Machine (SVM)** as a post hoc detector to identify whether a given test-time input falls within the training distribution [30].

The one-class SVM constructs a hyper-surface that encloses the majority of training data in a high-dimensional space. Any new input falling outside this region is flagged as *out-of-distribution* (OOD), implying it may lead to unreliable predictions due to insufficient representation during training. Note that we train the OOD detector before principle component analysis method (PCA) compressing our input data into a lower dimensional sub-eigenspace (with explained ratio = 0.95) to prevent overfitting.

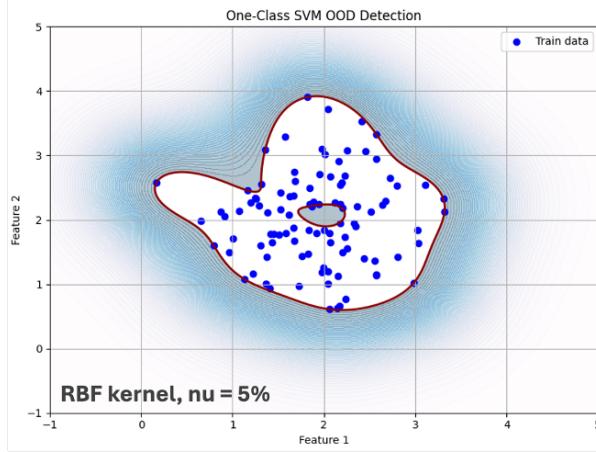


Figure 7: Toy example of one-class SVM for OOD detection. Blue dots represent training data in 2D feature space. Contour lines of decision score are presented in blue, in which the red one shows the decision boundary learned by a one-class SVM with an RBF kernel. A non-zero $\nu = 5\%$ allows the model to ignore a small fraction of outliers, resulting in a tighter and more realistic boundary. At test time, points outside this region are flagged as out-of-distribution and treated as epistemically uncertain.

3.4 Loss function

To supervise the model effectively while accounting for uncertainty and input noise, we design a composite loss function consisting of three components, each targeting a specific modelling goal.

3.4.1 Tile-level likelihood

We treat the aggregated prediction over each tile as a Gaussian random variable, where the predicted tile-level mean is the sum of grid-level predictions, and the tile-level variance is the sum of predicted variances from each grid cell (assuming independence). This probabilistic formulation encourages the model to learn both accurate predictions and well-calibrated uncertainty. Therefore we have:

$$\begin{aligned} \hat{y}_{\text{tile}} | \mathbf{X}_{\text{tile}}, \theta &\sim \mathcal{N}(\mu_{\text{tile}}, \sigma_{\text{tile}}) \\ &= \mathcal{N} \left(\sum_{\text{patch} \in \text{tile}} \mu_{\theta}(\mathbf{X}_{\text{patch}}), \sum_{\text{patch} \in \text{tile}} \left[\frac{\sigma_{\theta}^2(\mathbf{X}_{\text{patch}})}{X_{G,\text{centre}}^{(\text{patch})2} + \text{eps}} \sum_{k=1}^{25} X_{G,k}^{(\text{patch})2} \left(\frac{\partial \mu_{\theta}(\mathbf{X}_{\text{patch}})}{\partial X_{G,k}} \right)^2 \right] \right) \end{aligned}$$

Its negative logarithm, or negative log-likelihood (NLL) will then be the primary part of loss function:

$$\mathcal{L}_{\text{NLL}} = \frac{1}{2} \sum_{\text{tile}} \left[\log(\sigma_{\text{tile}}^2) + \frac{(\mu_{\text{tile}} - y_{\text{tile}})^2}{\sigma_{\text{tile}}^2} \right] \quad (3)$$

where the tile-level target:

$$y_{\text{tile}} = \sum_{\text{patch} \in \text{tile}} X_{G,\text{centre}}^{(\text{patch})}$$

3.4.2 Grid-level regularisation

To avoid degenerate behaviour in uncertainty estimation—such as arbitrarily large predicted variances that trivially minimize the loss—we introduce a soft regularisation term on the grid-level predictions. This term penalizes excessive deviation between the model’s predicted mean $\mu_\theta(X)$ and the original GEM value at the centre of each patch, denoted $X_{G,\text{centre}}^G$. Formally, this regularisation is scaled by a hyperparameter $\alpha > 0$ and takes the form:

$$\mathcal{L}_{\text{grid-reg}} = \sum_{\text{patch}} \max \left\{ 0, \left| \mu_\theta(\mathbf{X}_{\text{patch}}) - X_{G,\text{centre}}^{(\text{patch})} \right| - \alpha X_{G,\text{centre}}^{(\text{patch})} \right\}^2$$

Which essentially forms a soft bound on grid-level prediction:

$$\mu_\theta(\mathbf{X}_{\text{patch}}) \in_{\text{soft}} \left[(1 - \alpha) X_{G,\text{centre}}^{(\text{patch})}, (1 + \alpha) X_{G,\text{centre}}^{(\text{patch})} \right]$$

In this way, the tolerance for deviating from GEM reflects how much the model ”trusts” the original data. A larger α allows more flexibility to override GEM, while a smaller α anchors predictions closer to GEM values. The degree of this deviation, especially when paired with predicted uncertainty, can serve as an indicator of the model’s confidence in GEM versus auxiliary covariates (Lit, Pop).

3.4.3 GEM sensitivity penalty

To promote robustness in re-estimating asset values, we introduce a sensitivity regularisation term that penalizes the model’s dependence on the GEM input channel. Since GEM values may be noisy or uncertain, we want the model’s predictions to remain stable under small perturbations to this input. Minimizing the gradient of the output with respect to GEM channel reduces this sensitivity and encourages the model to rely more on auxiliary signals when appropriate [12].

In addition, because GEM is used both as input and target, this regularisation also mitigates data leakage by discouraging the model from trivially copying the GEM values. To preserve the interpretation of the α -band as a belief margin, we apply this penalty only when the prediction falls inside the α -band. In that case, agreement with GEM is allowed, but only if supported by auxiliary features; otherwise, excessive GEM sensitivity is penalized inside the α -band. When the prediction deviates from GEM beyond the α -band, the sensitivity penalty is suppressed to avoid conflicting with the model’s incentive to correct potentially unreliable inputs.

Mathematically, such sensitivity loss takes the form:

$$\text{First we define a mask: } \mathcal{M}_\alpha = \left\{ \text{patch} : \left| \mu_\theta(\mathbf{X}_{\text{patch}}) - X_{G,\text{centre}}^{(\text{patch})} \right| < \alpha X_{G,\text{centre}}^{(\text{patch})} \right\}$$

The mask \mathcal{M} prediction falls inside feasible region:

$$\text{Sensitivity punishment: } \mathcal{L}_{\text{sen-reg}} = \sum_{\text{patch} \in \mathcal{M}_\alpha} \left\| \nabla_{x_G} \mu_\theta(\mathbf{X}_{\text{patch}}) \right\|_2^2$$

In essence, this term encourages the model to exploit auxiliary information (Lit , Pop), but only when the prediction already aligns with GEM within the belief-consistent threshold. This allows the model to remain faithful to GEM where appropriate, while learning to down-weight it where other features provide stronger evidence.

4 Experiments and Results

4.1 Overview of Input Data

We use three geospatial datasets as inputs to our asset value prediction model: night-time light intensity (Lit), population count (Pop), and asset exposure data (GEM). All datasets are used in their most recent available versions and are spatially aligned at a common resolution of 30 arcsec.

The GEM exposure data is available in three distinct asset categories: residential, commercial, and industrial. While it is possible to train separate models or design multi-output architectures to accommodate different asset types, this study focuses primarily on residential exposure values. Residential assets represent the dominant share of overall asset value in most regions, and the other two categories often suffer from limited spatial coverage and data sparsity, making them less suitable for supervised learning at fine spatial resolutions.

4.1.1 Night-time Light Data (Lit)

Nighttime light intensity is obtained from NASA's **VNP46A4 Yearly Moonlight-Adjusted Nighttime Lights product (2023)** [31]. This dataset provides yearly composites of light emissions observed at night, adjusted for lunar illumination effects [32] [33]. The data is packaged as global GeoTIFFs with multiple data channels.

Radiance values in this product are reported in $nWcm^{-2}sr^{-1}$. In addition, to align it with the other inputs, we downsample the nightlight layer from 15 arcsec to 30 arcsec using bilinear interpolation.

4.1.2 Population Data (Pop)

The population data comes from NASA's **Gridded Population of the World, Version 4 (GPWv4): Population Count, Revision 11**, specifically at a resolution of 30 arcsec. GPWv4 estimates are based directly on national census data, collected from the finest administrative units available in each country. This results in a census-driven and spatially disaggregated representation of population distribution, suitable for integration with other Earth observation datasets [34].

4.1.3 Asset Exposure Data (GEM)

The asset exposure data is obtained from the Global Earthquake Model (GEM) Foundation [3]. GEM provides a detailed spatial breakdown of asset values based on a bottom-up assessment methodology (see Literature Review for more details). The GEM dataset is manually structured in 30 arcsec grid cells, making it directly compatible with our chosen resolution. Note that while GEM's aggregated exposure data is freely available at the first administrative (Admin 1) level, the dense high-resolution dataset used in this study is provided exclusively through academic collaboration with GEM and is not publicly accessible.

4.1.4 Training Region and Input Channels

We train our model on data from Florida, a region frequently impacted by tropical cyclones and other natural disasters. This choice is motivated by two key factors:

- **Relevance to Disaster Risk:** As a cyclone-prone area, Florida provides a realistic setting for assessing exposure prediction in high-risk coastal zones.
- **Data Quality Considerations:** Being a high-income region, Florida likely benefits from higher-quality and more granular statistical reporting, which directly supports the

bottom-up methodology used in the GEM exposure model. As a result, we consider GEM data over Florida relatively trustworthy and suitable for supervised training.

Three raster layers form the basis of the input to our model, which is vectorised into a structured tensor of size $5181 \times 25 \times 3 \times 5 \times 5$, corresponding to 5181 tiles, each comprising 25 patches, each containing three channels (Lit, Pop, GEM) over a 5×5 grid centred on the prediction tile.

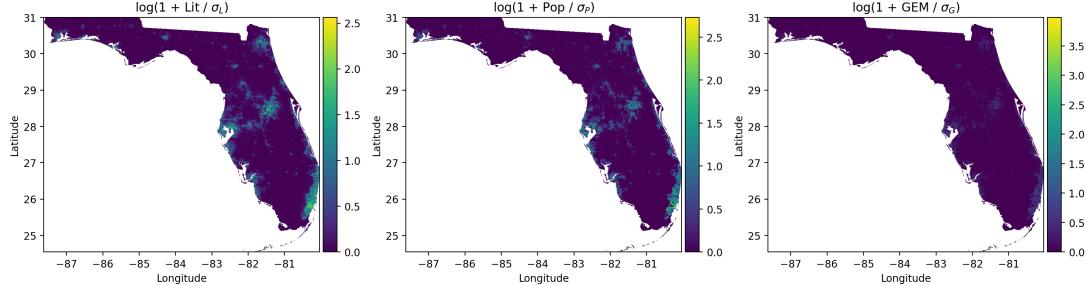


Figure 8: Log of scaled-Input channels over Florida (from left to right: Nighttime Lights (Lit), Population count (Pop) and GEM Residential Asset Values), all projected to 30 arcsec resolution testing.

To understand the relationship between auxiliary inputs and target values, we compute grid-wise Pearson correlations between the input layers and GEM exposures, as shown in Figure 9. The residential asset values show a weak positive correlation with population density and a weak or slightly negative correlation with nighttime light intensity. For industrial assets, the trend appears to be reversed, with light intensity showing a slightly positive correlation and population exhibiting a weak or negative one. In both cases, the correlation values are small and scattered, indicating a noisy relationship. These patterns suggest that neither auxiliary input offers strong linear predictive power individually, reinforcing the need for a learning-based model to capture more complex associations.

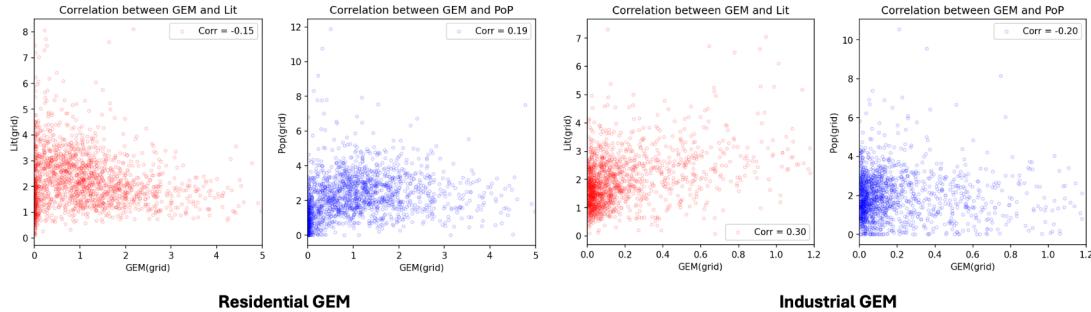


Figure 9: Scatter plots and correlation coefficients between: Lit vs. Residential GEM (left panel, blue dots) Pop vs. Residential GEM (left panel, red dots) Lit vs. Industrial GEM (right panel, blue dots) Pop vs. Industrial GEM (right panel, red dots). Note that all values has been scaled down by 1 standard deviation of respective channel.

4.2 Model Training Setup

This section provides an overview of the training configuration . All theoretical aspects of the model architecture and training objective have been discussed in the Methods section.

4.2.1 Training loop skeleton code

With the ML setup of PyTorch, the data initialisation and training loop follows the below skeleton code.

Algorithm 1 Model Training loop

- 1: **Input:** Training data X_{train} (log-scaled tensor), model f_θ , batch size B , learning rate η , regularisation weights λ_{reg} , λ_{sen} , tolerance α , number of epochs T
- 2: Initialize model parameters θ
- 3: Set optimizer: Adam(θ , lr = η)
- 4: **for** epoch = 1 to T **do**
- 5: Rotate patches in X_{train} randomly for augmentation
- 6: Split X_{train} into batches of size B
- 7: **for** each batch \mathcal{B} **do**
- 8: Extract GEM masks from \mathcal{B} , apply to the batched tensor (only compute non-zero GEM grids)
- 9: Compute targets y_{tile} : sum of GEM values per tile
- 10: Forward pass: $(\{\mu_p\}, \{\sigma_p^2\}) \leftarrow f_\theta(\mathcal{B})$
- 11: Compute tile level prediction: $(\mu_{\text{tile}}, \sigma_{\text{tile}}^2)$
- 12: Compute negative log-likelihood:

$$\mathcal{L}_{\text{NLL}} = \frac{1}{2} \sum_{\text{tile}} \left[\log(\sigma_{\text{tile}}^2 + \text{eps.}) + \frac{(y_{\text{tile}} - \mu_{\text{tile}})^2}{\sigma_{\text{tile}}^2 + \text{eps.}} \right]$$
- 13: Compute masked regression loss:

$$\mathcal{L}_{\text{reg}} = \lambda_{\text{reg}} \cdot \sum_p \text{Softplus} \left(\left| \mu_p - X_{G, \text{centre}}^{(p)} \right| - \alpha X_{G, \text{centre}}^{(p)} \right)^2$$
- 14: Compute per-patch Jacobians $\left\{ J_p = \left(\frac{\partial \mu_p}{\partial \mathbf{X}_p} \right) \in \mathbb{R}^{3 \times 5 \times 5} \right\}$ via `vmap(jacrev)`
- 15: Compute soft mask from:

$$\mathcal{M}_\alpha^{(p)} = 1 - \text{Sigmoid} \left(10\alpha \left| \frac{\mu_p - X_{G, \text{centre}}^{(p)}}{X_{G, \text{centre}}^{(p)}} \right| - \alpha \right) \in [0, 1] \quad \text{and apply .detach()}$$
- 16: Extract GEM channel squared gradient for all patches via `Jp[C=G, :, :, :].pow(2).sum()`
- 17: Apply soft mask to gradient regularisation term and compute sensitivity loss:

$$\mathcal{L}_{\text{sen}} = \lambda_{\text{sen}} \cdot \sum_p \mathcal{M}_\alpha^{(p)} \|\nabla_{X_G} \mu_p\|_2^2$$
- 18: Total loss: $\mathcal{L} = \mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{sen}}$
- 19: Backpropagate and update model: $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}$
- 20: **end for**
- 21: **end for**

Note that in step 12, we add a small `eps = 1e-4` in the logarithm argument as well as the denominator of the second term for numerical stability. Also, step 15-17 in above training loop, to ensure stable training behaviour, we adopt a soft mask instead of a hard cut-off (see below Figure 10) when applying sensitivity regularisation around the α -band. While the α -band defines a tolerance region based on the GEM reference value, a hard binary mask would introduce discontinuities that often lead to unstable gradient updates. The soft mask provides a

smoother transition around the band boundaries, allowing gradients to flow more naturally and avoiding sharp penalty jumps that could destabilize optimization. Note that the `.detach()` PyTorch built-in function remove its gradient calculation, which model will simply treat the mask it as a constant.

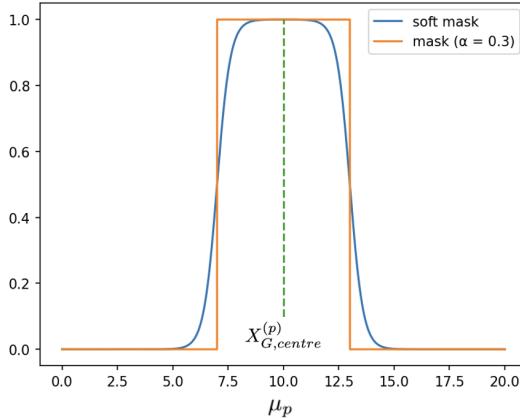


Figure 10: Soft mask (blue) versus hard binary mask (orange) for $\alpha = 0.3$. The smooth boundary of the soft mask helps ensure more stable training dynamics by avoiding abrupt gradient changes.

4.2.2 Hyperparameter Configuration

The model is trained for 80000 epochs using the Adam optimizer with a learning rate of 3×10^{-5} . The batch size is set to 700 tiles (bags), which is close to the maximum memory capacity observed for this model configuration. Larger batch sizes led to significant drops in training throughput, likely due to memory limitations when computing second-order quantities for the Jacobian-based sensitivity loss.

The learning rate is deliberately kept small to stabilize the interaction between two competing objectives: the grid-level loss, which penalizes predictions that fall outside a specified tolerance band (alpha-band) around the original GEM value, and the masked sensitivity loss, which discourages the model from overfitting to GEM input features by penalizing strong gradient dependencies. A larger learning rate was found to destabilize this dynamic — in such cases, the masked loss would aggressively push predictions away from the target band, sometimes overshooting and leading to exploding gradients. The smaller learning rate helps maintain a more stable equilibrium between pulling predictions toward reliable targets and enforcing insensitivity to noisy features.

All inputs are scaled down channel-wise using the global standard deviation over the training set. For the detail model structure, please refer to Algorithm 2

Parameter	Value
Optimizer	Adam
Learning Rate	3×10^{-5}
Batch Size	700 tiles (bags)
Epochs	80000+
Dropout rate	0.2
Loss Function	$\lambda_{\text{reg}} = 1, \lambda_{\text{sen}} = 1$.
Input Tensor Shape	$4145 \times 25 \times 3 \times 5 \times 5$ (80-20 train-test split from 5181 residential tiles)

Table 1: Training hyperparameter used in all experiments.

Algorithm 2 Forward Pass of deep learning model. The input tensor has shape $(N_p, 3, 5, 5)$, where N_p is the total number of patches from all tiles (bags) in the minibatch. Since we perform instance-level prediction, all patches are flattened across bags before being passed into the model.

- 1: Apply $\log(x + 1)$ elementwise
 - 2: Reshape $x \rightarrow \mathbb{R}^{N_p \times 25 \times 3}$ (Tokenize)
 - 3: Project input tokens: $x \leftarrow \text{Linear}(3 \rightarrow 6)(x)$ # x is now $\mathbb{R}^{N_p \times 25 \times 6}$
 - 4: Add positional encoding: $x \leftarrow x + P \in \mathbb{R}^{25 \times 6}$
 - 5: Save residual: $r \leftarrow x$
 - 6: Apply multi-head self-attention: $x \leftarrow \text{SelfAttention}(x)$
 - 7: Residual connection and normalization: $x \leftarrow \text{LayerNorm}(r + x)$
 - 8: Flatten: $x \leftarrow \text{reshape}(N_p, 150)$
 - 9: Apply MLP head: $x \rightarrow \text{FC}_{64} \rightarrow \text{ReLU} \rightarrow \text{FC}_{32} \rightarrow \text{ReLU} \rightarrow \text{FC}_{16} \rightarrow \text{ReLU} \rightarrow \text{FC}_2$
 - 10: Apply Softplus: $x \leftarrow \text{Softplus}(x)$ # Ensure positive prediction
 - 11: **Output:** Predicted mean $\hat{\mu} = x[:, 0]$, predicted variance $\hat{\sigma}^2 = x[:, 1]$
 - 12: Unsqueeze the output dimension back to (#tile, #patch=25) via `mu = mu.view(-1, 25)`
-

4.3 Training performance

The model was trained on Florida data for 80000+ epochs using aggregated supervision at the coarse tile level. Predictive accuracy was evaluated on the test set using the coefficient of determination (R^2) and the negative log-likelihood (NLL) computed from aggregated predictions, treating the aggregated target values as ground truth. While α -regularisation and sensitivity punishment terms were included in the loss to stabilise fine-grid predictions, we focus here on the aggregated performance metrics. Across all tested α values, the tile-level R^2 exceeded 0.97, indicating an excellent fit to the aggregated targets.

Figure 11 shows the evolution of the residential tile-level NLL during training for $\alpha = 0.2$, whereas Table 2 summarises the final R^2 and NLL values across all tested α settings.

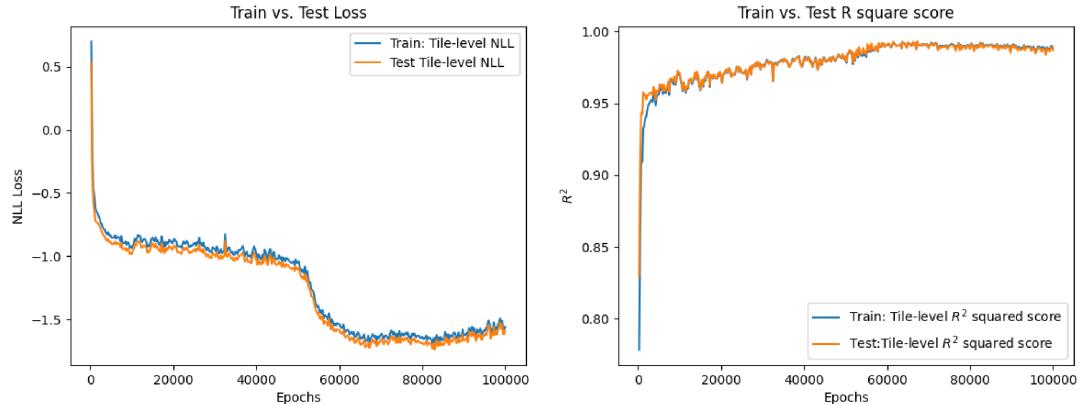


Figure 11: Training curves for $\alpha = 0.15$ over 100,000 epochs. Left: tile-level NLL computed on the testing set. Right: tile-level R^2 fit on the testing set. Both metrics were recorded every 250 epochs. The model shows rapid improvement in the early stages, with both metrics approaching their final values by approximately 60,000 epochs, after which changes are minimal.

Table 2: Tile-level R^2 and NLL on the test set for different α values. Note that the RMSE and NLL are calculated with the scale of empirical GEM standard deviation = 1

α	R^2	RMSE	NLL(loss)
0.10	0.996	0.374	-1.40
0.15	0.992	0.503	-1.49
0.20	0.989	0.609	-1.32
0.25	0.982	0.767	-1.48
0.30	0.975	0.917	-1.42

4.4 Prediction outputs

4.4.1 Re-estimated asset value (mean)

Figure 12 shows re-estimated asset values at 30 arcsec resolution for an area in the Miami metropolitan area in Florida (part of the training region) with $\alpha = 0.15$. The OOD detector was not applied for this example. A grid-level comparison against the original GEM values yields $R^2 \approx 0.92$, whereas the aggregated tile-level fit attains $R^2 \approx 0.99$ on the test set. This difference between tile- and grid-level scores is expected under aggregated supervision: the model preserves tile totals while redistributing values across grids within each tile, i.e., it re-disaggregates within-tile values rather than reproducing the original per-grid labels.

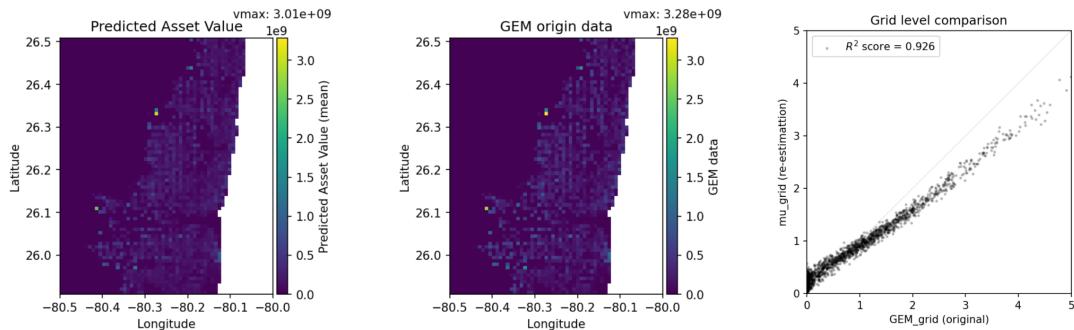


Figure 12: Re-estimated residential asset values at 30 arcsec resolution for the Miami area, Florida (training region).

For New York City (outside the training region), some grids exhibit population densities exceeding those present in the Florida training data. To handle this distributional shift, an out-of-distribution (OOD) detector was applied to flag grids whose covariate values fall outside the training range. Again we choose a model trained with $\alpha = 0.2$ for demonstration. Figure 13 presents three maps for New York City: (a) model re-estimation without OOD, (b) original GEM asset values, and (c) model re-estimation with OOD, where flagged grids retain their original GEM values. The comparison of tile-level R^2 metric tells the OOD is necessary for such framework. While the plot (c), the OOD detector turned on reaches $R^2 \approx 0.98$, we see the one plotted with all models prediction only reach a poor $R^2 \approx 0.42$, meaning it fully breaking the premise of tile-level ground truth.

To illustrate the effect of the α soft-bound across a test area with minimal OOD, Figure 14 shows a series of re-estimated maps for Birmingham, Alabama, USA, for $\alpha = 0.10, 0.15, 0.20, 0.25$ and 0.30 . Given the limited OOD presence in this region, the OOD detector was not applied, and the maps highlight the visual impact of varying α on the re-estimated outputs.

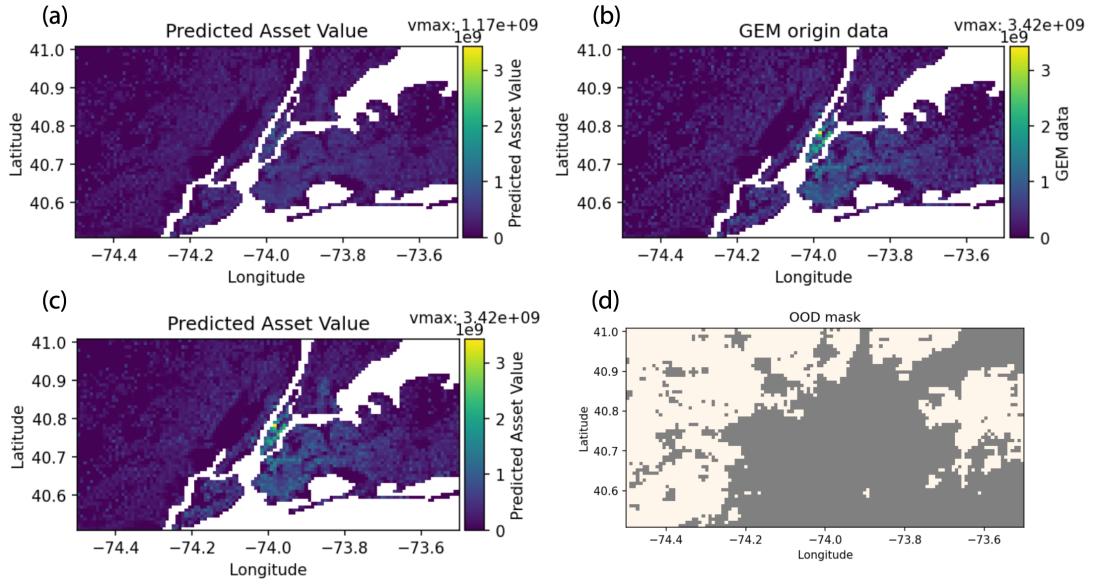


Figure 13: Effect of the OOD detector in New York City, New York state (outside training region, with $\alpha = 0.2$). (a) Model re-estimation without OOD detector. (b) Original GEM asset values. (c) Model re-estimation with OOD detector applied; grids flagged as OOD retain their original GEM values. (d) OOD detector mask

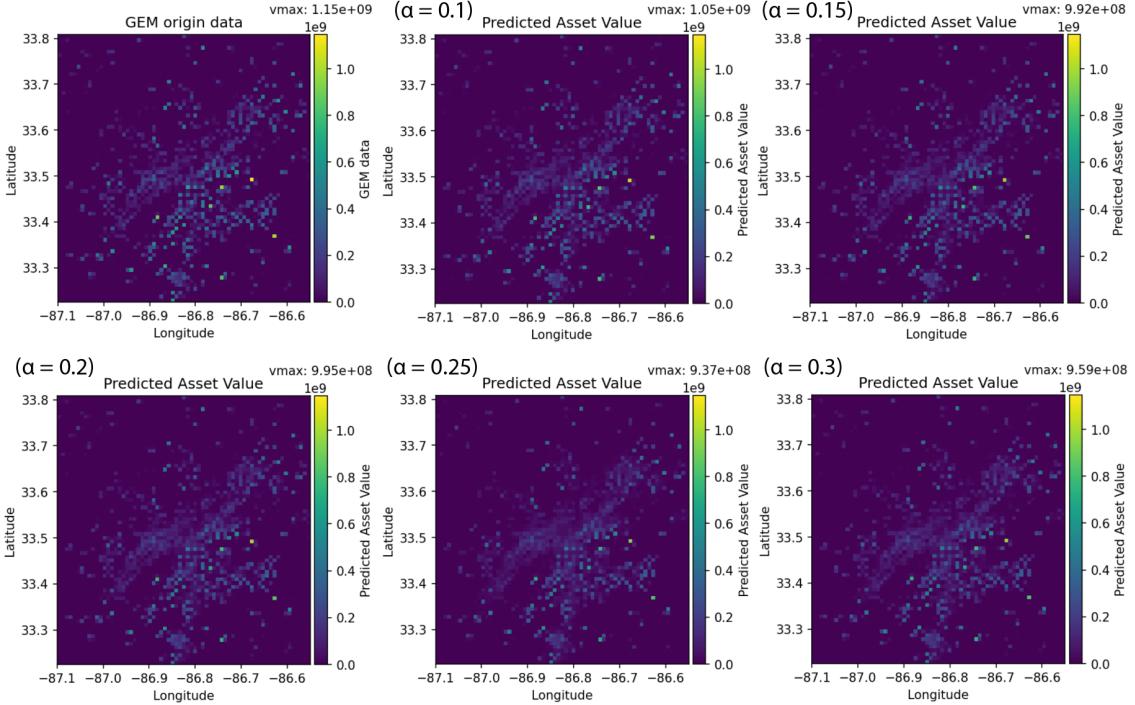


Figure 14: Series of re-estimated asset value maps for Birmingham, Alabama (test region with minimal OOD), across $\alpha \in \{0.10, 0.15, 0.20, 0.25, 0.30\}$. The OOD detector was not applied. Panels share a common colour scale for comparability

4.4.2 Uncertainty prediction (standard deviation)

The model produces a per-grid predictive variance as part of its output, learned via the tile-level negative log-likelihood (NLL) objective. For each 30 arcsec grid cell, the network outputs a mean μ and a variance σ^2 ; the latter quantifies the model’s predictive uncertainty at that location. To visualise uncertainty, we report the uncertainty map as the per-grid standard deviation $\sigma = \sqrt{\sigma^2}$ on the same spatial grid and extent as the re-estimated values, using a separate colour scale. Figure 15 shows uncertainty heatmap in 30 arcsec resolution with $\alpha = 0.2$.

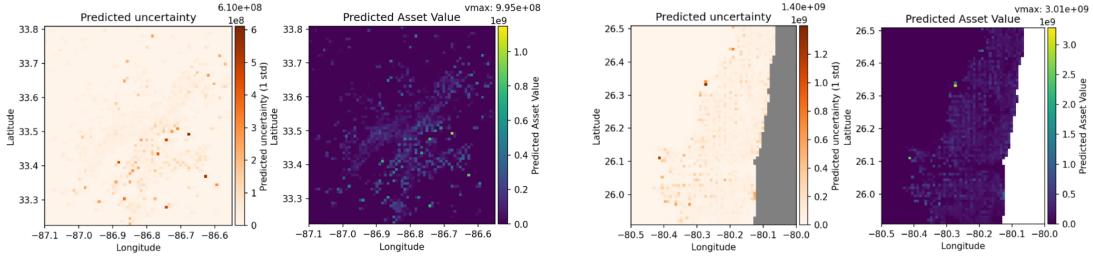


Figure 15: Predicted uncertainty (standard deviation) with predicted asset value at 30 arcsec resolution with $\alpha = 0.2$. (left panel: Birmingham, Alabama, US) and training region (right panel: Miami, Florida, US)

Below shows the empirical coverage of tile-level Gaussian predictive intervals on the training set, computed without applying any OOD. Intervals were formed as $\mu \pm z_p\sigma$, where z_p is the standard normal quantile for nominal two-sided coverage p . Table 3 summarises the observed coverage for $p \in \{0.50, 0.683, 0.80\}$ for a range of $\alpha \in \{0.10, 0.15, 0.20, 0.25, 0.30\}$.

Table 3: Tile-level empirical coverage of Gaussian predictive intervals on the training data, computed for a series of α .

α	$p = 0.50$	$p = 0.683$	$p = 0.80$
0.10	0.480	0.614	0.772
0.15	0.471	0.638	0.791
0.20	0.445	0.612	0.769
0.25	0.486	0.734	0.849
0.30	0.466	0.742	0.865

4.5 Sensitivity

For each prediction we feed the model a $3 \times 5 \times 5$ input patch (three channels: GEM, night lights, population; over a 5×5 grid). Sensitivity simply asks: *if we nudge one element of that input patch a tiny bit, how much does the predicted value change?* Doing this for every element gives $3 \times 5 \times 5 = 75$ sensitivities per patch.

Figure 16 shows the residential GEM sensitivity maps for Miami across α , and Figure 17 shows the corresponding series for Birmingham city. All panels use a consistent colour scale within each region to facilitate comparison across α .

The heatmaps shown here correspond to the sensitivity of the central grid’s prediction with respect to the GEM channel values across the entire 5×5 patch. In other words, each heatmap cell shows how much the predicted central grid value would change if we slightly perturbed the GEM input at the location of that cell in the patch, indicating how ‘robust’ our re-estimation is with respect to the original GEM data.

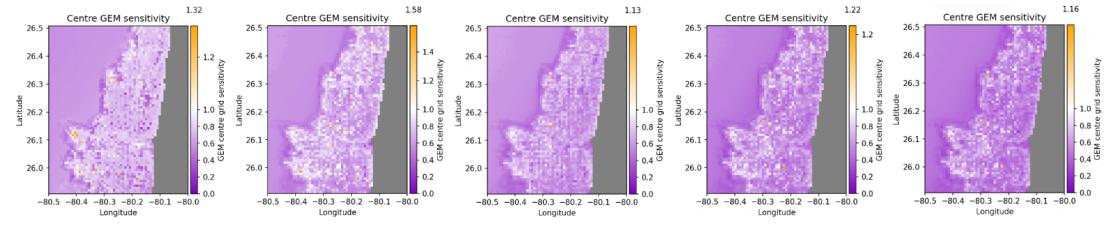


Figure 16: GEM-channel sensitivity heatmaps for Miami, Florida (training region) across $\alpha \in \{0.10, 0.15, 0.20, 0.25, 0.30\}$ (from left to right). Panels share a common colour scale for interval $[0,1]$, purple to white, for comparability.

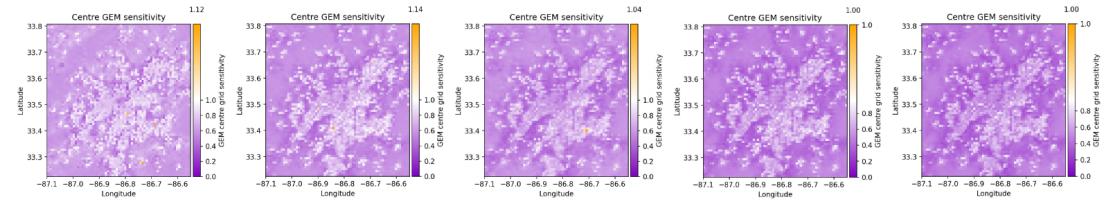


Figure 17: GEM-channel sensitivity heatmaps for Birmingham, Alabama (test region) across $\alpha \in \{0.10, 0.15, 0.20, 0.25, 0.30\}$ (from left to right). Panels share a common colour scale for interval $[0,1]$, purple to white, for comparability.

To summarise the results concisely, we report two sensitivity metrics for each channel: median of the **central-grid sensitivity** (sensitivity at the patch centre) and the **patch-sum sensitivity** (sum of sensitivities across the 5×5 patch), ignoring those originally non-valued grids.

While we have not presented re-estimation results for industrial data owing to the scarcity of GEM records in this category, where most asset value is concentrated in a small area, making re-disaggregation ineffective, we still include sensitivity summaries for completeness. These industrial results (limited coverage) are reported only for the model trained with $\alpha = 0.2$, which will suffice to serve as a feature comparison between residential and industrial assets will be discussed later.

Table 4 and Table 5 list these metrics for residential data in the areas of Birmingham and Miami same as we plot the heat map above, respectively, in all α . Table 6 reports the same metrics for industrial data (limited availability), aggregated over available industrial patches.

Table 4: Residential sensitivity metrics in Birmingham (test): central-grid and patch-sum sensitivities for Lit, Pop, and GEM across α .

α	Central grid			Patch sum		
	Lit	Pop	GEM	Lit	Pop	GEM
0.10	-0.001	0.030	0.761	0.001	0.003	0.838
0.15	-0.002	0.019	0.724	-0.015	-0.004	0.838
0.20	-0.002	0.025	0.666	-0.011	0.019	0.792
0.25	-0.003	0.033	0.613	-0.006	0.023	0.770
0.30	-0.004	0.040	0.568	-0.002	0.036	0.733

Table 5: Residential sensitivity metrics in Miami (train): central-grid and patch-sum sensitivities for Lit, Pop, and GEM across α .

α	Central grid			Patch sum		
	Lit	Pop	GEM	Lit	Pop	GEM
0.10	-0.003	0.025	0.775	-0.015	0.010	0.923
0.15	-0.003	0.021	0.703	-0.011	0.009	0.867
0.20	-0.003	0.028	0.638	-0.019	0.020	0.802
0.25	-0.003	0.031	0.589	-0.013	0.020	0.790
0.30	-0.003	0.037	0.541	-0.016	0.031	0.747

Table 6: Industrial data (limited availability): central-grid and patch-sum sensitivities for Lit, Pop, and GEM, aggregated over testing industrial patches with $\alpha = 0.2$.

α	Central grid			Patch sum		
	Lit	Pop	GEM	Lit	Pop	GEM
0.2	0.001	-0.001	0.307	0.066	-0.039	0.354

5 Discussion, Analysis and Conclusion

5.1 Effect of hyperparameter α on Re-estimated Predictions

Figure 17 illustrates the re-estimated predictions in Alabama for a range of α values. A clear trend emerges: as α increases, the predictions become progressively smoother. This behaviour stems directly from the grid-level regularisation imposed in the loss function, which forms a soft bound requiring predictions to remain within an α -fraction error relative to the original GEM data. A small α enforces a narrow bound, constraining the predictions to adhere closely to the GEM values and producing a sharper heatmap. Conversely, a large α relaxes the bound, granting more freedom for the prediction to deviate, align with the sensitivity-based explanations [12].

From our perspective, smoothing should not be interpreted as a flaw. On the contrary, it supports our working proposition that the true asset footprint may extend beyond the central grid cell into neighbouring cells. A smoother distribution can therefore be more plausible than an overly sharp one, particularly in regions where the original GEM values are uncertain or sparse.

The smoothing effect is also reflected in the sensitivity analysis (Table 4 and Table 5). When comparing the sensitivity of the GEM channel in the central grid versus the patch sum (the sum over the 5×5 patch):

- At low α , the central grid sensitivity is close to the patch sum sensitivity, indicating that the prediction is dominated by the GEM value at the centre.
- At high α , the patch sum sensitivity far exceeds that of the central grid, showing that the model increasingly draws information from the surrounding grids. This shift confirms that a higher α effectively redistributes predictive weight from the central GEM value to its neighbours, consistent with the observed smoothing.

To further illustrate the role of α , Figure 18 presents the case where grid-level regularisation is removed, i.e., $\alpha \rightarrow \infty$. In this setting, the predictions become over-smoothed to the point where fine-grained local variation is lost. This behaviour underscores the importance of tuning α : a value too small risks overfitting to noisy GEM data, while a value too large washes out genuine local structure. Ultimately, the choice of α should reflect how much the user trusts the original GEM data; α becomes a practical control knob for balancing spatial fidelity and smoothness in a rational range.

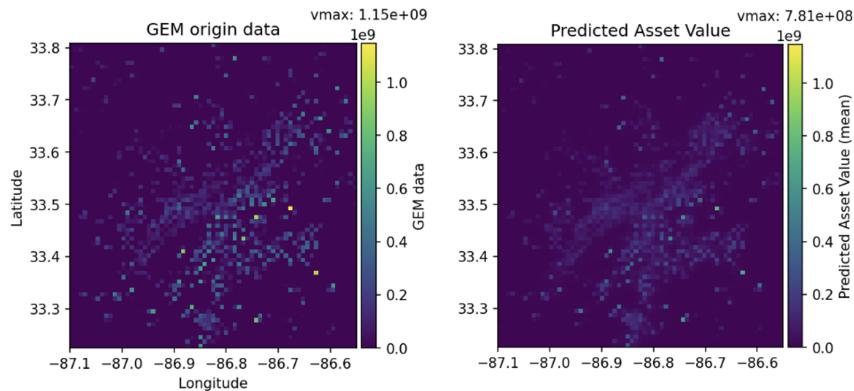


Figure 18: Asset value heatmap for Birmingham, Alabama (test region). Left panel: Original GEM data, right panel: re-estimated asset value without grid-level regularisation term. Panels share a common colour scale for comparability

5.2 Evaluation of Predictive Uncertainty

We assess predictive uncertainty using two perspectives: (1) how well the predicted variances match the observed errors at the tile level, where the Gaussian likelihood is applied during training, and (2) how the predicted variances behave when inspected at the grid level, even though our modelling assumptions do not treat grid-level residuals as ground truth.

Tile-level behaviour Since the likelihood loss is fitted on aggregated tile predictions, we expect predicted variance to be well-calibrated when evaluated at the same level of aggregation. This is confirmed by Table 3, where empirical coverage is reasonably close to nominal p values across α settings, indicating decent convergence of the uncertainty fit. The tile-level scatter plot of predicted standard deviation versus tile-level residual (Figure 19(a)) displays the characteristic “cone” shape expected from a heteroscedastic Gaussian random variable: as predicted σ increases, the possible range of residuals broadens proportionally, producing a narrow cluster near the origin for low σ and a wide spread for high σ .

Grid-level behaviour When shifting to the grid level, the interpretation changes fundamentally. Because our modelling assumption explicitly treats the original GEM values at the grid scale as noisy and potentially unfavourable, the grid-level residual is not a reliable target. Therefore, the absence of high- σ /low-residual points in the grid-level scatter, as shown in Figure 19(b), does not indicate a poor fit. Instead, it simply reflects that grid-level errors are not what the model is optimising for.

However, the grid-level patterns can still be explained. The smoothing effect, driven by MIR and sensitivity punishment together, described in plays a key role: grids with large original GEM values are systematically adjusted downward, while their surrounding lower-value grids are adjusted upward as complementary redistribution (see grid-level scatter plot from Figure 12). Although an individual grid’s residual means little on its own, its contribution to the tile-level residual is not trivial. In fact, an originally high-valued grid almost guarantees a high or at least messy tile-level residual, prompting the model to assign larger predicted variance to such locations. This mechanism is visible in the grid-level plot of the predicted σ versus the GEM value Figure 19(c)), where high-GEM grids tend to have higher predicted variance [12].

When plotting predicted σ against original GEM value for all grids, we observe that in low-GEM regions the variance spreads widely, reflecting the influence of neighbouring grids with very different values. Low-valued grids can therefore show a wide range of uncertainty predictions, especially if they lie next to high-valued cells. As GEM value increases, the predicted variance range narrows, and the correlation between GEM and σ increases. Such pattern extend to highest GEM values until the relationship becomes almost linear (more value \rightarrow more uncertainty).

Taken together, these observations indicate that our training algorithm produces meaningful and reasonably calibrated uncertainty estimates at the tile level (the level at which it is trained). At the grid level, patterns in predicted variance arise from the smoothing-induced redistribution of asset values and the way tile-level residuals are formed. This design naturally limits the model’s ability to produce informative, non-trivial uncertainty estimates for high-valued grids, but retains interpretive power in low-value regions where spatial context has greater influence.

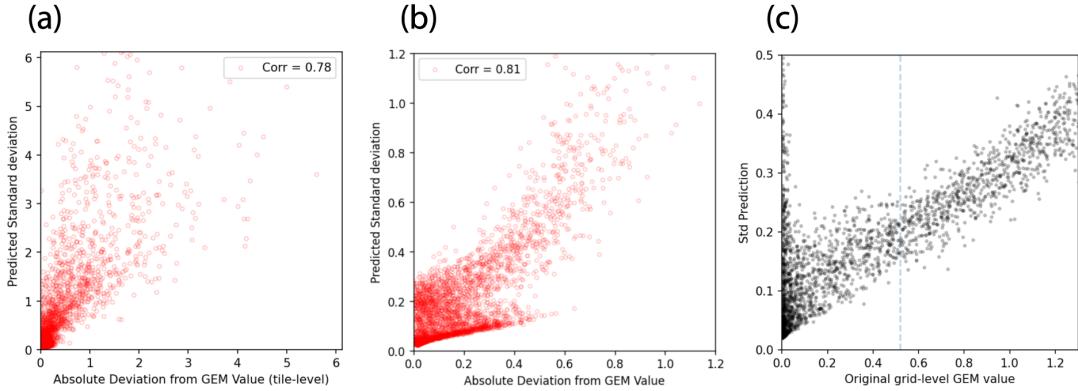


Figure 19: Uncertainty prediction characteristic (training data) (a) Tile-level σ vs. tile-level residual, displaying a well-spread pattern consistent with Gaussian likelihood fitting and the good coverage. (b) Grid-level σ vs. grid-level residual. (c) Grid-level σ vs. GEM value for tiles aggregated from (a), showing a wide spread in low-GEM grids and a narrowing range as GEM increases, with high-GEM grids almost linearly linked to higher σ . This reflects the MIR tendency to assign large variance to grids that strongly influence tile-level residuals. The light-blue dash line indicates 50 % training data fallen in the region on its left.

5.3 Feature importance, sensitivity Analysis: Interpreting Learned Spatial Dependencies

To understand how the model utilizes different input features to re-estimate asset values, we perform a sensitivity analysis. Specifically, we compute the gradient of the predicted mean $\mu_\theta(\mathbf{X}_{\text{patch}})$ with respect to each element in the $3 \times 5 \times 5$ input patch. This reveals how perturbations to each grid-level input affect the model’s output.

Table 5 reports the median of these metrics under series of α , using held-out data from a part of the Miami metropolitan area for residential assets.

Importantly, the model exhibits behaviour that can be interpreted as a form of smoothing. However, unlike naive filters (e.g., box or Gaussian blurs), this smoothing is “learned” and context-aware. The model adapts its spatial influence based on auxiliary signals (Lit, Pop) and does not simply average over neighbouring values. For example, the GEM channel consistently shows a patch-sum sensitivity less than 1.0, indicating that the model does not respond linearly to uniform input shifts across the patch. This distinguishes it from traditional convolution-based smoothing methods (i.e. Gaussian blur).

Moreover, as the regularisation parameter α increases, the model’s dependence on GEM clearly decreases indeed, for which a wider degree of freedom for re-estimation to deviate from original value. This is evident both numerically (through the decreasing sensitivity of GEM in Table 4, Table 5) and visually (see Figure 16 and Figure 17, where the spatial gradient of GEM decreases at higher α (more purple-ish grids)). This shift is accompanied by increased reliance on auxiliary channels, confirming that the model rebalances its attention as its trust in GEM weakens.

Below Figure 20 and Table 7 summarise the median of sensitivity pattern from the OOD-filtered Miami metropolitan area with $\alpha = 0.2$, demonstrating the reliance on auxiliary vary a lot for two asset types. Several additional insights emerge from the spatial patterns are elaborated below, which plausibly agree with the cross-channel data correlations as Figure 9.

Nighttime Light channel: For residential data, light intensity shows weak or negative sensitivity at the central grid and consistently negative influence from adjacent grids. This implies that brighter surroundings may be interpreted as indicative of commercial or mixed-use areas, thereby reducing residential asset. In contrast, for industrial assets, the light channel behaves

differently: while central sensitivity remains weak, surrounding grids exhibit consistent positive influence with broader illumination associated with industrial activity. We will conclude such sensitivity pattern is actually rational since we do not expect much light from residential asset since the light data is shotted midnight, whereas industrial assets, such as factories, keep lit up overnight.

Population channel: For residential assets, population shows a strong positive sensitivity at the centre grid, while adjacent grids together contribute slight negative sensitivity. This suggests a contrast-based pattern: the model prefers population to be locally concentrated at the centre and mildly penalizes spatial dispersion, which may indicate less distinct residential presence. In contrast, the pattern reverses for industrial assets. Population across the entire patch becomes weakly negative in influence, though marginally, possibly reflecting the model’s tendency to associate industrial assets with low surrounding residential density. This interpretation is logically consistent with earlier observations: residential asset predictions are negatively affected by surrounding bright grids (from the light channel), which in turn are positively linked to industrial zones. Together, these patterns suggest that residential and industrial asset types exhibit opposing preferences regarding both population distribution and surrounding urban context.

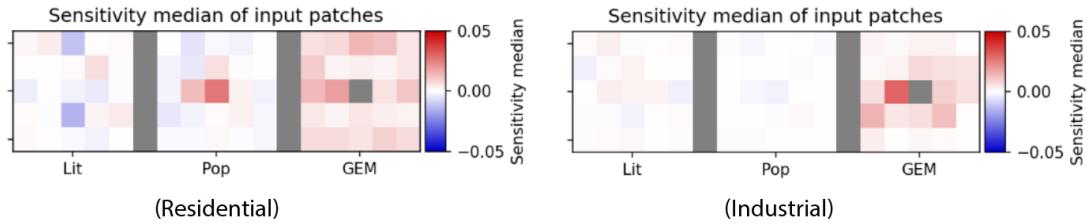


Figure 20: Spatial sensitivity maps for patched channels with ($\alpha = 0.2$). Median sensitivity across all 5×5 patch positions for each channel (GEM, Lit, Pop), computed on held-out data from the Miami metropolitan area. The left panel shows results for residential assets; the right panel shows industrial assets. The relatively high prediction-sensitive central GEM channel grid is deliberately masked out for better visualising other features

Table 7: Summary of Figure 20. Data are drawn from Miami metropolitan area, Florida, US

Asset type ($\alpha = 0.2$)	Central grid			Patch sum		
	Lit	Pop	GEM	Lit	Pop	GEM
Residential ($N_{\text{patch}} = 2513$)	-0.003	0.027	0.624	-0.018	0.019	0.793
Industrial ($N_{\text{patch}} = 2489$)	0.003	-0.003	0.414	0.010	-0.006	0.544

5.4 Resilience to noisy GEM data input

In this subsection, we evaluate the robustness of our model’s predictions to local noise in the GEM input—motivated by the assumption that gridded GEM values can be noisy or imprecise at high spatial resolution. Specifically, we simulate this scenario by injecting multiplicative 10% Gaussian noise into the GEM input at test time:

$$X_G^{\text{perturbed}} = X_G \cdot (1 + \varepsilon), \quad \varepsilon \sim \mathcal{N}(0, (10\% \cdot X_G)^2)$$

and observe how predictions change, grid by grid, across the Miami metropolitan area.

The goal of this experiment is to assess how sensitive the model remains to perturbations in the GEM input under different regularisation levels α . As shown in Figure 21, each scatter plot

compares the injected GEM noise against the resulting prediction shift for individual fine-grid cells. The slope of the fitted regression line quantifies the average propagation of noise into the predictions.

The results clearly demonstrate that higher values of α reduce this propagation effect. As α increases from 0.1 to 0.3, the slope of the fit line decreases significantly, indicating that the model has become less reliant on GEM and more robust to its fluctuations. This aligns with the sensitivity analysis (Table 4) and confirms that the re-estimation algorithm not only down-weights GEM influence but also builds resilience to noisy inputs.

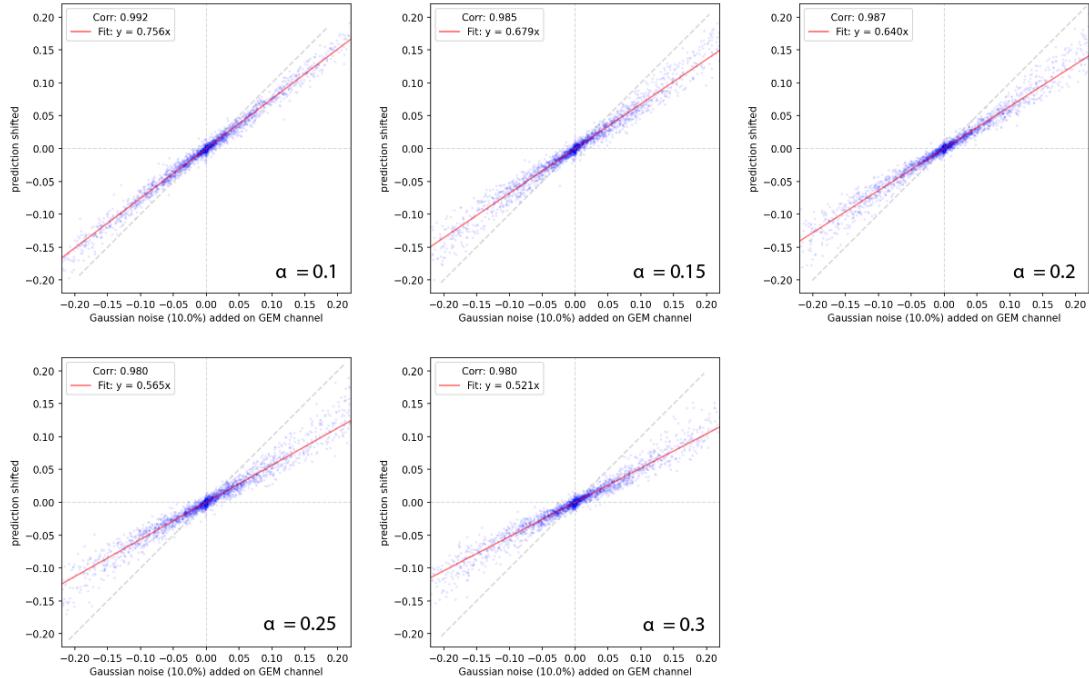


Figure 21: Prediction response to noisy GEM input. We add Gaussian noise (10% standard deviation) to the GEM channel at test time and compare prediction shift to the input perturbation on a per-grid basis across the Miami metropolitan area. Each subplot corresponds to a different regularisation level α . The fitted slope quantifies the propagation of GEM noise into the re-estimated asset predictions. As α increases, the slopes consistently decrease, indicating improved robustness and reduced dependency on potentially unreliable GEM grid inputs.

5.5 Summary of findings

The proposed methodology combines spatially contextual prediction, uncertainty estimation, and feature sensitivity regulation to robustly re-estimate asset values at high spatial resolution.

Key components and insights include:

- **Patch-based input design:** Each prediction is conditioned on a $3 \times 5 \times 5$ input patch, allowing the model to capture localized spatial context from the GEM, night lights, and population layers. This structure enables information sharing across neighbouring grids.
- **Uncertainty-aware prediction:** By modelling both mean and variance as outputs, the model learns aleatoric uncertainty directly from data. Moreover, using a Taylor approximation, we incorporate the impact of uncertain GEM input values on output variance, yielding a richer uncertainty characterization.

- **Aggregated Supervision:** Predictions are trained via a likelihood-based loss at the tile level, aggregating over 25 grids per tile. This reduces overfitting to noisy fine-grid targets and allows the model to infer meaningful structure through spatial aggregation.
- **Learned smoothing with auxiliary support:** Sensitivity analysis shows that the model performs a form of spatial smoothing, but in a feature-aware and data-driven manner. Unlike naive filters, the influence is asymmetric and channel-dependent, with meaningful variation across asset types.
- **Robustness to GEM noise:** Experiments injecting Gaussian noise into the GEM input reveal that models trained with higher α are significantly more robust. The propagation of input noise into predictions is systematically damped, validating the intended effect of regularisation and auxiliary learning.

5.6 Future Work

While the current framework demonstrates strong spatial awareness and robustness, several limitations and avenues for improvement remain.

- **Improving uncertainty estimation at high-value grids:** Our current method for modelling uncertainty with MIR & sensitivity punishment framework tends to produce predictable and non-informative variance estimates in regions with high original GEM asset values. Future work should explore more expressive uncertainty models or hierarchical uncertainty supervision to better reflect spatial heterogeneity in high-value zones.
- **Expanding auxiliary feature sets:** The present model relies solely on nighttime lights and population density as auxiliary covariates. However, recent geospatial machine learning literature [13] [14] has demonstrated the value of incorporating additional spatial signals such as building footprint fractions, average building height, and land-use or zoning maps. These features may provide complementary cues to improve both prediction accuracy and uncertainty calibration.
- **Broader application to geospatial disaggregation tasks:** The modelling approach developed here is well-suited for integration with popular disaggregation frameworks. In particular, high-resolution population mapping and re-estimation of building exposure maps (e.g., GHSL building layer [35]) represent promising domains for extension. Since such datasets often lack direct ground truth and exhibit high regional variability, the uncertainty-aware, auxiliary-driven approach proposed here can offer more robust and interpretable disaggregation pipelines.

References

- [1] M. Marulanda and M. Salgado-Gálvez. Unisdr's global exposure database and disaster risk: current developments and required improvements. *Natural Hazards*, 86:185–187, 2017.
- [2] Charles Scawthorn, Paul Flores, Neil Blais, Hope Seligson, Eric Tate, Stephanie Chang, Edward Mifflin, Will Thomas, James Murphy, Christopher Jones, et al. Hazus-mh flood loss estimation methodology. ii. damage and loss assessment. *Natural Hazards Review*, 7(2):72–81, 2006.
- [3] Catalina Yepes-Estrada, Alejandro Calderon, Catarina Costa, Helen Crowley, Jamal Dabbeek, Maria Camila Hoyos, Luis Martins, Nicole Paul, Anirudh Rao, and Vitor Silva. Global building exposure model for earthquake risk assessment. *Earthquake Spectra*, 39(4):2212–2235, 2023.
- [4] Vitor Silva, Helen Crowley, Marco Pagani, Damiano Monelli, and Rui Pinho. Development of the openquake engine, the global earthquake model's open-source software for seismic risk assessment. *Natural Hazards*, 72(3):1409–1427, 2014.
- [5] C. Aubrecht, R. Gunasekera, and O. Ishizawa. 1 urban characterization aspects and associated spatial scale considerations in the context of global-level exposure modeling : Comparative data evaluation. 2014.
- [6] J. Dabbeek, H. Crowley, V. Silva, G. Weatherill, N. Paul, and C. Nievas. Impact of exposure spatial resolution on seismic loss estimates in regional portfolios. *Bulletin of Earthquake Engineering*, 19:5819 – 5841, 2021.
- [7] Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.
- [8] Soumya Ray and David Page. Multiple instance regression. pages 425–432, 2001.
- [9] James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *The knowledge engineering review*, 25(1):1–25, 2010.
- [10] Martin Riedmiller and A Lernen. Multi layer perceptron. *Machine learning lab special lecture, University of Freiburg*, 24:11–60, 2014.
- [11] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.
- [12] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*, 2017.
- [13] Neal Jean, Marshall Burke, Michael Xie, W Matthew Alampay Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.
- [14] Forrest R Stevens, Andrea E Gaughan, Catherine Linard, and Andrew J Tatem. Disaggregating census data for population mapping using random forests with remotely-sensed and ancillary data. *PloS one*, 10(2):e0107042, 2015.
- [15] Vitor Silva, Desmond Amo-Oduro, Alejandro Calderon, Catarina Costa, Jamal Dabbeek, Venetia Despotaki, Luis Martins, Marco Pagani, Anirudh Rao, Michele Simionato, et al. Development of a global seismic risk model. *Earthquake Spectra*, 36(1_suppl):372–394, 2020.
- [16] Ranasinghe Gunasekera, Oscar A Ishizawa, Ryo Maruyama, Rajeev Murlidharan, and Samitha Keerthiratne. Developing an adaptive global exposure model to support the

- generation of country disaster risk profiles. *Environmental Modelling & Software*, 67:151–159, 2015.
- [17] Daisuke Murakami and Yoshiki Yamagata. Downscaling gdp and population for constructing global ssp-based gridded data. *Sustainability Science*, 14:607–614, 2019.
 - [18] Andrea De Bono and Marco Giovanni Mora. A global exposure model for disaster risk assessment. *Environmental Science & Policy*, 40:132–136, 2014.
 - [19] Samuel Eberenz, Dario Stocker, Thomas Röösli, and David N Bresch. Litpop: Global exposure data for disaster risk assessment: Asset exposure data for global physical risk assessment. 2019.
 - [20] F. Dell’acqua, Paolo Gamba, and Kishor Jaiswal. Spatial aspects of building and population exposure data and their implications for global earthquake exposure modeling. *Natural Hazards*, 68:1291–1309, 2013.
 - [21] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. *Advances in neural information processing systems*, 10, 1997.
 - [22] Kiri L Wagstaff, Terran Lane, and Alex Roper. Multiple-instance regression with structured data. In *2008 IEEE international conference on data mining workshops*, pages 291–300. IEEE, 2008.
 - [23] Xiaoyu Wang, Yuchi Ma, Yijia Xu, Qunying Huang, Zhengwei Yang, and Zhou Zhang. Learning county from pixels: corn yield prediction with attention-weighted multiple instance learning. *International Journal of Remote Sensing*, 46(7):2815–2845, 2025.
 - [24] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.
 - [25] Robert M West. Best practice in statistics: The use of log transformation. *Annals of clinical biochemistry*, 59(3):162–165, 2022.
 - [26] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient back-prop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002.
 - [27] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
 - [28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
 - [29] David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN’94)*, volume 1, pages 55–60. IEEE, 1994.
 - [30] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.
 - [31] Zhuosen Wang, R. Shrestha, M. Roman, and V. Kalb. Nasa’s black marble multiangle nighttime lights temporal composites. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.
 - [32] Michael Carlowicz. Measuring nighttime lights, 2017. Accessed on [Insert Date].
 - [33] Tong Lee, Steven D Miller, Rongjun Li, and Greg Kopp. Calibration and validation of suomi npp viirs day-night band radiances. *IEEE Transactions on Geoscience and Remote Sensing*, 52(10):5748–5760, 2014.

- [34] E. Doxsey-Whitfield, K. Macmanus, Susana B. Adamo, L. Pistolesi, J. Squires, Olena Borkovska, and S. Baptista. Taking advantage of the improved availability of census data: A first look at the gridded population of the world, version 4. *Papers in Applied Geography*, 1:226 – 234, 2015.
- [35] Martino Pesaresi and Panagiotis Politis. Ghs-built-s r2023a - ghs built-up surface grid, derived from sentinel2 composite and landsat, multitemporal (1975-2030), 2023. [Dataset].

Acknowledgments

I acknowledge the use of generative AI (ChatGPT, GPT-4o) only for linguistic refinement and translation purposes to ensure clarity and readability of this review.