DOI: 10.1111/cgf.70390

EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 2

Authoring Terrestrial Planets with Diffusion Models

Oliver Borg[1], James Gain[1], Éric Guérin[2], Adrien Peytavie[3], Marie-Paule Cani[4], Eric Galin[3], Guillaume Cordonnier[5]

[1]University of Cape Town, South Africa
[2]Univ Lyon, INSA-Lyon, CNRS, LIRIS, France
[3]Univ Lyon, Université Lyon 1, CNRS, LIRIS, France
[4]LIX, Ecole Polytechnique/CNRS, IP Paris, Palaiseau, France
[5]Inria, Université Côte d'Azur, France

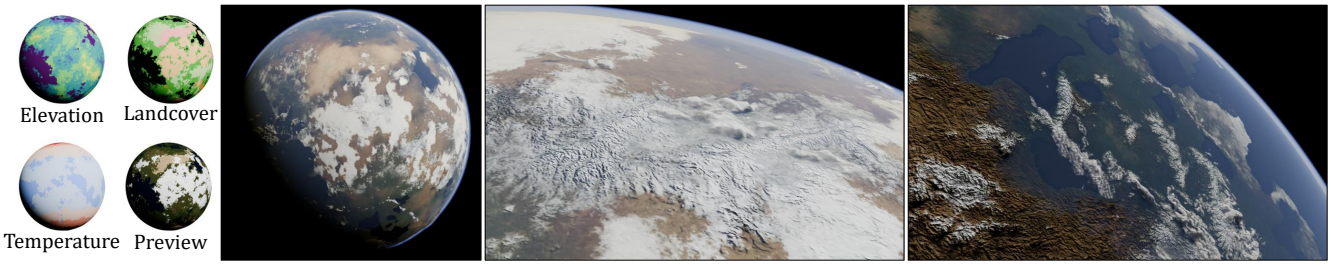

Figure 1: *From interactively-designed environmental atlases with an accompanying satellite-image preview (left), a globally consistent planet is generated using diffusion (center left). Thanks to its high resolution, the planet can then be explored via flyovers as low as* 3000*km above the planet surface (right and center right).*

Abstract

To support the design and subsequent generation of terrestrial planets for use in the creative media, we propose a solution that employs a generative model trained on satellite data from planetary bodies with a defined solid surface, such as the Earth and Mars. A user sketches coarse elevation, landcover, temperature, and precipitation directly onto a globe. Our model then infers high-resolution heightmap and surface appearance layers at planetary scales, with sufficient detail to enable animated flyovers within the exosphere at a distance of a few thousand kilometers from the planet surface. We address the issue of distortion in the mapping from atlas to globe using a quadsphere representation, and the consistency of large-scale geomorphological features by extracting a global river network from the sketch inputs and providing this as conditioning to the diffusion. As our results demonstrate, our generative model provides a balance between: authoring control through a multi-layer painting interface with a satellite image pre-visualization; computation times proportional to the surface area being generated; landscape diversity, displaying, without repetition artefacts, the full range of elevation and landcover features drawn from multiple source planets, and geomorphological plausibility through the provision of a consistent uninterrupted exorheic global river network, where the input sketches allow.

CCS Concepts

• ***Computing methodologies*** → ***Machine learning; Computer graphics; Shape modeling;***

1. Introduction

Computer-generated planets are a staple of speculative media and appear widely in films and video games. They are also useful in an educational context to illustrate the possible structure of our Earth in the distant past or far future.

To generate terrestrial planets and satellites – *i.e.*, those with a defined rocky surface, similar to the Earth, the Moon, and Mars – a digital artist typically creates successive layers of heightfield elevation, landcover texture, and atmosphere (where present) either manually, procedurally, or by simulation. The key issue in this authoring process is that the appearance of terrestrial planets is dictated by complex processes. For instance, the Earth's geosphere alone is impacted by plate tectonics, uplift, meteor impact, volcanic eruption, and glacial, hydraulic, and mass-wasting erosion, among other effects. This is without even considering the bio-

sphere and the range of feedback cycles that lead to the formation of different biomes. A knowledgeable artist could reproduce the resulting detail, but it would be both time-consuming and painstaking. Existing procedural and simulation methods do not capture the full range of these complex interconnected formative processes, although there has been recent work on procedurally emulating plate tectonics [CPGG19] at the global scale.

Recently, generative models have been successfully applied to meso-scale terrain authoring [GDG*17,LGP*23], where the height field extent is generally limited to several dozen kilometers. These models are able to bypass complex geological formation processes by instead learning the features that make up realistic terrain implicitly from data. Taking this as inspiration, our goal is to explore the use of generative image models as an authoring tool for full terrestrial planets. More specifically, can tiled diffusion models be adapted to the large-scale multilayered task of planet authoring?

Developing an effective authoring solution requires balancing several considerations, including user control, computational efficiency, diversity, and plausibility. In terms of user control, we seek to support the specification of significant features, such as the shape of continents and the layout of mountain chains and major biomes. Finer control, such as the placement of particular landcover, should be available when needed. From an efficiency perspective, planet generation is particularly challenging since it requires combining large-scale generation with high-resolution results. In this regard, we target a resolution of 2.4km per pixel to allow exosphere-level flyovers at a few thousand kilometers from the planet surface, in which the planet dominates the entire view. Current generative models are not able to provide an interactive response at such resolutions. This makes the use of intuitive yet reliable authoring tools critical, so that users can not only accurately specify their intentions but also preview the full planet, before committing to any lengthy generation process. In support of diversity, it should be possible to create planets showing the full range of geomorphological features and landcover classes present in the training data, in proportions chosen by the artist. Furthermore, even if one landcover class or terrain feature is chosen as dominant (*e.g.*, a planet composed entirely of islands), the result should be free from unrealistic structural repetition. Finally, in terms of plausibility, the emergent detail in the elevation and surface detail should be mutually consistent, irrespective of the scale at which the planet is observed.

Achieving these goals requires overcoming two types of technical challenges. The first relates to the structure of the input and output data. Let us consider the output scale: Extending diffusion-based terrain generation to an Earth-sized planet with a sampling resolution of 2.4×2.4 km per pixel requires the equivalent of a $9{,}410 \times 9{,}410$ image, which means that run times are correspondingly inflated and need to be managed. Moreover, while smaller-scale terrains can be adequately approximated by a rectangular image, generating a spherical planet requires explicit consideration of the mapping from atlas image to globe, *i.e.*, finding a scheme that minimises the per-pixel distortion inherent in any such transformation. While diffusion techniques exist that can generate images at such a resolution, we require the image to seamlessly map to a spherical representation with minimal distortion, necessitating a blended tiling approach. In addition, such generation is prone to insufficient data: currently, the only terrestrial planets and satellites available as data sources (*i.e.*, with high-resolution elevation maps and landcover textures) are the Earth, the Moon, and Mars. Even for the Earth, which is the largest, a tile resolution of 256×256 pixels at our chosen resolution only provides 2048 unique tiles for training, more than half of which lie in the ocean. This is insufficient for most diffusion models, meaning that overfitting is a real danger.

The second category of challenge relates to the plausibility goal, which is particularly involved in the case of terrestrial planets. Indeed, the emergent details should not only be locally geomorphologically and botanically consistent – meaning that the elevation and landcover should match – but also maintain consistency at a larger scale. For instance, orthogonal erosion channels should be derived for long mountain chains spanning several tiles. Rivers should be uninterrupted and, barring endorheic basins, flow consistently downstream into the sea. Note that while only the water surface of the largest rivers are visible in the input data, their impact can be observed both in terms of elevation changes (*e.g.*, successive V-shaped valleys) and riparian landcover on riverbanks (*e.g.*, the green banks of the Nile). Given that such long rivers can span hundreds of kilometers, they represent a global feature, meaning that they need to span potentially many tiles.

Our solution for planet-scale authoring balances user control, computation times, landscape diversity, and geomorphological plausibility. The core mechanism involves inferring a complete high-resolution planet from a coarse multi-layer user sketch of elevation, temperature, landcover, and precipitation. This is achieved by generating many small-scale image tiles containing elevation and satellite image data using a diffusion model and then blending them into a quadsphere representation of the planet. A quadsphere, or, more formally, a quadrilateralised spherical cube map projection [CO75], provides a fully connected, consistently indexed mapping with less distortion at the poles and more even distribution elsewhere than most other equal area projections. A painting metaphor is adopted for authoring the planet, with a palette of discrete bands (for temperature and elevation) and categories (for landcover) designed to better match an author's conceptualisation. For example, while a tropical rainforest is expressed in the output satellite image in various shades of green that are difficult for a user to replicate, it is represented in our painting interface as a forest landcover class combined with high temperature. Even though the different input layers are separate and distinct, they are effectively fused by the diffusion model during inference. Additional authoring tools, for outlining continents and introducing simplex noise, are included to speed up the painting process.

An explicit global river network can provide much needed additional conditioning to improve non-local geomorphological consistency, but the required fine-scale dendritic structure is very time consuming to author. Instead, we automatically derive a river network from the authored elevation sketch, by converting it to an upsampled continuous heightfield, over which we run a flow simulation. This generates a curvilinear river network with per-pixel upstream area information, representing the flow contribution from the upstream catchment. User control over rivers is provided through a precipitation sketch. A particular sketched precipitation

value is combined with the upstream area of any river within its area. This effectively mimics the effect of variations in rainfall, thereby suppressing or exaggerating rivers.

To sum up, our main contributions are:

1. An authoring framework enabling users to specify the main landscape features by sketching landcover, temperature, elevation, and precipitation. The latter is applied to a consistent global river network automatically derived through flow simulation.
2. A diffusion model specialised to terrestrial planet generation, based on the blended tiling of a subdivided quad-sphere. This model is guided by the combined globe sketch inputs, while incorporating additional conditioning from the river network.

As Figure 1 shows, our solution achieves planet generation with global, independent user control of multiple environmental factors, while evidencing fine-grained consistency at high resolutions. The results are validated in terms of quality, diversity and adherence to the input maps.

2. Previous work

The question of how to generate digital terrain has been extensively investigated for many years. Solutions typically use or combine procedural methods that algorithmically replicate the desired final appearance, simulation methods that iteratively reproduce erosion and other geomorphological processes, or example-based synthesis processes building on the transformation of scanned real-world data.

Rather than summarising this substantial body of work, we refer the reader to the survey of Galin *et al.* [GGP*19], and instead focus on the generation and authoring of entire digital planets. However, we first consider the application of generative machine learning to meso-scale terrains, since this was an inspiration for our work.

2.1. Terrain authoring with machine learning

There is a rich corpus of scanned elevation data for different regions of the Earth available from multiple sources in the form of gridded digital elevation models (DEMs), which in some cases have a resolution as fine as 50cm or 1m per pixel. This makes generative machine learning models, such as conditional generative adversarial networks (CGANs) and diffusion models an attractive proposition for synthesising terrain. The task can be posed as an image-to-image translation, with the input image providing user control through a top-down sketch, and the output image representing a heightfield terrain.

The first machine learning approaches to terrain synthesis were built on convolutional neural networks [Rah18, ACA18, KSR20], but these early experiments lacked authoring control and output detail. This was overcome through the use of CGANs. After the initial Pix2pix-inspired application of CGANs to terrain [GDG*17], work was undertaken to improve style diversity and control [ZLB*19, PPB*23, LLXT22], support multilayer output, including landcover [SW19, VRGZS20, ZLZ*22], and incorporate architectural enhancements, such as self-attention, latent space encoding, and variable levels of detail [CXY*22, NJSR22, JSR24]. This included extension to high-resolution, multi-scale or unbounded scenes [FAW19, LCL*22, WDJ*24]. From an efficiency and realism standpoint CGANs are well suited to interactive authoring of meso-scale terrains. Their weaknesses are instability during training and a tendency to introduce repetition artefacts in areas that are under-specified by the user [GDG*17]. With recent improvements in sampling speed and image quality [HJA20, ND21, SME20, SH22, XKV22], diffusion models offer a compelling alternative. Accordingly, a number of authors have applied diffusion models to terrain, with a focus on authoring [LGP*23], as well as multi-phase [HHM*24] and multi-resolution [JSR22] synthesis. While most of these methods specifically focus on elevation, other diffusion models have been trained to output both topography and the chromatic bands of satellite imagery. MESA [CMR*25] leverages latent diffusion trained on global Copernicus remote sensing data to synthesise 2.5D terrain representations conditioned on natural language prompts. An alternative to text is climate conditioning, for instance, using Köppen's climate classification to texture a DEM with seasonal variations [KEK24]. DiffusionSat [KLZ*24] generalises satellite imagery generation by proposing a foundational model that outputs multispectral images conditioned on metadata, such as geolocalisation, cloud cover, etc. While these methods specifically target satellite image generation, they do not operate at the scale of an entire planet.

An important perspective is the authoring control offered by these different approaches, encompassing painting elevation bands and terrain types [GDG*17, VRGZS20, LLXT22, PPB*23], top-down sketching of landform features, such as ridges, rivers, and peaks [GDG*17, CXY*22, LGP*23, HHM*24], and the specification of an overarching landscape class [ZLB*19, LGP*23]. In the more general space of diffusion-based image generation methods, explicit user control has been explored through additional conditioning inputs. These methods generally rely on a pre-trained diffusion model, complemented by a component fine-tuned to the control modality. For instance, Voynov *et al.* [VACO23] employ a latent guidance predictor trained to control the diffusion model with sketches, while ControlNet [ZRA23] uses a twin network, adaptable to any desired input.

The memory costs of generative models place a practical limit on the size of the generated terrains (typically in the range $1-30$km on a side), which depends on the pixel sampling resolution (usually $1-30$m per pixel). This limitation is generally overcome by tiling, in which individual terrain patches are synthesised and merged to create a larger whole. Several strategies are possible: adapting the pixel sampling resolution and inserting tiles into a hierarchical spatial decomposition scheme, such as a quadtree [JSR22, JSR24] or merging tiles with a uniform resolution using various matching and blending approaches, such as Poisson blending [CVG*15] or latent-space matching [FAW19]. This scale issue applies equally to diffusion models, as memory and attention costs grow rapidly with image resolution. Some approaches adapt the diffusion architecture to enable ultra-high-resolution synthesis [YJH*24, ZHL*25], or rely on super-resolution, for instance in the latent space [JHKK25]. These high-resolution methods operate in a fixed frame, which is inconvenient for projection onto a planetary surface. In contrast, tiling-based methods, such as Tiled Diffusion [MF25], decompose large images into overlapping regions whose diffusion trajectories

are fused or blended to maintain consistency. We adapt this approach to planetary projection through a quadsphere representation. However, Tiled Diffusion relies on narrow overlap regions, which limits information propagation to distant tiles. We therefore prefer the DiffInfinite approach [ANH*23], which promotes long-range interactions by diffusing randomly sampled patches and is easily parallelisable. A similar random patch selection is used in MultiDiffusion [BTYLD23] to guide generation through multiple diffusion paths, but it remains constrained to a bounded frame.

Our work is closest in spirit to that of Lochner *et al.* [LGP*23] in that we use a similar diffusion model. However, we completely replace the front-end conditioning since their vector sketch approach for drawing ridges, valleys, and flat areas combined with a style embedding is ill suited to authoring at the planet scale. Furthermore, the problem we tackle requires additional mechanisms for mapping to the globe and tiling to handle the shear size of the output data.

2.2. Planet generation

While there have been no previous generative machine learning solutions to planet-scale authoring, the problem has been tackled in other ways. Early procedural efforts, such as midpoint displacement [FFC82] with GPU acceleration [BW06], are rapid but lack diversity. Derzapf *et al.* [DGGK11] recognise that a coherent river network is key to believability and use this as a foundation for their procedural real-time adaptive planet generator. They start from a coarse mesh and use edge splits to introduce detail that respects the prevalent emergent properties of river networks, for instance being centered in valleys and terminating in water bodies. Such procedural techniques are also widely used in commercial products, such as Terragen [Pla], because their shortcomings in diversity and realism are offset by memory and computation efficiency.

Plate tectonics are one of the primary geophysical forces at the planetary scale and Cortial *et al.* [CPGG19] exploit this through an iterative user-controlled procedural emulation of the drift, collision, and deformation of tectonic plates, leading to the formation of continents, oceanic ridges, significant mountain ranges, and island arcs. Users can guide plate movement, which takes place at interactive rates, but because of the simulation-like formulation it can be difficult to achieve an intended outcome. Another issue is that the results lack detail at finer scales. To combat this the authors develop [CPGG20] a hyperamplification scheme that procedurally subdivides a planetary surface from a 50km to 50cm sampling resolution in real time. The introduced detail is sensitive to scale, context and hydrology. A point of similarity with our work and departure from that of Derzapf *et al.* [DGGK11] is the extent of control that the user has over the coarse initial surface through their provision of elevation, humidity, orogeny, and landcover maps.

The fundamental issue with attempting to simulate or procedurally emulate planet formation is the bewildering range and complexity of geophysical, chemical and biological processes that dictate the final appearance. We sidestep this issue, and achieve greater realism, through diffusion model inference based on satellite and elevation data, which by its nature already encodes these processes. Moreover, directly generating the target state of a planet allows for greater authoring precision, in contrast with planetary formation processes, which only indirectly influence the target outcome.

3. Overview

The essence of our planet authoring framework (see Figure 2) is the conversion through an image-to-image diffusion model of a set of low-resolution feature sketches provided by the user into high-resolution elevation and satellite maps covering the globe, suitable for downstream 3D rendering.

The challenge on the input side is to provide sufficient detail in terms of feature classes and pixel resolution to enable a user to adequately capture major landscape features, while ensuring that authoring does not become too complex or time consuming. For this purpose, we settled on a set of discretised sketches for elevation (5 bands), landcover (8 classes, including ocean, tree cover, grassland, bare, shrubland, cropland, snow and ice, and a reserved category for Mars data), temperature (5 bands), and precipitation (a scalar value), with a per-pixel extent of 78 km (6048 km^2). We adopt a standard painting metaphor for the interface, with a palette of feature classes, brushes applied directly onto a globe, and additional tools for painting simplex noise and drawing the outlines of continents (see Section 4).

On the output side we produce detailed elevation and satellite image maps since these are essential to a rendered portrayal of a planet: the former because topography, particularly in mountainous regions, such as the Himalayas on Earth and Olympus Mons on Mars, is visible from space, and the latter because it provides essential surface colour and texture. Our choice of a per-pixel output extent of 2.4 km (a 32-fold upsampling) was motivated by a desire to provide planetary views from as close as 2000 km outside the atmosphere. This value of 2.4 km is a consequence of resizing our output data from $21,600 \times 10,800$ to $16,384 \times 8,192$ to allow convenient base-2 upscaling. Dividing the circumference of 40,075 km by the atlas width of 16,384 leads to a resolution of 2.4 km per pixel at the equator. This scale also provides sufficient data, with some augmentation through flips and rotations, for training our diffusion model. We decided to outsource the generation of the cloud layer because this can be achieved through other mechanisms, such as simulation.

Our diffusion model is strengthened by an additional derived input in the form of a global river network (see Section 5) at the 2.4 km per pixel scale. This is necessary to ensure that rivers are hydrologically consistent (uninterrupted, flow directional, and sea and ocean terminating) when performing localised inference. Even if a river's width falls below the 1 pixel visibility threshold, it can still have a significant secondary impact through hydrological erosion of the elevation and riverbank enrichment of the landcover. The river network is a raster of upstream area values obtained from the elevation sketch using a flow simulation. The upstream area values are adjusted according to the user's precipitation sketch, and the results are rendered into a river network map (see Figure 2) collated as input alongside the user's sketches, after which the precipitation sketch is dropped from the pipeline.

In the next phase (see Section 6), our diffusion model generates a detailed elevation and satellite image atlas for the planet. To avoid distortion and seams when mapping from this atlas to the globe, we use the well-recognised quadrilateralised spherical cube map (quadsphere) as the underlying data structure, along with its attendant curvilinear equal area projection. This structure can be

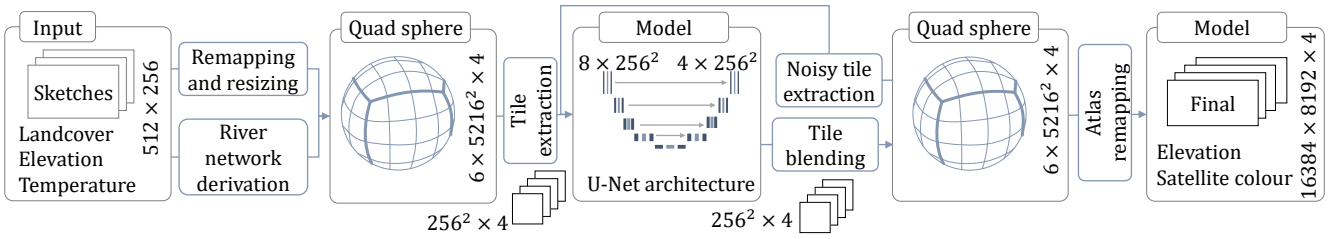

Figure 2: *System overview: the input globe sketches (landcover, elevation, and temperature) along with a derived river network are fed into a diffusion model using tiles on a quad-sphere. The results are mapped and blended onto the quad-sphere to produce a final planet with detailed elevation and satellite colour.*

configured for planets of different diameters through subdivision of the cube faces, terminating when the fixed 2.4 km pixel scale is reached. Inspired by the DiffInfinite strategy [ANH*23], we generate multiple overlapping tiles with random placement and blend them on each diffusion timestep, by placing pixels only where none have been previously generated, which avoids visible seams and blurring.

Of necessity this leads to lengthy execution times, of up to a couple of hours on an H100 NVidia GPU (see Section 7). Fortunately, due to our tiled approach, subsequent editing and regeneration of sub-areas can be performed in a proportionately-reduced time, making iterative design more feasible. Also, we extract a coarse version of the satellite image from a combination of the landcover and temperature sketches, to serve as a pre-visualisation of the final output (as further discussed in Section 4).

4. Authoring

Their inherent scale and the need to create a wide variety of landform patterns makes the authoring of planets both time-consuming and technically difficult. Consequently, in designing our interface we seek to provide control without burdening the user with unnecessary complexity and workload, thus offering a balance between the competing aims of specificity and tersity. We adopt a sketching interface, in which users paint features directly onto coarse globes at a pixel size of 78.3×78.3 km. This resolution is chosen because it is sufficient for inference to capture and elaborate on major details, such as continent shape and placement of mountain chains and landcover, while providing the equivalent of a manageable canvas resolution of 512×256 for an earth-scale planet.

We provide four different categories of globe sketches: elevation, landcover, temperature, and precipitation (see Figures 1 and 4). This split enables a reasonable limit of $5-8$ brush types and associated semantics for each globe sketch. The alternative, a direct mapping of inputs to outputs, would require users to painstakingly paint the many colours of a satellite image devoid of associated meaning. For example, the satellite version of a boreal forest would involve painting subtle shades of dark green, while in our interface it is achieved by painting a forest class into the landcover and a cool band into temperature. This also enables separated control, such as creating a frozen earth simply by overpainting colder temperatures (see Figure 9). The downside is that the user is responsible for maintaining a rough correspondence of feature placement between categories. From an authoring perspective, we compensate for this by allowing the user to view 3 globes with synchronised cursor positions, viewpoints and zoom levels. The primary globe is used for authoring and the other two have dropdown menus to switch between the other sketch types, generated outputs or even Earth data for reference. In this case the globes may be decoupled to allow a different area of Earth to provide reference for a user trying to mimic particular terrain. We also provide a biome sketch preview at the same resolution as the user sketches. From an inference perspective, we rely on the recognised ability of diffusion models to compensate for any remaining mismatches. In practice, we find that the model falls back on the data distribution in the case of mismatched or highly underrepresented combinations. For example layering the snow and ice class with the highest temperature band will result in a barren satellite image with temperature dominating. To help the user build intuition as to the likelihood of different temperature-biome combinations, we provide a complementary uncertainty map (see Figure 3), which colour codes each pixel in the sketch as red, orange, yellow or green, depending on the frequency of occurrence of the given class combination in the training data. At one extreme red means the combination does not occur, while at the other extreme green represents a class with occurrence greater than the mean. The intention behind this uncertainty map is to provide guidance rather than enforcement.

We observed early during interface design, that many planetary regions are a mix of two brush types, such as mountains mixed with foothills, or clumps of shrubland within broader grassland. Based on this we include simplex noise brushes, which combine different brush types at configurable frequencies, so that a user can concentrate on high-level features but still produce detail.

Finally, to compensate for the lengthy generation times, we add a non-editable preview globe sketch that portrays a pre-visualisation of the satellite image so that users can commit to the generation process after viewing an indication of the final result.

Next, we provide further detail on each globe sketch.

Landcover input consists of a palette of eight classes: ocean, tree cover, grassland, bare, shrubland, cropland, and snow and ice. An extra Mars class can also be painted alongside other landcover. We decided to exclude some biomes by merging them with others (e.g., wetland and mangroves are classed as tree cover) because of their relatively small coverage in the source data.

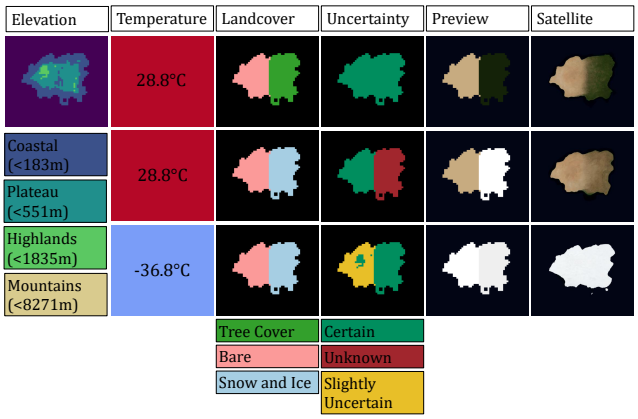

Figure 3: *Our uncertainty map helps users understand when a given input is unlikely to produce the expected result. For instance, Snow and Ice with a high temperature does not produce snow, but instead falls back to the temperature sketch and surrounding terrain producing a dry barren terrain.*

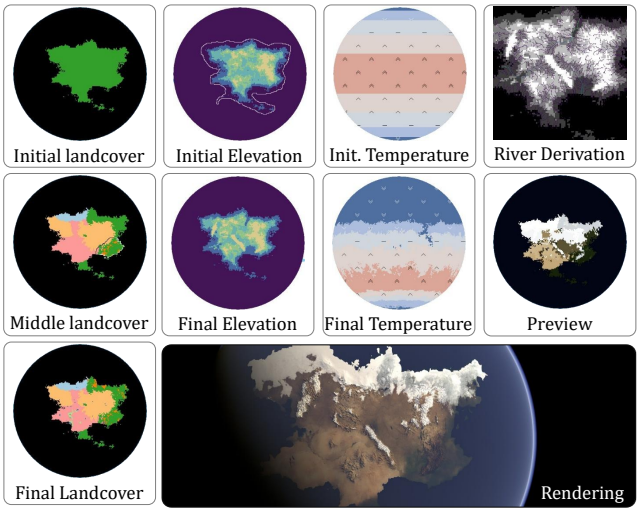

Figure 4: *Progressive authoring. Early, intermediate, and final globe sketches in the modelling of a small continent. Note the pen outline of the continent in the initial DEM. The river network, satellite preview, and final result are also shown.*

Temperature input has five bands, which evenly divide the global minimum to maximum monthly temperature range of -45°C to 37°C. The most common use case is to base temperature on latitude so we provide an interface for defining circular latitudinal bands, with controllable noise introduced to add variety. Furthermore, we consult the elevation sketch to adjust temperature with altitude based on a user-defined lapse rate. While this conveniently automates the relationship between elevation and temperature such that mountains generally have a lower surface temperature than the surrounding terrain, it can be disabled if it fails to match the users intentions. As always, the user can directly paint temperature as required. For example, to ensure a thick layer of snow, it is best to align both snow and ice in landcover and the coldest band in temperature. Figure 5 shows the impact of temperature on different landcover classes.

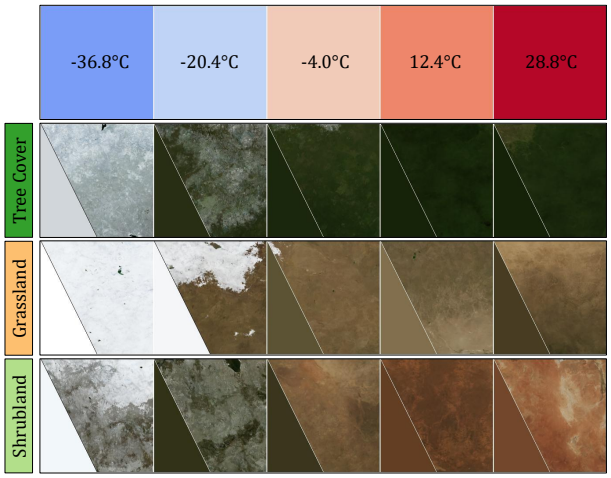

Figure 5: *A grid of generated satellite tiles and their previews for three different landcover classes with increasing average temperature values.*

Elevation input gives a rough indication of the geomorphological form of the planet surface. It consists of five altitude bands: ocean (0m), coastal ($0-183$m), plateau ($183-551$m), highlands ($551-1835$m), and mountain ranges ($1835-8271$m). These unequal ranges are designed to balance the distribution of elevation data between bands. In addition to the standard two-tone simplex brush, we also include a pen for roughly outlining the shape of a continent (see Figure 4), which is then infilled using a distance field to blend in noise thereby creating coastline detail.

Precipitation input, with an associated effect on the river network, consists of a globe sketch of scalar values representing rainfall. This contributes, together with topography, to river prominence in different regions, but we also provide secondary control enabling users to further swell or shrink local rivers. These two layers of control are interleaved with pre-set values for precipitation, and a flow simulation for river network computation, as detailed in Section 5, to achieve consistency between hydrology and other environmental factors. To automatically populate the precipitation map, we first build a table of precipitation values in which each combination of landcover and temperature is assigned a value obtained by taking the mean precipitation for each class from an overlayed precipitation map. Then for each map location the local landcover and temperature is used to index a precipitation value in this table.

The satellite preview is an output globe derived from the landcover and temperature sketches with the purpose of providing a coarse pre-visualisation of the final satellite image generated by the diffusion model. We build a satellite colour table, similar in structure to the precipitation table, by first finding the mode of the satellite colour for each combination of landcover and discretised

temperature. We then optimise L2 loss against the original satellite data to choose the number of temperature bands between 4 and 8, leading to a choice of 5. This assigns a unique colour to each combination of landcover and discretised temperature. For instance the tree cover class combined with the hottest temperature band returns a strong green representative of a rainforest biome.

5. Hydrological consistency

A key failing of many existing terrain synthesis systems is that they do not address hydrological consistency [SD22]. At some level the generation process needs to be explicitly aware of the river network that is knitted into the landscape. While some control should be provided, it is unreasonable to expect a user to draw the fine branching detail of such a network by hand, particularly at the planetary scale.

We handle this dichotomy by interleaving user control with simulation, which is done in three phases: (1) high-level control of precipitation, (2) river flow simulation based on precipitation and elevation, (3) finer-grained user edits to scale or cut-off rivers, without changing their route. The resulting coherent, detailed river network, augmented with water volume information at each river pixel, is then used for conditioning the generation process.

Sketching high-level precipitation: This first stage allows users to design a precipitation globe sketch, while maintaining the desired level of coherence with the previously sketched globes, namely landcover and temperature. Indeed, rather than starting from a uniform map, the initial values of the precipitation globe can be automatically derived by consulting a mean monthly precipitation table, indexed on a combination of landcover and temperature. The user then has two options: Either directly edit the initial precipitation globe sketch, as seen in Figure 7, or indirectly adjust precipitation, by altering indexed values in the precipitation table, which maintains the rainfall consistency of the same biome appearing in different regions of the planet.

River flow simulation: To maintain consistency between elevation and river data in the generated planet, the key ingredient is to use the elevation input as a basis for computing river flow.

We first need a means of converting the discretised low-resolution elevation globe sketch into a continuous high-resolution heightmap atlas suitable for the river network extraction procedure. Recall that the elevation globe sketch has 5 altitude bands. After upsampling the sketch, we mark out the contours between pixels from different bands. Next, we assign altitudes to pixels based on an interpolation of values from the two closest distinct contours. Lastly, noise is added to prevent overly straight river trajectories. The different stages of the pipeline are depicted in Figure 6.

We then run a flow simulation using the PySheds library to derive the upstream area for every atlas pixel. The upstream area, otherwise known in geomorphology as the drainage area, can be combined with precipitation to provide a location-based measure of the volume of water in the river, which we term the *accumulation*

The simulation stage results in a high-resolution river network encoded in a quadsphere atlas at 2.4 km per pixel, and annotated with accumulation values for each river pixel.

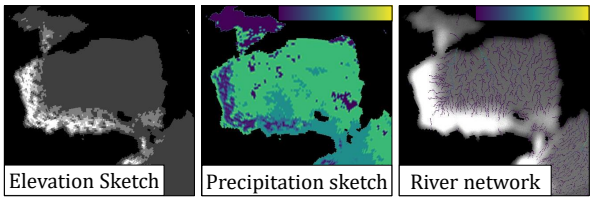

Figure 6: *Precipitation and smoothed elevation sketches are combined and used by a river simulation to derive river networks.*

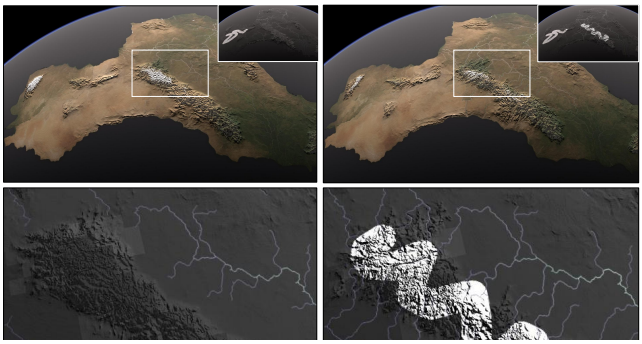

Figure 7: *The impact of the user-provided precipitation sketch on the final result. The derived river network contains larger rivers after painting a region with high precipitation (bottom right).*

Finer-grained edits: While the simulation stage results in a dense river network, consistent not only with elevation, but also with landcover and temperature if they were used for establishing precipitation, users may still want to emphasise or remove some of the computed rivers. Allowing such fine-grained control is not unrealistic since micro-climates can give rise to highly local precipitation.

Rather than iteratively editing the precipitation globe sketch, which would require locally re-simulating river flow, we instead allow users to directly scale or clamp the accumulation values along the simulated rivers, without altering their route. In support, users can view, and zoom in on, water accumulation values on the globe (adequately scaled to make them visible) before making adjustments.

Conditioning: Finally, the resulting high-resolution river network atlas, incorporating the user-edited water accumulation values at each pixel, is used to condition the diffusion process that generates the planet.

In order to strengthen adherence to this river network we make use of classifier free guidance [HS22]. This is a well-established technique in text-to-image diffusion models, but we show that it also has value in our image-to-image context. The usual procedure is to run inference twice for every timestep, once with and once without the prompt. The difference between the results is scaled by a guidance factor and added back to the unconditioned output. In our case, instead of dropping a prompt, we drop the river network atlas, with the expectation that the difference image will encode and enable enhancement of the river network contribution. The caveat is that this relies on a coherent, well-formed river network.

6. Diffusion Model

We train an image conditioned diffusion model to generate 256×256 satellite images and DEM tiles from Earth and Mars. The input to this model consists of extracted landcover, temperature and elevation sketches at a resolution of 78.3 km per pixel (8×8 input tiles resized to 256×256) as well as a river network matching the output resolution of 2.4 km per pixel (256×256 input tiles). This model is then applied to generate full planets by using a variant of DiffInfinite [ANH*23] tile blending.

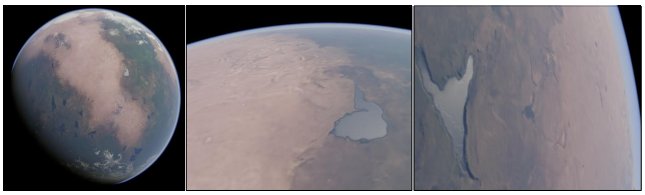

Figure 8: *Combining Earth and Mars landcover in a single sketch allows to create a combined planet with seamless blending.*

6.1. Data sources and preparation

The primary training data is derived from five datasets (see Table 5): satellite, elevation, temperature, landcover, and river upstream area, each at a spatial resolution of 2.4×2.4 km per pixel.

For satellite images, we settled on the months of June and December as being the most visually distinct. Our data is sourced from the BMNG satellite image dataset, which is only available for 2004, because it provides complete globally processed images that can be uniformly sampled with ease. Their preprocessing includes cloud removal, water shading and correction in regions with low contrast (salt pans) or high light absorption (rainforests). The visualisation focus of this data aligns with our goal of providing high visual fidelity. These are all stored as high resolution images using an equiangular WGS84 projection (EPSG:4326). All data is normalised to an 8-bit integer range, except river upstream area which is stored as 32-bit floating point values. Despite our best efforts, it was, unfortunately, not possible to source all data with sufficient resolution from the same year. In the case of landcover we chose ESA WorldCover, which despite only being available from 2020 onward has a small and manageable number of 11 distinct landcover classes. We then performed additional pre-processing to reassign underrepresented classes, such as moss and lichens, herbaceous wetland, and mangroves.

We also sourced secondary training data in the form of satellite image and elevation datasets for Mars. This presents a complication because the temperature and landcover layers required for integrating with Earth data are not included. However, Mars has distinct soil colouring in different regions, depending on the mineral and frozen water and dry ice composition, including white, red, brown, gold, and tan. Since these zone colours do not map to readily understandable landcover classes, we chose to treat them instead as temperature bands. Then, in terms of landcover all Mars data is assigned to a single unique class. By reserving this separate landcover class, we teach the model to only generate Mars elevation and satellite data when this class is present in the input landcover sketch.

The images in our dataset have a resolution of 16384×8192, but we perform training and inference on 256×256 pixel image tiles, as this provides the best balance between data quantity and inference speed. Before extracting tiles for training we need to overcome the atlas projection issue, which can otherwise cause stretching and misalignment. For instance, the WGS84 projection significantly overrepresents Antarctica, which will lead to inference bias.

We choose to project our atlases onto a quadrilateralised spherical cube (quadsphere) representation to avoid just such shortcomings. It provides a minimally distorted equal area projection with connected edges. The quadsphere itself is stored as six faces in a cube configuration and there are thus three main cases to implement when sampling a tile. If a tile lies completely within a single face, we simply crop the tile directly from the face. If a tile spans two faces, we need to align their edges by calculating the correct index rotation. In the case where a tile covers three faces, there is no simple way to derive contiguous indices. We resolve this by minimally shifting the tile position so that it becomes an instance of the second case (spanning two faces). In this way we avoid sampling tiles that overlap all three faces. This sometimes leads to small line artefacts at the 8 cube corners, but it is a necessary price to pay for the benefits of the quadsphere.

We randomly extract patches from the quadsphere, as is common in patch based diffusion models [WJZ*23], without the requirement of informing the model of patch positions. This results in training tiles overlapping arbitrarily, which we reason is no worse than running multiple epochs on the same images. Unfortunately, the dataset is still sparse with less than 2000 completely unique tiles. In order to provide sufficient data to the model, we rely on data augmentation. We perform combinations of random horizontal reflections and rotation about the tile center in the range -15 to 15 degrees, which is suitable for top-down satellite imagery and elevation. Vertical reflection is reserved for validation and testing because splitting the data set by other means is non-trivial in the presence of overlaps.

The learning is conditioned on discretised low-resolution tiles that match the structure of the user's globe sketches. These are extracted from the corresponding fine-resolution data by a process of downsampling and quantisation. To downsample tiles we employ either bilinear or majority resampling for non-discrete and discrete data, respectively. To quantise elevation and temperature tiles, we simply assign values to the relevant bands. One subtlety is that we apply a small random offset in the range $(-3.2, 3.2)^\circ C$ to temperature before quantisation. This regularisation prevents overfitting.

6.2. Training

Our diffusion model uses low-resolution tiles from the landcover, temperature, and elevation sketches and a high-resolution tile from the derived river network as inputs and generates high-resolution satellite image and elevation tiles as outputs. The model itself is a denoising diffusion probabilistic model with the learning objective of predicting the total noise added to an image at each step. The input sketches are simply concatenated onto the noisy outputs during

training to provide the model with spatial and semantic conditioning.

The model was trained using L2 loss, as is standard. We experimented with a weighted interpolation between L1 and LPIPS perceptual loss based on the remaining timesteps, but found that this introduced additional noise. We undertook random hyperparameter sweeps, optimising on the quality of generated samples as the basis for our choice of hyperparameters, as detailed in Table 6.

6.3. Inference

We have established that generating and blending many smaller tiles to form a larger planetary whole is the only practical way to overcome the irregular structure of a low-distortion connected sphere representation. What remains to be decided is the best choice of blending procedure. Even with conditioning, adjacent tiles are often too disparate for traditional merging with graph cuts or alpha blending. We also experimented with outpainting based on partially masked images but found that the unmasked context was only respected near masked boundaries, leading to highly salient gridding artefacts. We finally settled on a parallelizable hierarchical diffusion process, inspired by DiffInfinite [ANH*23]. This algorithm denoises randomly positioned tiles on each timestep within the diffusion process, iteratively and simultaneously converging on a fully denoised and fully covered image. We further improve on the original by introducing irregular tile boundaries for the final 10% of timesteps to prevent any possible straight lines in the output.

The irregular tile boundaries are derived by eroding the edges of the square tile masks using low frequency simplex noise, resulting in a mask with curved edges. This, unfortunately, comes at a cost of around 20% extra inference time, but significantly reduces the line artefacts otherwise visible with a low number of timesteps (see Figure 17). We also achieve quantitative improvement using this method at lower timesteps, as discussed in Section 7. For higher timesteps, this feature can be disabled for faster inference as using more steps mitigates the line artefacts.

To further improve memory and computation efficiency, we implement bounding box inference, by separating landmasses in the sketch into individual components, placing them within a rectangular bounding box and generating them in isolation. The results are then inserted back into the globe at the identified position. This is only really helpful for planets with large oceans and distinct landmasses, but can, in such cases, greatly reduce overheads.

7. Results

Our method was implemented using PyTorch and executed on several different GPUs based on cluster scheduling. These include an NVIDIA H100, Quadro RTX 6000, Quadro RTX 8000, and a personal computer with an RTX 3070. All planets shown in the paper were generated using our model and then rendered off-line in Blender. The typical run-time for generating an entire Earth-scale planet with 100 diffusion iterations (timesteps) is around 1.5-5 hours on a single 20GB NIVIDA H100 partition, depending on the relative proportion of land and ocean as well as the river network density. This density is the proportion of tiles that contain a river and we find this to be around 0.6 on average for earth-like planets. The coarse satellite preview bypasses diffusion and is almost instantaneous, while a diffusion preview at 10 iterations requires $10-30$ minutes to compute. We can estimate the time a planet will take to generate based on the required number of tiles (N):

GPU	Best	Average	Worst
GeForce RTX 3070	11.66	7.29	5.83
Quadro RTX 8000	14.84	9.28	7.42
Quadro RTX 6000	16.44	10.27	8.22
H100	47.64	29.77	23.82

Table 1: *Tile generation rates (R) for different NVIDIA GPUs at a batch size of* 8*. The best case is when there are no rivers or Classifier Free Guidance is disabled. The average case has a river density of* 0.6 *and the worst case has a river density of* 1.0*, both with Classifier Free Guidance enabled.*

$$N = \left(\frac{6 \cdot w_f^2}{w_t^2}\right) \cdot P_L \cdot T \cdot \eta^{-1} = \left(\frac{6 \cdot 5216^2}{256^2}\right) \cdot P_L \cdot T \cdot 0.45^{-1} \quad (1)$$
$$\approx 5535.21 \cdot P_L \cdot T$$

In Equation 1, N is a function of the width of each quadsphere face (w_f), the width of each tile (w_t) adjusted by land proportion (P_L) and the number of diffusion timesteps (T). The variable η accounts for the efficiency of blending overlapping tiles, which we find to be 0.45 on average. Validating the equation on several examples with $T = 100$ gave an error margin of less than 1%. Using this, we can then approximate the total generation time ($\mathcal{T}$) using the number of tiles generated per second (R) from Table 1 as: $\mathcal{T} \approx N/R$.

7.1. Variety and Control

The figures in this paper show the combination of variety and control afforded by our method. Users have access to a brush toolbox to paint the various globe sketch inputs. Typically, an author conversant with the system can complete the sketching of a full planet in about 5-10 minutes.

We have included a number of examples of different use cases in Figures 9 to 10 and 13 to 11. The temperature sketch allows for a dramatic change of the planet climate, enabling an illustration of a planet locked in an ice age (Figure 9). An example of how the temperature sketch affects different landcovers is given in Figure 5. We note that the coarse temperature and landcover conditioning is intended to provide ease of use and control rather than highly-realistic climatic consistency. Authoring fantastical planets that mix Mars and Earth features is also possible and the blend between planetary features is surprisingly seamless (Figure 8). Our model is also capable of producing scientific reconstructions of Earth during the Palaeolithic period (Figure 10) when the continents had a very different configuration. Finally, Figure 11 further demonstrates the variety of achievable planetary landscapes, ranging from reddish rock planets to water and ice worlds.

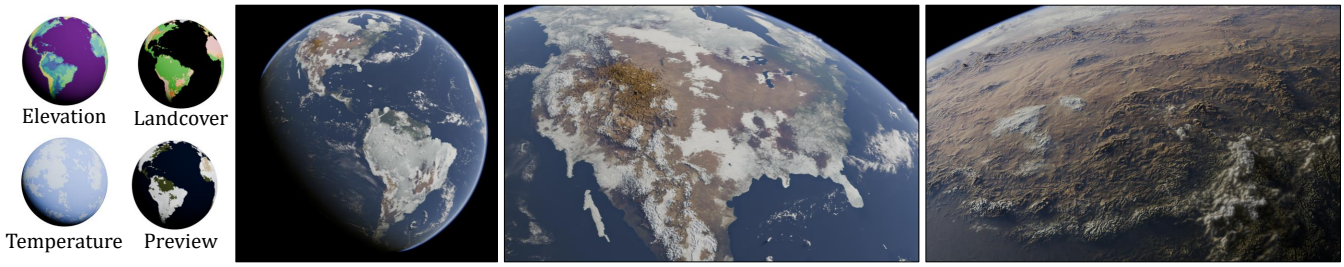

Figure 9: *Frozen earth. A cold temperature sketch is applied to invoke a new ice age.*

	$P_L = 0.3$			$P_L = 0.5$			$P_L = 1.0$		
GPU	$T = 10$	$T = 25$	$T = 100$	$T = 10$	$T = 25$	$T = 100$	$T = 10$	$T = 25$	$T = 100$
GeForce RTX 3070	45m	1.5h	6.25h	1h	2.75h	10.5h	2h	5.25h	21h
Quadro RTX 8000	30m	1.25h	5h	45m	2h	8.25h	1.75h	4.25h	16.5h
Quadro RTX 6000	30m	1h	4.5h	45m	1.75h	7.5h	1.5h	3.75h	15h
H100	9m	30m	1.5h	15m	45m	2.5h	30m	1.25h	5.25h

Table 2: *Approximate generation times on different GPUs for planets with various land proportions (P_L). Earth has $P_L \approx 0.3$. The average case tile rate (R) is used to estimate full generation times.*

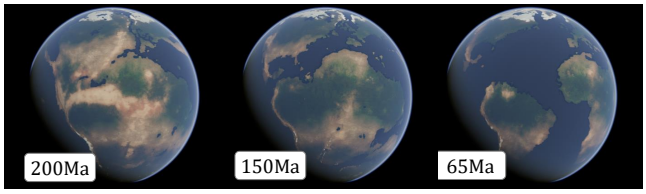

Figure 10: *Three different Palaeolithic versions of the Earth from 200, 150 and 65 million years ago.*

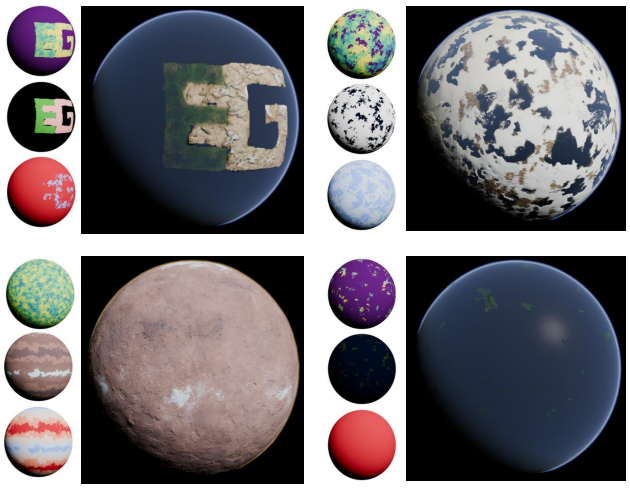

Figure 11: *Diverse examples: EG-logo, frozen, mars-like, and island planets. On the left of each are (from top to bottom) the elevation, landcover, and temperature globe sketches.*

7.2. Validation

We run three forms of inference for evaluation purposes. The first (*tiles*) involves generating individual tiles at a 256×256 pixel resolution to judge the best case quality of the model against previous single tile approaches. The second (*blended*) generates larger, 1024 blended patches to test the impact of blending. The last (*full*) synthesises fully-formed planets for visual inspection and rendering. The *tiles* and *blended* validations are based on vertically flipped automatically-derived sketches and matching outputs from the test set, while the full planets are primarily generated from user sketches. Figure 13 shows a full planet generated from these flipped sketches.

We use two sets of standard machine learning metrics for evaluation, namely: the distribution-based Fréchet Inception Distance (FID) and the visual-based Learned Perceptual Image Patch Similarity (LPIPS). LPIPS is often used to evaluate diversity by comparing random pairs of generated and real samples, but it can measure quality by comparing corresponding pairs, with lower scores being better.

For our third metric we forego a conventional distance measure because it is invalidated by the uneven ranges of the sketch bands. Instead, we formulate a sketch loss, by taking the output and "sketchifying" it through downsampling and quantisation (as outlined in Section 6.2). By counting the proportion of exact matches between the original input and derived output sketches we obtain an adherence score. For the satellite preview and output image a simple RMSE of the colour difference suffices. This is preferred to LPIPS because the preview is pixelated and has very different perceptual characteristics.

For individual tiles, we are primarily concerned with quality as a function of timesteps for many samples. Accordingly, we plot FID, LPIPS and sketch loss scores versus Timesteps for individual

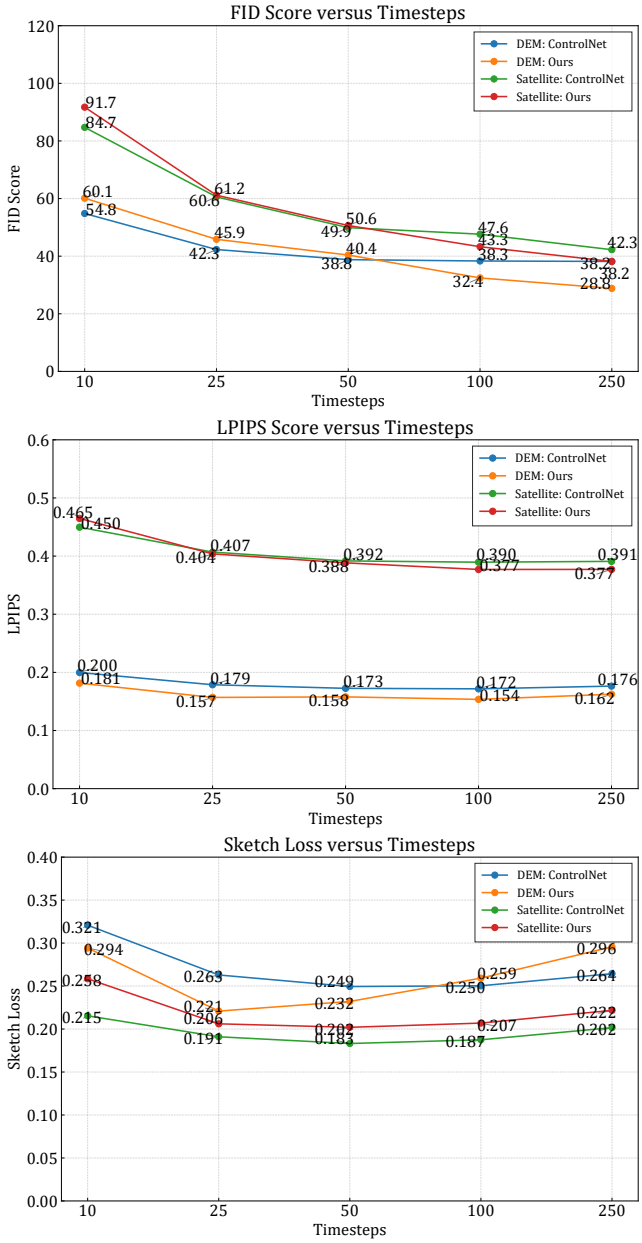

Figure 12: *Impact of the number of timesteps on FID, LPIPS, and Sketch loss for individual tiles generated with our method and a ControlNet baseline.*

tile inference in Figure 12. We also discuss the impact of the river guidance factor on general quality, although the results are less pertinent.

For blended outputs, where grid artefacts have a significant impact on the rendered planet, we evaluate the difference in FID scores between using regular and irregular tile blending and also compare them against individual tiles.

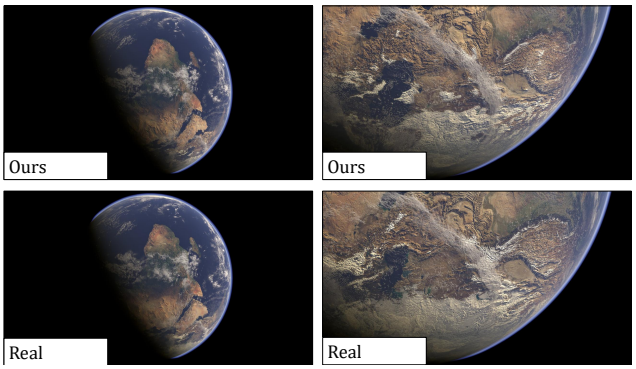

Figure 13: *Flipped Earth. A reflection of the source earth data is compared against results generated by our model from reflected sketches. While our overall landcover and elevation match the real data, the model generates different, yet consistent fine detail.*

7.3. Discussion

Timesteps: Figure 12 shows graphs of the numerical results we obtain for the evaluation of FID, LPIPS and Sketch Loss.

We note that for individual tiles FID follow a logarithmic-like decrease with the number of timesteps, demonstrating sharper and more refined detail, but with diminishing returns. This encourages the use of smaller timesteps to trade off generation time and quality.

LPIPS can be used to assess both quality and diversity. However, as seen in Figure 14, the only meaningful change occurs between 10 and 25 timesteps, because this is the point at which a perceptual improvement in detail occurs. We show that our model is able to produce diverse outputs with an average random pair LPIPS score of 0.734 for satellite tiles and 0.427 for DEM tiles, compared to the average pairwise scores of 0.402 and 0.162, respectively.

Sketch loss follows an interesting pattern: it starts high at 10 timesteps due to a lack of inference quality, drops to a minimum at 25 timesteps because fine divergent details have not yet been introduced, and then increases gradually as more diverse detail is introduced.

Figure 17 illustrates how the quality of a full planet typically improves for different timesteps. We observe that at 5 timesteps, there are many gridded artefacts and the landcover does not align well with the benchmark of 100 timesteps. At 10 timesteps the alignment improves more for irregular than regular tile boundaries, making the former setting more suitable for an intermediate preview.

ControlNet comparison: To benchmark our model against ControlNet [ZRA23], we train and evaluate a ControlNet adapter. We first convert our existing trained model to an unconditional model by slicing away the weights of the input convolutional layer to remove the last 4 channels that process the input sketch. We then freeze this model and train ControlNet to inject features extracted from the sketch into the UNet. With this configuration, ControlNet produces similar FID and LPIPs scores to our method (see Figure 12). It does, however, provide slightly better sketch loss for the satellite images and better DEM sketch loss at higher timesteps.

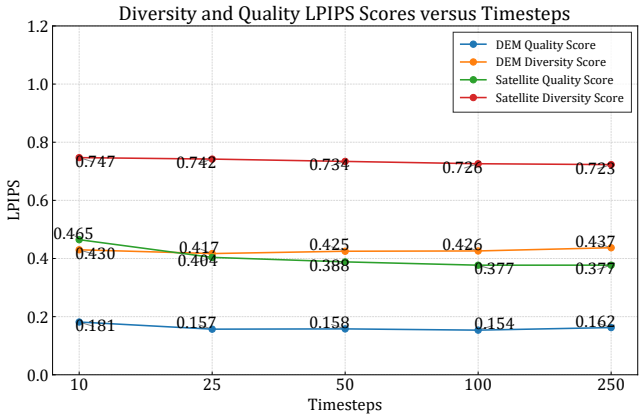

Figure 14: *Impact of the number of timesteps on LPIPS quality and diversity scores.*

Nevertheless, this DEM control comes at the cost of distribution accuracy as seen in the lack of improvement in ControlNet DEM FID scores at timesteps 100 and 250. ControlNet also incurs higher computation and memory costs, since it requires an extra pass through the copied encoder, with associated additional storage. Table 3 demonstrates that our model is significantly faster than ControlNet with a $\approx 40\%$ increase in speed for the model in isolation and a $\approx 26\%$ increase when considering the full overhead of generating and blending a planet. We also see a $\approx 24\%$ reduction in memory for a batch size of 8, but this reduction is only due to the extra 1.22GB of memory from the ControlNet weights, so scaling the batch size up will render this proportionally less significant. Due to the structure of our coarse input sketches, ControlNet does not provide markedly better quality and comes with an increase in generation time. On this basis we recommend our concatenation conditioning in preference to ControlNet.

Type	CFG	Method	Speed (Tiles/s)	VRAM (GB)	Δ Speed
Blended	Off	ControlNet	12.72	5.63	–
Blended	Off	**Ours**	16.02	4.43	+26.0%
Blended	On	ControlNet	6.36	5.63	–
Blended	On	**Ours**	8.02	4.43	+26.1%
Tiles	Off	ControlNet	17.49	6.00	–
Tiles	Off	**Ours**	24.43	4.43	+39.7%
Tiles	On	ControlNet	8.75	6.00	–
Tiles	On	**Ours**	12.22	4.43	+39.7%

Table 3: *Speed and memory usage for our method versus a ControlNet baseline on an NVIDIA Quadro RTX 6000 of batch size* 8.

Rivers: We found that FID and LPIPS scores were little influenced by the river guidance factor. This is likely because most rivers fall below the scale of a single pixel even at the highest resolution. However, the sketch loss increased more noticeably with an increased guidance factor. This can be explained by the boolean nature of this metric. Any change in landcover class or elevation band introduced by the river is not reflected in the original sketches and will increase the sketch loss.

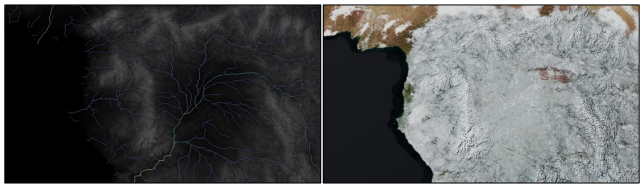

Figure 15: *An example of long-distance river adherence in the DEM and satellite image. We shade the river in the DEM to show how it shapes the surrounding terrain.*

In Figure 15 we demonstrate that our model is able to sustain rivers that span many tiles and flow consistently to shape the surrounding terrain, thereby improving the structural realism of the landscape. In Figure 16 we see the effects of Classifier Free Guidance (CFG) on river adherence and consistency. While ControlNet does provide better river adherence, the continuity and visible impact of rivers are greatly improved by river CFG.

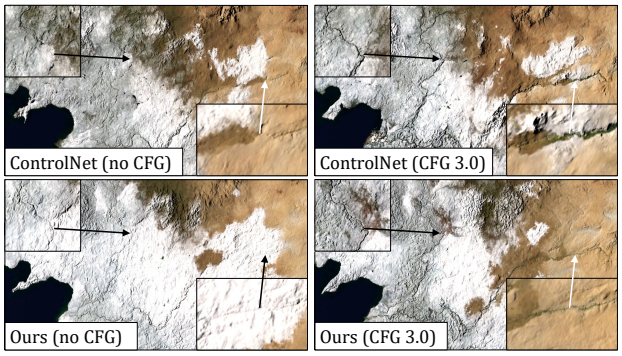

Figure 16: *A qualitative comparison of rivers generated with ControlNet and our method, both with and without Classifier Free Guidance (CFG).*

Tile blending. Our blending architecture departs from the DiffInfinite template in its use of irregular tile boundaries and it is worth assessing the impact of this choice. Figure 17 demonstrates how the use of irregular tiles acts to reduce visual artifacts at low timesteps, while Figure 18 provides an ablation based on the FID score. Irregular outperforms regular for low numbers of timesteps (10 and 25) but they eventually converge. As is to be expected the baseline of individual tiles outperforms both, because this is the task on which the model is specifically trained. Interestingly, individual tiles are marginally worse than blending at 10 timesteps. We hypothesise that the blending model gains better context and consistency as it iterates in overlapping areas many times per step rather than just once. Individual tiles catch up on this repetition at higher timesteps and the training specificity comes to dominate.

Intermediate scale models: While curtailing the number of timesteps to 10 or 25 does provide an initial approximation, this cannot be used as a checkpoint and the inference needs to be

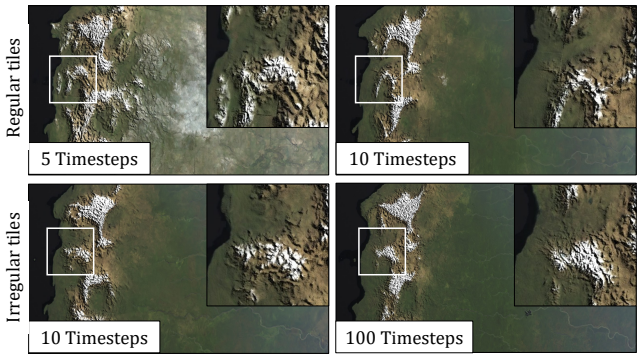

Figure 17: *Visual impact of the number of timesteps and use of irregular tiles on the final result.*

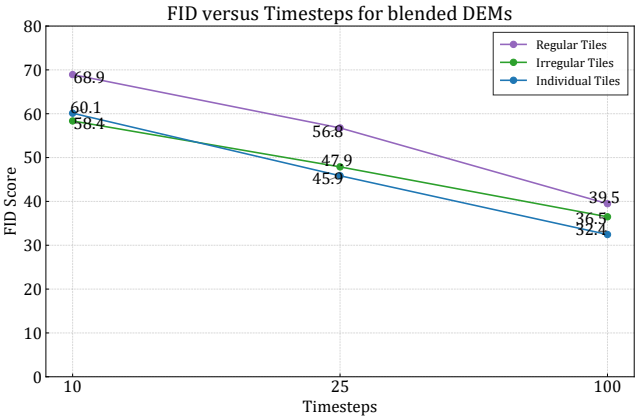

Figure 18: *Comparing the impact of the number of timesteps on FID for individual tiles, blended regular, and blended irregular tiles.*

restarted afresh for a full run. An appealing alternative is to train a pair of models: the first to generate a coarse resolution atlas (at say 19.6km/pixel) and the second to upscale this to the final resolution. The complication is that this significantly decreases the training corpus. We tried reducing the tile size to 64×64 pixels but overfitting was still present. Our intuition is that the convolution input size for the diffusion model unet remained unchanged, so that, even though the number of tiles increased, the number of unique convolution patches did not. One way of addressing this is to generate many diverse planets using our current architecture and then train the coarse resolution model on these outputs once downsampled, but we leave this to future work.

Previous quantitative results: in Table 4, we provide a comparison against reported FID and LPIPS for published DEM and satellite image generators. Our approach improves on previous work on all metrics except satellite FID, where DiffusionSat [KLZ*24] outperforms our method.

Previous qualitative results: Compared to previous planet generation techniques, such as [CPGG20, DGGK11, CPGG19], our method is the only one able to generate both realistic and diverse

	Satellite		DEM	
Technique	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
[GDG*17]	–	–	≈ 70	≈ 0.215
[JSR22]	–	–	54.44	–
[LGP*23]	–	–	≈ 40	≈ 0.195
[KEK24]	–	0.440	–	–
[KLZ*24]	**15.8**	0.622	–	–
Ours	43.3	**0.377**	**32.4**	**0.154**

Table 4: *A quantitative comparison against previous models reporting LPIPS and FID scores. For diffusion techniques, we report results at 100 timesteps. Arrows indicate that lower values are better.*

texturing and terrain elevation. We provide greater user control than these methods through a multi-layer sketching interface to directly specify the generated biomes and elevation, and indirectly control the size and flow of river networks through precipitation and elevation sketches. We are the first to ground the realism and diversity in real data, with global consistency introduced through a simulated river network. Unfortunately, these advantages currently come at the cost of relatively low precision and much slower generation times. We also lack global geological formation processes, such as plate tectonics [CPGG19], but our system would be able to reverse engineer sketches from DEMs generated with simulation methods, thereby providing our model with accurate global geological features.

7.4. Limitations

We have demonstrated the capability of our method to generate diverse, high-quality, and controllable large-scale planets. However, some limitations remain. First, we do not explicitly enforce geological consistency. The diffusion model captures plausible geomorphological patterns *locally*, but, with the exception of river networks, lacks global information on tectonics or erosion informed by the climatic history of the planet, which may result in inconsistent large-scale structures. Second, since we focused primarily on the land surface, our representation of the oceans is binary; adding bathymetric information or coastal processes would provide more diversity in water shading near the continents. Third, we used a simple procedural model for cloud coverage, which does not reflect meteorological patterns consistent with landcover and climate. Fourth, we selected a $32\times$ upsampling factor from the coarse input sketches as a balance between ease of authoring and visual quality. As such, blocky edges can occasionally appear in coastlines and continental boundaries (see Figure 15). Fifth, the river derivation process does not guarantee hydrological termination in lakes or oceans, since users are not prevented from producing elevation sketches with endorheic basins. Finally, while our training inputs are derived from real-world climate, landcover and elevation data, our system is intended for artistic use, so global climatic and geological realism is relaxed in favor of controllability and simplicity.

8. Conclusion

In this paper, we have presented a method for the creation of terrestrial planets. Our approach tackles three main challenges of planet authoring. First, we enable user control at the planetary scale through coarse proxy sketches provided by the user, which inform the model with variations of elevation, landcover, temperature, and precipitation for rivers. Second, we infer elevation and satellite imagery at a resolution suitable for exosphere-level flyovers. To this end we employ a diffusion model to generate and blend tiles located in a quadsphere representation that minimises atlas to globe distortion. Finally, we enforce a degree of global geomorphological consistency through explicit river conditioning based on the computation of a river network annotated with water accumulation values. Our results demonstrate that the method supports efficient authoring, visual diversity, and consistent large-scale structure across a variety of planetary configurations.

Future work will address the outstanding limitations of the method. We see strong potential in coupling our data-driven approach with physics-based simulation. For instance, simulating plate tectonics and erosion could enhance large-scale consistency and reveal the temporal evolution of the planet over geological timescales. Similarly, integrating a weather model combined with timestamped cloud coverage from satellite data could produce a globally consistent cloudscape – with the subsequent challenge of generating high-resolution animated textures. Finally, we aim to further increase the diversity of possible planets. Achieving this will require synthetic data augmentation, as planets increasingly distant from Earth suffer from lower-quality satellite imagery.

Acknowlegments

This work is funded by the project EOLE ANR-23-CE56-0008, supported by Agence Nationale de la Recherche (France) to Guérin.

This project was sponsored by the Agence Nationale de la Recherche project Invterra ANR-22-CE33-0012-01 to Cordonnier.

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

Channel	Source	Year
Satellite image	NASA BMNG	06 & 12, / 2004
Elevation	ETOPO1: Global 1 Arc-Minute Elevation	2008
Temperature	ERA5-Land Monthly Aggregated	06 & 12, 2004
Landcover	ESA WorldCover 10m v100	2020
Precipitation	ECMWF / C3S ERA5 Monthly Aggregated	06 & 12, / 2004
Upstream area	WWF HydroSHEDS Free Flowing Rivers Network v1	2000
Mars satellite	USGS Astrogeology Science Center	1975-1980
Mars elevation	USGS Astrogeology Science Center	1997-2001

Table 5: *Data sources*

Hyperparameter	**Searched Values**	**Chosen Value**
Learning Rate (LR)	$\{1\times10^{-4}, 5\times10^{-4}, 1\times10^{-5}\}$	1×10^{-4}
Batch Size	–	16
UNet Layers per Block	$\{2,3,4\}$	3
UNet Base Dimension	$\{64,128\}$	128
UNet Channel Multipliers	$\{[1,2,4,8],[1,1,2,2,4,4]\}$	$[1,2,4,8]$
UNet Attention Head Dim.	$\{4,8,16,32\}$	8
Noise Scheduler	–	DDIM
Optimizer	–	AdamW
Loss function	{Weighted perceptual, L1, L2}	L2
Eta (η)	–	1.0
Beta Schedule	{linear, cosine}	linear
Beta Start	$\{1\times10^{-3}, 1\times10^{-4}, 1\times10^{-5}\}$	1×10^{-4}
Beta End	$\{2\times10^{-1}, 2\times10^{-2}, 2\times10^{-3}\}$	2×10^{-2}
Clip Sample	{True, False}	True
Clip Gradient Norm	{True, False}	True
Weight Decay	$\{0.0, 1\times10^{-5}, 1\times10^{-2}\}$	1×10^{-2}
Learning Rate Scheduler	{cosine, plateau}	plateau
Gradient Accum. Steps	$\{1,16\}$	1
Use EMA	{True, False}	False

Table 6: *Model hyperparameters: ranges searched and selected values.*

References

[ACA18] ARGUDO O., CHICA A., ANDUJAR C.: Terrain super-resolution through aerial imagery and fully convolutional networks. *Computer Graphics Forum 37*, 2 (2018), 101–110. 3

[ANH*23] AVERSA M., NOBIS G., HÄGELE M., STANDVOSS K., CHIRICA M., MURRAY-SMITH R., ALAA A., RUFF L., IVANOVA D., SAMEK W., KLAUSCHEN F., SANGUINETTI B., OALA L.: Diffinfinite: Large mask-image synthesis via parallel random patch diffusion in histopathology, 2023. arXiv:2306.13384. 4, 5, 8, 9

[BTYLD23] BAR-TAL O., YARIV L., LIPMAN Y., DEKEL T.: Multidiffusion: fusing diffusion paths for controlled image generation. In *Proceedings of the 40th International Conference on Machine Learning* (2023), ICML'23, JMLR.org. 4

[BW06] BOKELOH M., WAND M.: Hardware accelerated multi-resolution geometry synthesis. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2006), I3D '06, Association for Computing Machinery, p. 191–198. 4

[CMR*25] CZERKAWSKI M., MARTIN R., ROUFFET R., ET AL.: Mesa: Text-driven terrain generation using latent diffusion and global copernicus data. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 3067–3075. 3

[CO75] CHAN F., O'NEILL E.: *Feasibility Study of a Quadrilateralized Spherical Cube Earth Data Base.* Tech. Rep. CSCTR756007, Defense Technical Information Center, 1975. 2

[CPGG19] CORTIAL Y., PEYTAVIE A., GALIN E., GUÉRIN E.: Procedural tectonic planets. *Computer Graphics Forum 38* (5 2019), 1–11. 2, 4, 13

[CPGG20] CORTIAL Y., PEYTAVIE A., GALIN É., GUÉRIN É.: Real-time hyper-amplification of planets. *The Visual Computer 36*, 10-12 (2020), 2273–2284. 4, 13

[CVG*15] CRUZ L., VELHO L., GALIN E., PEYTAVIE A., GUÉRIN E.: Patch-based terrain synthesis. In *International Conference on Computer Graphics Theory and Applications* (2015), pp. 6–pages. 3

[CXY*22] CAI X., XI M., YU N., YANG Z., SUN H.: A terrain elevation map generation method based on self-attention mechanism and multifeature sketch. *Computational Intelligence and Neuroscience 2022*, 1 (2022), 9481445. 3

[DGGK11] DERZAPF E., GANSTER B., GUTHE M., KLEIN R.: River networks for instant procedural planets. *Computer Graphics Forum 30* (2011), 2031–2040. 4, 13

[FAW19] FRÜHSTÜCK A., ALHASHIM I., WONKA P.: Tilegan: synthesis of large-scale non-homogeneous textures. *ACM transactions on graphics 38*, 4 (2019), 1–11. 3

[FFC82] FOURNIER A., FUSSELL D., CARPENTER L.: Computer rendering of stochastic models. *Communications of the ACM 25*, 6 (1982), 371–384. 4

[GDG*17] GUÉRIN E., DIGNE J., GALIN E., PEYTAVIE A., WOLF C., BENES B., MARTINEZ B.: Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Trans. Graph. 36*, 6 (nov 2017). 2, 3, 13

[GGP*19] GALIN E., GUÉRIN E., PEYTAVIE A., CORDONNIER G., CANI M.-P., BENES B., GAIN J.: A review of digital terrain modeling. *Computer Graphics Forum 38*, 2 (2019), 553–577. 3

[HHM*24] HU Z., HU K., MO C., PAN L., WANG Z.: Terrain diffusion network: Climatic-aware terrain generation with geological sketch guidance. *Proceedings of the AAAI Conference on Artificial Intelligence 38*, 11 (2024), 12565–12573. 3

[HJA20] HO J., JAIN A., ABBEEL P.: Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems 33* (2020), 6840–6851. 3

[HS22] HO J., SALIMANS T.: Classifier-free diffusion guidance, 2022. arXiv:2207.12598. 7

[JHKK25] JEONG J., HAN S., KIM J., KIM S. J.: Latent space super-resolution for higher-resolution image generation with diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 2355–2365. 3

[JSR22] JAIN A., SHARMA A., RAJAN K.: Adaptive & multi-resolution procedural infinite terrain generation with diffusion models and perlin noise. In *Proceedings of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing* (2022), pp. 1–9. 3, 13

[JSR24] JAIN A., SHARMA A., RAJAN K.: Learning based infinite terrain generation with level of detailing. In *2024 International Conference on 3D Vision (3DV)* (2024), IEEE, pp. 1048–1058. 3

[KEK24] KANAI T., ENDO Y., KANAMORI Y.: Seasonal terrain texture synthesis via köppen periodic conditioning. *The Visual Computer 40*, 7 (2024), 4857–4868. 3, 13

[KLZ*24] KHANNA S., LIU P., ZHOU L., MENG C., ROMBACH R., BURKE M., LOBELL D. B., ERMON S.: Diffusionsat: A generative foundation model for satellite imagery. In *The Twelfth International Conference on Learning Representations* (2024). 3, 13

[KSR20] KUBADE A. A., SHARMA A., RAJAN K. S.: Feedback neural network based super-resolution of DEM for generating high fidelity features. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium* (2020), pp. 1671–1674. 3

[LCL*22] LIN C. H., CHENG Y.-C., LEE H.-Y., TULYAKOV S., YANG M.-H.: InfinityGAN: Towards infinite-pixel image synthesis. In *International Conference on Learning Representations* (2022). 3

[LGP*23] LOCHNER J., GAIN J., PERCHE S., PEYTAVIE A., GALIN E., GUÉRIN E.: Interactive authoring of terrain using diffusion models. *Computer Graphics Forum 42*, 7 (2023), e14941. 2, 3, 4, 13

[LLXT22] LI S., LI K., XIONG L., TANG G.: Generating terrain data for geomorphological analysis by integrating topographical features and conditional generative adversarial networks. *Remote Sensing 14*, 5 (2022), 1166. 3

[MF25] MADAR O., FRIED O.: Tiled diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 7795–7804. 3

[ND21] NICHOL A. Q., DHARIWAL P.: Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning* (2021), PMLR, pp. 8162–8171. 3

[NJSR22] NAIK S., JAIN A., SHARMA A., RAJAN K.: Deep generative framework for interactive 3d terrain authoring and manipulation. In *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium* (2022), IEEE, pp. 6410–6413. 3

[Pla] PLANETSIDE SOFTWARE: Terragen 4. 4

[PPB*23] PERCHE S., PEYTAVIE A., BENES B., GALIN E., GUÉRIN E.: Authoring terrains with spatialised style. *Computer Graphics Forum 42*, 7 (2023), e14936. 3

[Rah18] RAHMON G.: *Evaluation of procedurally generated terrains via artificial and convolutional neural networks*. Master's thesis, Fen Bilimleri Enstitüsü, 2018. 3

[SD22] SCOTT J. J., DODGSON N. A.: Evaluating realism in example-based terrain synthesis. *ACM Trans. Appl. Percept. 19*, 3 (sep 2022). 7

[SH22] SALIMANS T., HO J.: Progressive distillation for fast sampling of diffusion models, 2022. arXiv:2202.00512. 3

[SME20] SONG J., MENG C., ERMON S.: Denoising Diffusion Implicit Models. *arXiv e-prints* (Oct. 2020), arXiv:2010.02502. arXiv:2010.02502. 3

[SW19] SPICK R. J., WALKER J. A.: Realistic and textured terrain generation using gans. In *European Conference on Visual Media Production* (2019), pp. 1–10. 3

[VACO23] VOYNOV A., ABERMAN K., COHEN-OR D.: Sketch-guided text-to-image diffusion models. SIGGRAPH '23, Association for Computing Machinery. 3

[VRGZS20] VALENCIA-ROSADO L. O., GUZMAN-ZAVALETA Z. J., STAROSTENKO O.: Generation of synthetic elevation models and realistic surface images of river deltas and coastal terrains using cgans. *IEEE Access 9* (2020), 2975–2985. 3

[WDJ*24] WOLSKI K., DJEACOUMAR A., JAVANMARDI A., SEIDEL H.-P., THEOBALT C., CORDONNIER G., MYSZKOWSKI K., DRETTAKIS G., PAN X., LEIMKÜHLER T.: Learning images across scales using adversarial training. *ACM Trans. Graph. 43*, 4 (July 2024). 3

[WJZ*23] WANG Z., JIANG Y., ZHENG H., WANG P., HE P., WANG Z., CHEN W., ZHOU M.: Patch diffusion: Faster and more data-efficient training of diffusion models, 2023. arXiv:2304.12526. 8

[XKV22] XIAO Z., KREIS K., VAHDAT A.: Tackling the generative learning trilemma with denoising diffusion gans, 2022. arXiv:2112.07804. 3

[YJH*24] YANG Z., JIANG H., HONG W., TENG J., ZHENG W., DONG Y., DING M., TANG J.: Inf-dit: Upsampling any-resolution image with memory-efficient diffusion transformer. In *European Conference on Computer Vision* (2024), Springer, pp. 141–156. 3

[ZHL*25] ZHANG J., HUANG Q., LIU J., GUO X., HUANG D.: Diffusion-4k: Ultra-high-resolution image synthesis with latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 23464–23473. 3

[ZLB*19] ZHAO Y., LIU H., BOROVIKOV I., BEIRAMI A., SANJABI M., ZAMAN K.: Multi-theme generative adversarial terrain amplification. *ACM Trans. Graph. 38*, 6 (nov 2019). 3

[ZLZ*22] ZHANG J., LI C., ZHOU P., WANG C., HE G., QIN H.: Authoring multi-style terrain with global-to-local control. *Graphical Models 119* (2022), 101122. 3

[ZRA23] ZHANG L., RAO A., AGRAWALA M.: Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision* (2023), pp. 3836–3847. 3, 11