Supplementary Material for GeoDiT: A Diffusion-based Vision-Language Model for Geospatial Understanding

1. Implementation Details of Remasking Strategy

The low-confidence remasking strategy is the core mechanism for iterative refinement in the non-autoregressive inference process of our model, GeoDiT. This strategy is designed to dynamically focus the model's generative capacity on the most uncertain parts of a prediction while preserving high-confidence tokens that have already been determined. This progressive refinement facilitates the generation of globally consistent and structured outputs, following effective practices established in discrete diffusion models.

The inference process commences with a template T_N composed entirely of `[M]` tokens and proceeds over N discrete timesteps. At each iteration k (from timestep t_k to t_{k-1}), the strategy is executed as follows:

1. **Full Sequence Prediction:** The model, p_θ, takes the masked sequence at the current step, T_{t_k}, and the visual condition vectors, C_v, as input. It produces a probability distribution over the entire vocabulary for each position in the sequence, $p_\theta(T_0|T_{t_k}, C_v)$. From this distribution, a provisional but complete sequence prediction, $\hat{T}_0$, is generated by selecting the token with the maximum probability (i.e., argmax) at each position.
2. **Confidence Score Acquisition:** For each predicted token $\hat{T}_0^i$ at position i in the provisional sequence $\hat{T}_0$, its confidence score is defined as the probability assigned to it by the model, $p_\theta(T_0^i = \hat{T}_0^i|T_{t_k}, C_v)$. This score directly reflects the model's certainty for the prediction at that specific position.
3. **Determining the Number of Tokens to Remask:** The number of tokens to be remasked at step k, denoted m_k, is determined by a predefined scheduling function $\gamma(t)$. This function maps the current timestep t_k (which anneals from 1 towards 0) to a ratio that dictates the percentage of the sequence to be masked. Specifically, $m_k = \lceil \gamma(t_k) \cdot L \rceil$, where L is the total sequence length. We employ a cosine schedule for $\gamma(t)$, which results in a higher masking ratio during the early stages of inference (when t_k is large) and a lower ratio in the later stages (when t_k is small). This facilitates a coarse-to-fine refinement process.
4. **Selecting and Applying the Mask:** Based on the confidence scores calculated in step 2, the model identifies the m_k tokens in $\hat{T}_0$ with the lowest scores. These are deemed the "low-confidence" tokens. The positions of these tokens are then reverted to the special `[M]` token, while all other high-confidence tokens from $\hat{T}_0$ are preserved. The resulting sequence, $T_{t_{k-1}}$, serves as the input for the subsequent iteration $k-1$.

This cyclic process of prediction and remasking allows the model to make judgments based on global information at every step, effectively avoiding the cascading errors that can occur in sequential, autoregressive generation. Our ablation study (see Table 4 in the main paper) confirms that this intelligent masking strategy is crucial for enhancing performance on tasks that demand high-precision details, such as bounding box coordinates and key object nouns, making it a key factor in achieving high-fidelity structured outputs.

2. Training Configurations

2.1. Stage I: Vision-Language Alignment

The initial vision-language alignment was conducted by training only the MLP projector, keeping the vision and language model backbones frozen[cite: 188]. The specific hyperparameters used, based on the provided training script, are detailed below.

- **Optimizer:** AdamW, with $\beta_1 = 0.9$ and $\beta_2 = 0.95$.
- **Learning Rate:** A peak learning rate of 1×10^{-3} was used.
- **Learning Rate Scheduler:** We employed a cosine decay schedule, with a warm-up phase over the first 3% of the total training steps.
- **Weight Decay:** 0.0
- **Training Epochs:** The alignment was performed for a single epoch.
- **Global Batch Size:**96. This was achieved with a per-device batch size of 16 and 1 gradient accumulation step, distributed across 6 GPUs.

- **Precision:** Training utilized BF16 and was accelerated with TF32 enabled.
- **Model Configuration:** The maximum model length was set to 8192 tokens. Gradient checkpointing was enabled to conserve memory. We used the 'sdpa' attention implementation.
- **Infrastructure:** The training was managed using DeepSpeed with a ZeRO Stage 3 configuration.

2.2. Stage II: Full Instruction Tuning

Following the initial alignment, the model underwent end-to-end fine-tuning on a large-scale, instruction-formatted remote sensing dataset. All major components of the model were unfrozen for this stage. The specific hyperparameters are detailed below.

- **Training Objective:** All primary model components (vision tower, MLP projector, and language model) were unfrozen and trained end-to-end. The model was initialized with the MLP projector weights obtained from Stage I.
- **Learning Rates:** A differential learning rate scheme was used. The main model components were trained with a peak learning rate of 1×10^{-5}, while the vision tower was fine-tuned with a lower learning rate of 2×10^{-6}.
- **LR Scheduler & Weight Decay:** Consistent with Stage I, a cosine decay scheduler with a 3% warmup ratio and a weight decay of 0.0 were used.
- **Global Batch Size:** 24. This was configured with a per-device batch size of 2, distributed across 6 GPUs with 2 gradient accumulation steps.
- **Training Epochs:** The model was fine-tuned for a single epoch on the instruction dataset.
- **Image Handling:** To process high-resolution imagery efficiently, we utilized an 'anyres' strategy. This involved variable grid pinpoints (from 1×1 up to 6×6) to create a maximum of 4 image patches per sample.
- **Infrastructure & Optimization:** The training setup leveraged DeepSpeed ZeRO Stage 3 and PyTorch's 'sdpa' attention implementation. For further optimization, the model was compiled using 'torch.compile' with the 'inductor' backend.
- **Precision & Model Configuration:** BF16 precision, TF32 acceleration, a maximum model length of 8192, and gradient checkpointing were enabled, consistent with the settings in Stage I.

3. Visualizations for the generation process of GeoDiT

To further substantiate the unique, non-autoregressive generation paradigm of GeoDiT, this section provides additional visualizations that complement the analysis presented in Figure 5 of the main paper. The examples below, featuring diverse geospatial scenes, demonstrate that the hierarchical, coarse-to-fine generation process is a consistent and

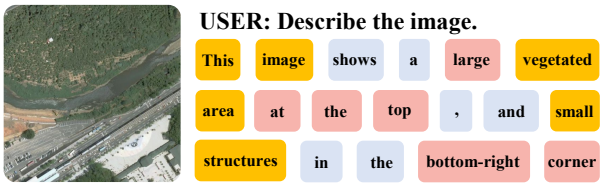

Figure 1. Visualization of the Hierarchical Generation Process for a Scene with Mixed Land Use.

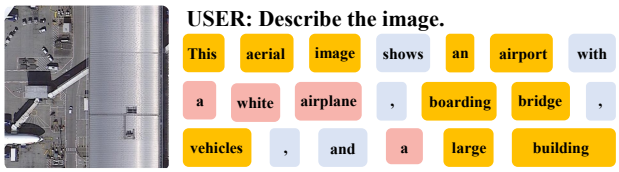

Figure 2. Visualization of the Hierarchical Generation Process for an Airport Scene.

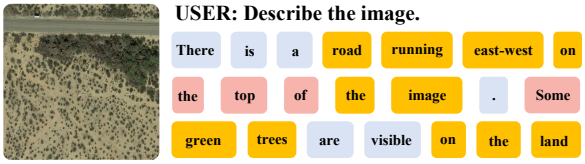

Figure 3. Visualization of the Hierarchical Generation Process for a Rural Landscape.

fundamental characteristic of our model's behavior, rather than an isolated phenomenon.

Across all presented cases, GeoDiT consistently exhibits a multi-stage, parallel refinement strategy. As illustrated in Figures 1, 2, and 3, the model first establishes the macroscopic scene context and key semantic anchors. These foundational elements, such as the primary location type (e.g., 'parking lot', 'airport') and the principal objects and their counts (e.g., 'seven', 'buses', 'airplanes'), are typically finalized in the early stages of the diffusion process (indicated by yellow tokens).

Subsequently, the model dedicates the middle stages to refining the description by adding attributes and secondary entities (pink tokens), such as descriptive adjectives ('yellow', 'large') or related objects ('shipping containers'). Finally, in the late stages, the model inserts the necessary grammatical and syntactical components ('containing', 'and', '.') to form a coherent and complete sentence (blue tokens). This coarse-to-fine process, which relies on a holistic understanding of the entire image-text relationship at every step, is structurally impossible for linear, token-by-token autoregressive frameworks.

Table 1. Quantitative analysis of failure modes.

Method	DIOR-RSVG				VRSBench			
	Det ↑	Dup ↓	Omi ↓	Fmt ↑	Det ↑	Dup ↓	Omi ↓	Fmt ↑
LLaVA-1.5	2.2	35.1	58.5	72.1	3.8	31.4	52.3	73.4
Qwen2.5-VL	17.9	15.2	21.5	87.6	19.6	13.8	19.4	85.8
GeoChat	1.4	36.8	61.3	79.2	3.4	32.5	54.0	78.7
VHM	3.4	29.5	49.2	77.6	2.8	33.8	55.3	74.3
EarthDial	13.1	18.5	29.8	82.3	0.7	45.2	65.7	81.4
GeoDiT (Ours)	**20.8**	**4.5**	**18.2**	**96.5**	**24.9**	**3.8**	**15.5**	**95.7**

4. Quantitative Analysis of Failure Modes

To further substantiate the structural superiority of our parallel refinement mechanism and provide a rigorous quantitative foundation for the qualitative failure modes highlighted in Figure 4 of the main text, we conduct a comprehensive error analysis on the multi-object detection task. Specifically, we evaluate model performance across four key dimensions: detection performance (**Det**: mAP@0.5), object duplication rate (**Dup**), object omission rate (**Omi**), and format consistency (**Fmt**: the percentage of valid, parsable bounding box coordinates generated by the model).

As detailed in Table 1, conventional autoregressive (AR) models exhibit consistently high duplication and omission rates across both the DIOR-RSVG and VRSBench datasets. This quantitative evidence strongly validates the path-dependent generative behavior visually demonstrated in Figure 4, where AR models frequently become trapped in generative loops—repeatedly predicting coordinates for the same salient object while failing to canvass the scene for other distinct entities.

In stark contrast, GeoDiT's parallel refinement mechanism effectively mitigates this sequential dependency. By resolving all spatial and semantic elements simultaneously from a global perspective, our model achieves substantially lower duplication rates (e.g., 4.5% compared to the second-best 15.2% on DIOR-RSVG) and omission rates. Furthermore, GeoDiT maintains the highest format consistency (96.5% and 95.7%), indicating a robust structural understanding. These results concretely demonstrate that breaking the sequential generation bottleneck is critical for robust, multi-object geospatial analysis.

5. Inference Efficiency Analysis

To address potential concerns regarding the computational overhead of our multi-step denoising process (specifically, utilizing N=8 inference timesteps), we conduct an empirical comparison of inference efficiency against representative autoregressive (AR) baselines. In this experiment, we evaluate the per-token inference latency alongside the total parameter size for each model to provide a comprehensive view of deployment efficiency. As summarized in Table 2, despite possessing a comparable or slightly larger parameter count (8.43B) relative to the AR baselines,

Table 2. Results on inference efficiency.

Metric	GeoChat	VHM	EarthDial	Qwen2.5-VL	Ours
Parameter Size	7.06B	6.77B	4.15B	8.29B	8.43B
Per-token Latency (ms)	26.92	21.92	13.97	31.84	**10.22**

GeoDiT achieves a highly competitive per-token latency of 10.22 ms. This empirically demonstrates that our non-autoregressive framework not only achieves superior generative quality on structured geospatial tasks but also translates to highly efficient inference speeds in practice. Notably, GeoDiT outperforms even substantially smaller AR architectures in terms of latency, such as EarthDial (4.15B parameters, 13.97 ms latency). These findings confirm that our method successfully avoids the sequential generation bottleneck of AR models, maintaining a strong balance between task performance and practical inference efficiency.

6. Controlled Experiments on Decoding Paradigms

To rigorously isolate the performance gains of our proposed diffusion-based parallel refinement from potential confounding factors—such as backbone capacity and vision encoder strength—we conduct a highly controlled comparative study. Furthermore, we address the hypothesis that the limitations of sequential autoregressive (AR) generation can be entirely overcome by advanced decoding strategies that incorporate revision or reasoning-like behaviors.

Table 3. Comparison of GeoDiT and sequential AR.

Method	Image Caption (CIDEr)		Object Detection (mAP@0.5)	
	UCM-Caption	RSICD	DIOR-RSVG	VRSBench
LLaMA-8B based (AR)	55.7	48.9	13.2	14.3
LLaMA-8B based (CoT)	56.4	49.9	13.2	14.7
LLaMA-8B based (MCMC)	62.9	75.6	14.0	17.6
GeoDiT (Ours)	**73.8**	**135.6**	**20.8**	**24.9**

Experimental Setup. We construct a size-matched AR baseline by substituting the 8B-scale diffusion generative core (LLaDA-8B) of GeoDiT with an equivalently sized 8B LLaMA-family model. To ensure strict fairness, we hold the visual encoder (SigLIP-2) and all training configurations identical to those of GeoDiT. Upon this retrained LLaMA-8B baseline, we evaluate three distinct decoding paradigms: (i) **Standard AR**, utilizing greedy, single-pass left-to-right decoding; (ii) **AR with Chain-of-Thought (CoT)**, which prompts the model to generate intermediate reasoning steps prior to final output prediction; and (iii) **AR with MCMC**, which applies Markov Chain Monte Carlo decoding to enable revision-like behavior during the AR generation process.

Results and Analysis. The quantitative comparison across image captioning (object-centric CIDEr) and object

detection (mAP@0.5) is presented in Table 3. While advanced decoding strategies do improve the performance of the AR baseline—particularly MCMC decoding, which introduces an explicit revision mechanism—GeoDiT's parallel refinement paradigm maintains a substantial and consistent advantage. For instance, on the RSICD benchmark, GeoDiT achieves a CIDEr score of 135.6, nearly doubling the performance of the strongest AR variant (MCMC: 75.6). Similarly, significant margins are observed in detection tasks (e.g., 24.9 vs. 17.6 on VRSBench).

These controlled empirical results strongly validate that our model's superiority is not merely an artifact of model scale or encoder strength. Rather, it reflects a fundamental, structural advantage of the diffusion-based holistic refinement process over sequential decoding when handling complex, non-narrative geospatial semantics.