VectorSynth: Fine-Grained Satellite Image Synthesis with Structured Semantics

Supplementary Material

1. Applications

We evaluate the utility of our generated imagery by running SegEarth-OV [5], a state-of-the-art open-vocabulary remote sensing segmentation model, on synthetic images created from structured OSM tags using our VectorSynth-COSA model. We also run the segmentation model on the grounded satellite imagery, and GeoSynth-OSM [7] for comparison. We define a subset of categories in our data that represent different land uses, buildings, and road types. We use segmentation accuracy to measure how well the generated image matches the given polygon labels, with higher accuracy meaning the generated image shows strong pixel-level fidelity to the class.

We consistently outperform GeoSynth-OSM across all categories, with particularly strong gains in fine-grained classes such as road types and distinct building uses. As shown in Table 1, our method achieves competitive results compared to real satellite imagery, and in several categories, such as land use residential, and natural regions, performance even surpasses that of real images. This indicates that our model generates semantically faithful scenes that align well with downstream open-vocabulary segmentation tasks. While challenging categories like industrial areas remain difficult due to visual ambiguity, our results demonstrate that our pretraining alignment and generation pipeline yields more spatially and semantically precise synthetic outputs.

We further perform some qualitative evaluations in Figure 1 on out of OSM distribution text prompts. We see that generated outputs adhere to spatial and semantic constraints.

2. Data

As seen in Figure 2, we densely sampled Los Angeles, New York City, Paris, and Berlin. These cities were chosen as they represent different urban planning styles: Los Angeles exemplifies low-density horizontal sprawl, New York is defined by verticality and a rigid grid system, Paris features radial layouts and dense historical cores, and Berlin reflects a blend of post-war reconstruction and structured zoning. Chicago is used as an out-of-space test city. In addition, we conduct one additional experiment on generating OSM annotations using Sydney, Australia.

We visualize the tag distribution in our training data as seen in Figure 3. Through the overlay visualization in Figure 4, we see that there is strong spatial grounding in the dataset.

Class	GeoSynth	VectorSynth	Original
place	79.54	81.26	81.81
natural region	25.26	26.04	25.55
building	19.52	32.03	32.43
industrial	4.15	18.23	36.19
apartments	5.00	14.95	22.92
school	0.50	11.39	20.69
landuse	44.62	55.90	55.06
residential	44.62	55.90	55.06
farmland	2.35	16.80	55.17
forest	12.39	28.12	36.47
sport	4.86	15.51	26.65
railway	2.93	13.51	42.04

Table 1. Segmentation accuracy (%) for parent and child classes in OSM tags using SegEarth-OV. Child classes are indented under their parent. We compare generated images from GeoSynth and VectorSynth, along with the original satellite image.

3. COSA

3.1. Architecture Details

Image Encoder. Our image encoder is built on top of SatlasNet [1]. SatlasNet is pretrained on high-resolution aerial imagery, consistent with our dataset, using a Swin-V2 backbone followed by a feature pyramid network (FPN) resulting in multi-scale feature maps of varying resolution. Following the FPN, our image encoder interpolates and concatenates the multi-scale feature maps, then passes the result through a learnable MLP adapter network to align the embedding dimension of the text encoder. Our adapter network consists of sequential 1×1 convolutional layers with ReLU activations and Batch Normalization, following a Conv2d–ReLU–BN–Conv2d–ReLU–BN structure. To encourage high resolution vision-language alignment, we freeze the Swin-V2 backbone and let the feature pyramid network, and adapter network be learnable.

3.2. Training and Implementation Details

Polygon Sampling. As the number of polygon-text pairs varies within a minibatch, contrastive sampling size can also fluctuate. To address this variability, we use a combination of minibatch size B and number of sampled polygon-text pairs K such that, with at least 95% confidence, the sampled K pairs is reached. In cases where K is not reached, we sample all polygon-text pairs within the batch. If the same multi-tag composition is in multiple images, the polygon pair is randomly chosen from one of the corresponding images. Our setup naturally introduces both *intra-*

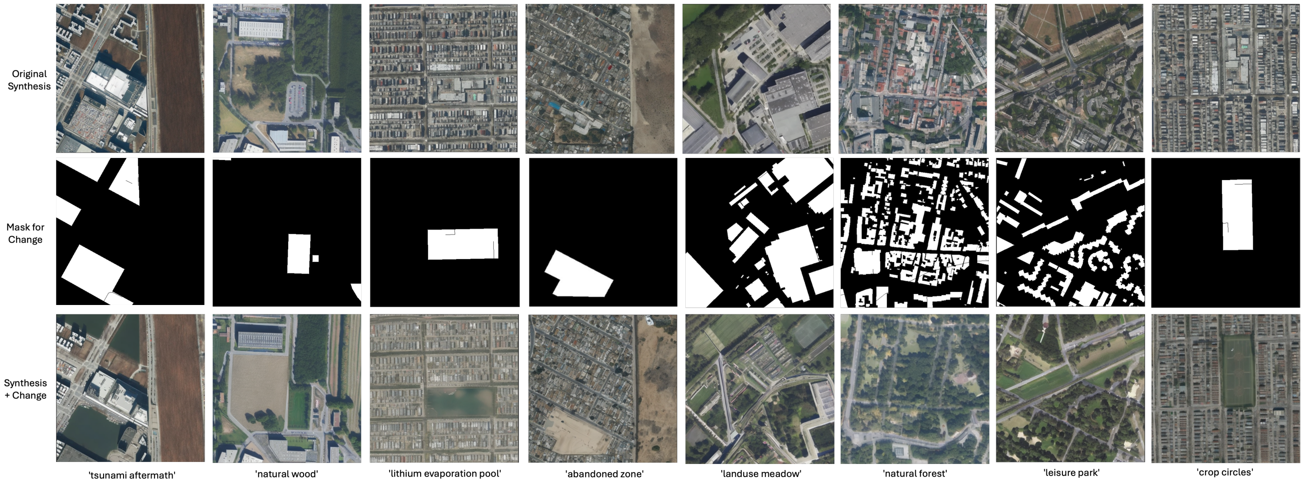

Figure 1. Qualitative evaluations on out-of-distribution text incorporated conditions, and purposeful edits. Each set shows: (top) original synthesized image, (middle) the injected control mask corresponding to the semantic change, and (bottom) the synthesized output conditioned on (below) the natural language description used for conditioning.

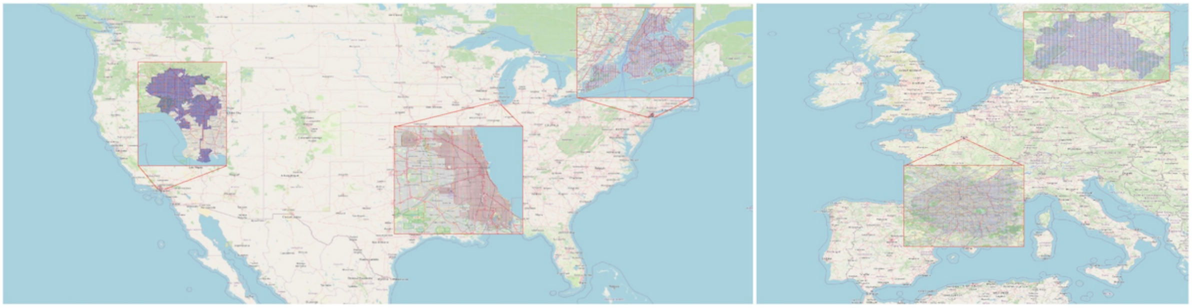

Figure 2. Geographic coverage of the dataset across five major cities: Los Angeles, New York City, Paris, Berlin, and Chicago. Each city includes training, validation, and held-out tiles, except for Chicago, which is fully held out and used only for testing. Training and validation tiles are shown in blue; Chicago test tiles are shown in red.

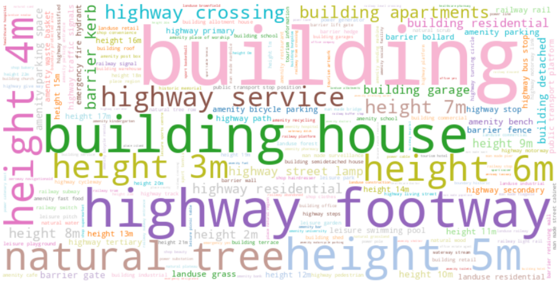

Figure 3. Word cloud of the most frequent OSM tags in the dataset. Font size reflects frequency across all tile-level tag lists. Common tags include urban structure types (e.g., `building residential`, `highway primary`), land use (`land use commercial`, `park`), and 3D features such as `height`.

image and *inter-image* negatives, encouraging distinction between semantically similar polygons-text pairs within the same image and across different images. In addition, for each contrastive pair, we sub-sample tag words in the multi-tag compositions during training to provide better generalization to varying text queries.

Optimization With Orthogonal Gradients. Due to the spatial nature of satellite images, polygon-text pairs often exhibit strong feature correlation, both within the same image and across images in the batch. This is especially true in urban areas, where OSM tag distributions and architectural layouts likely follow recurring spatial patterns (e.g., residential blocks, road grids, building clusters). Such correlations may limit learning efficiency and lead to gradient directions with poor diversity. For this reason, we implement orthogonal gradients [3], an optimization technique designed to promote diversity by projecting updates onto the gradients orthogonal component. This approach has shown effectiveness in data domains such as sequential

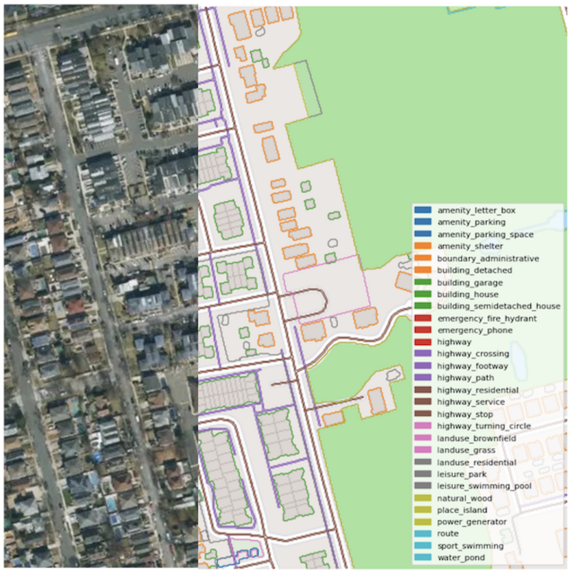

Figure 4. Overlay of OSM-derived tags on top of satellite imagery. Each region is annotated with semantically meaningful labels (e.g., `building residential`, `land use park`), showcasing the compositional richness and spatial precision of the dataset.

video frames, where the data is highly correlated. Specifically, we implement the *Orthogonal AdamW* variant implemented based on [3]. Orthogonal AdamW introduces an additional term controlled by a hyperparameter β_{ort}, set to 0.9 in our experiments.

Training Details. We train our model using the AdamW optimizer with a learning rate of $1\text{e}{-4}$, β values of $(0.9, 0.98)$, $\epsilon = 1\text{e}{-6}$, and a weight decay of 0.01. We use a cosine annealing warm restart learning rate scheduler with an initial cycle length of $T_0 = 20$ epochs. We train until early stopping with a minibatch size of $B = 6$ satellite images per GPU. Each batch includes $K = 128$ sampled polygon-tag pairs, drawn across the minibatch. Training typically ended around 80 epochs. We initialize the logit scale temperature parameter as $\log(1/0.07)$ and learn it during training. To avoid numerical instability, we clamp the logit scale to a maximum of $\log(100)$. To ensure reproducibility, we set all random seeds to 42 and disabled CuDNN benchmarking. All experiments are run on two GeForce RTX 4090 GPUs (24GB).

4. VectorSynth Controls

A qualitative comparison of the different control signals is provided in Figure 7. Visually, we observe that OSM tiles provide a high-level structural prior but lack semantic richness. While text-based pixel-level control maps introduce more diverse semantic information, our COSA control maps exhibit sharper transitions between objects, reflecting stronger inter-tag contrast and improved spatial grounding. This is especially evident in the fine-grained delineation of urban features. For example, residential and commercial buildings, as well as differences in heights of buildings, appear more homogeneous in CLIP maps, but are more distinctly separated in COSA. These improvements result from aligning OSM tag semantics with satellite imagery during pretraining, leading to control signals that are both semantically expressive and spatially localized.

5. Dealing with Sparsity

Geographic annotation datasets often suffer from inherent sparsity, where comprehensive polygon coverage is unavailable across all spatial regions. This sparsity presents significant challenges during both training and inference, as models must generate plausible geographic content even when provided with incomplete or limited control signals. We address this fundamental limitation through two complementary approaches: progressive masking during training and automated annotation enhancement using vision-language models.

5.1. Progressive Masking for Sparse Control Adaptation

To enable robust performance under sparse annotation conditions, we use a progressive masking training strategy that gradually reduces polygon coverage throughout the training process. This approach trains the model to effectively hallucinate plausible geographic features when given increasingly sparse control inputs.

Our progressive masking scheme linearly increases the proportion of masked polygons over training iterations (100% to 30%). This curriculum learning approach allows the model to first establish strong associations between dense annotations and corresponding geographic features, then gradually adapt to scenarios with limited supervisory information.

The progressive masking strategy demonstrates clear benefits for sparse control scenarios. As illustrated in Figure 8, models trained with this approach exhibit improved robustness when polygon coverage falls below 60% of the image area. However, we observe a trade-off in performance: while the progressively masked model excels with sparse controls, it slightly underperforms compared to the baseline model when provided with very dense annotation coverage. This behavior aligns with our training objective, as the model learns to rely less heavily on comprehensive annotation signals.

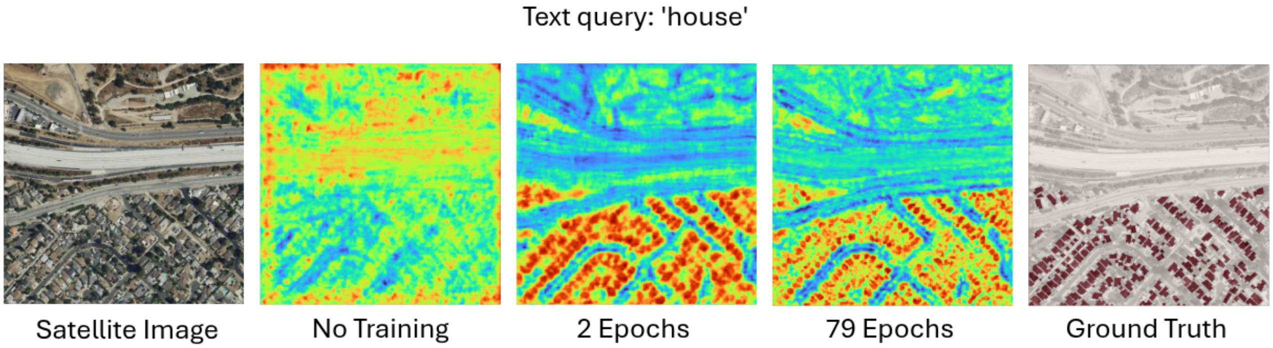

Figure 5. Similarity heatmaps for a satellite image given the text query 'house' inferred from COSA with no training, 2 epochs of training, and 79 epochs of training.

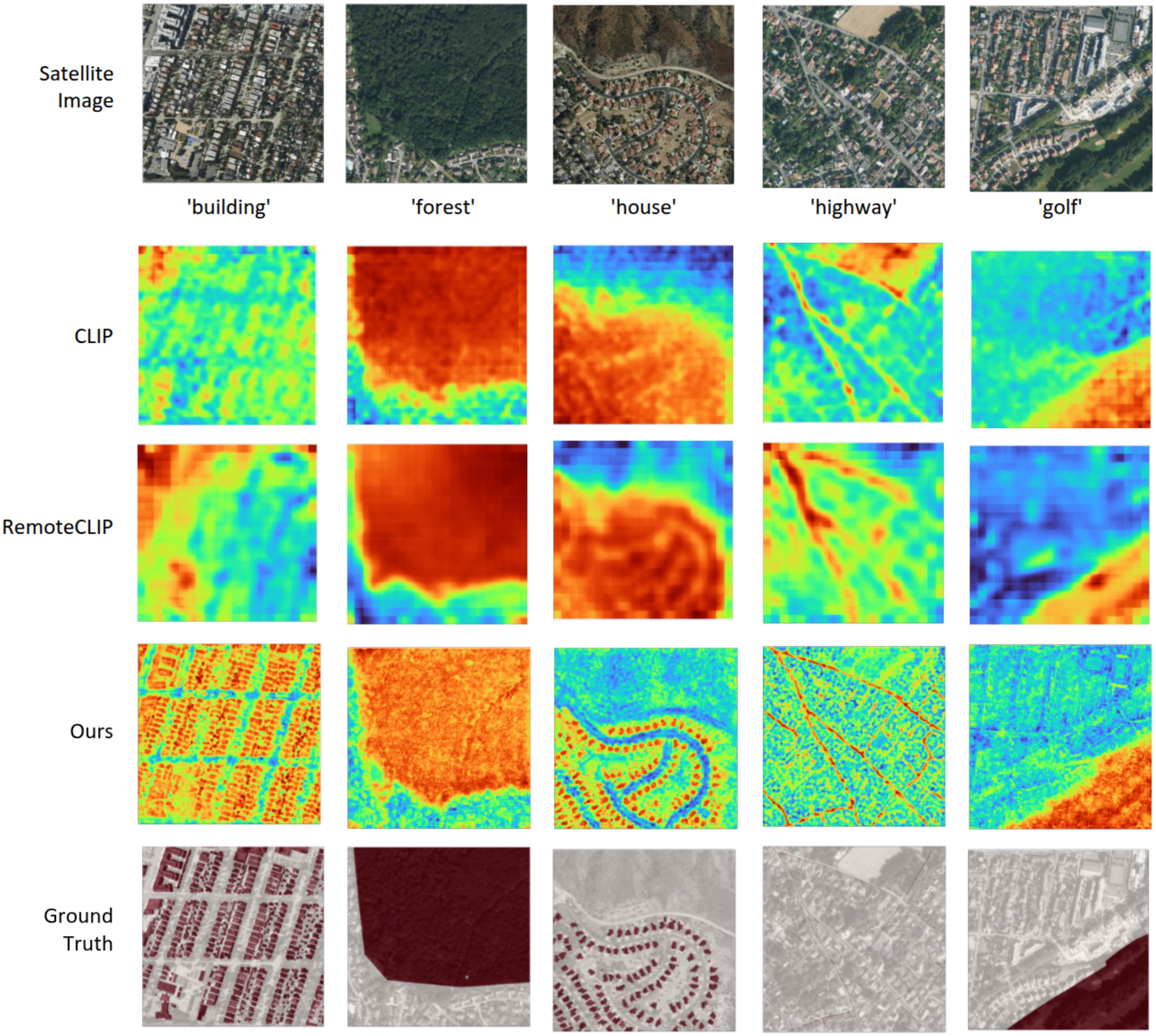

Figure 6. Similarity heatmaps given text queries comparing CLIP, RemoteCLIP, and our approach—COSA. Taking inspiration from [6], we use a sliding window inference approach to show high-resolution similarity heatmaps for CLIP and RemoteCLIP.

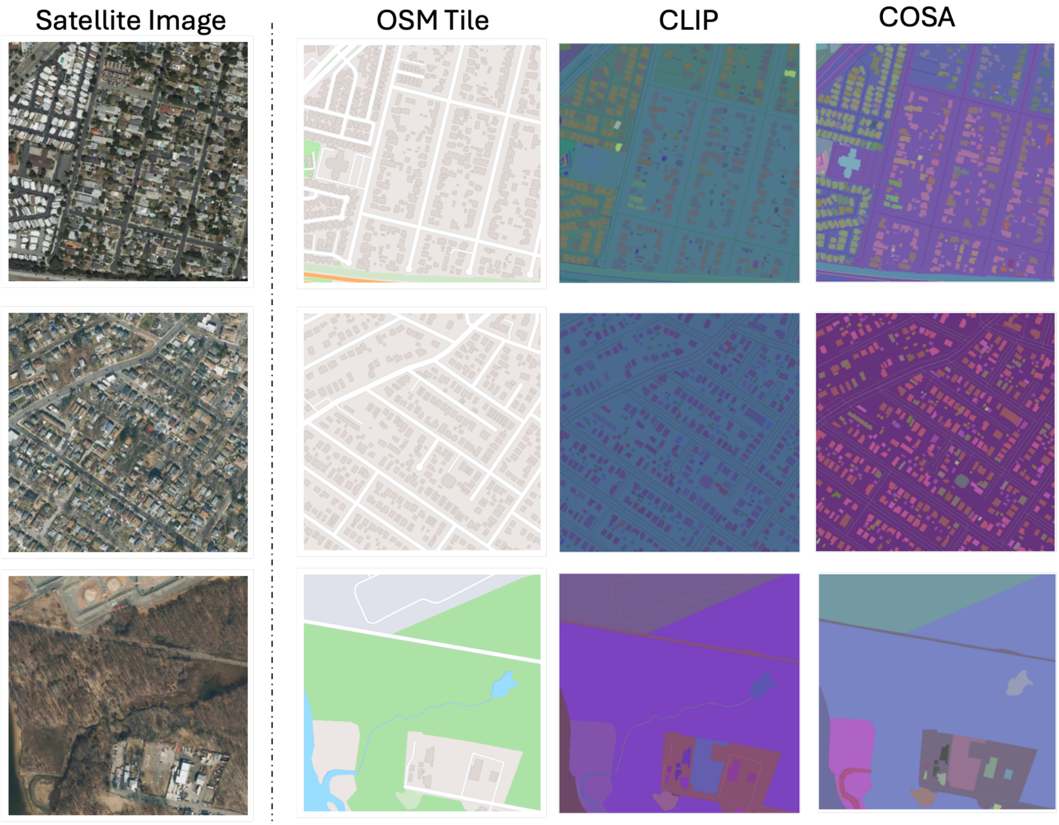

Figure 7. Qualitative comparison of control signals. OSM tiles provide coarse structural priors but lack semantic detail. Text-based maps offer richer semantics, while COSA maps show sharper object boundaries and better spatial grounding—especially in distinguishing urban features like residential and commercial buildings. These improvements stem from aligning OSM tags with satellite imagery during pretraining.

Model	**FID ↓**	**SSIM ↑**	**PSNR ↑**
GeoSynth	170.25	0.18	12.16
VectorSynth	177.17	0.17	11.99
VectorSynth (w/ generated tags)	154.13	0.18	12.11

Table 2. Comparison of FID, SSIM, and PSNR of satellite imagery across Sydney, Australia

5.2. Automated Annotation Enhancement via COSA VLM

To further address annotation sparsity, we leverage our COSA vision-language model (VLM) to automatically generate additional semantic annotations for sparse regions. This approach combines the Segment Anything Model (SAM) [4] for mask generation with our specialized COSA VLM for polygon-text retrieval, creating a pipeline that densifies sparse annotations with contextually appropriate semantic labels.

The annotation enhancement pipeline operates in three stages. First, we apply SAM [4] to the input satellite imagery to generate comprehensive segmentation masks covering all visible geographic features. Next, we utilize our COSA VLM to perform polygon-retrieval, generating semantically grounded text descriptions for each SAM-generated mask. These automatically generated text annotations are then integrated with existing sparse annotations to provide richer control signals during generation.

We evaluate this approach on an out-of-distribution dataset featuring high-resolution imagery of Sydney, Australia. Sydney's harbor-centric development and organic street patterns contrast with our training data from NYC, LA, Berlin, and Paris, which feature more geometric grids and radial planning structures. We compare three generation approaches: VectorSynth using only available Open-

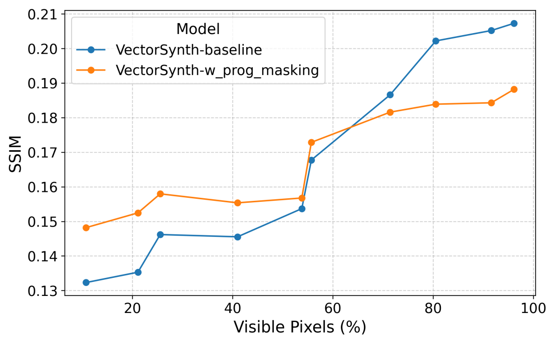

Figure 8. SSIM performance versus polygon coverage. Progressive masking (orange) outperforms baseline (blue) below 60% coverage but underperforms at dense coverage above 80%.

StreetMap (OSM) [2] tags without filtering, the baseline GeoSynth model, and VectorSynth enhanced with SAM + COSA VLM annotations. We note that we do not filter the coverage of the dataset; therefore, the OSM tags are very sparse, and many images do not contain any OSM tag information.

In Table 2, we see that using our text generation pipeline improves upon strictly using the OSM tags, and outperforms other baselines. Our experimental results demonstrate that the automated annotation enhancement pipeline can be an effective way to mitigates sparsity limitations and generate data that is useful for our vectorsynth generation.

The combination of progressive masking training and automated annotation enhancement provides a comprehensive solution to the sparsity challenge in geographic image synthesis. While progressive masking enables the model to perform well with inherently sparse controls, the COSA VLM pipeline allows us to artificially densify annotations when computational resources permit, achieving the best of both sparse and dense control paradigms.

References

[1] Favyen Bastani, Piper Wolters, Ritwik Gupta, Joe Ferdinando, and Aniruddha Kembhavi. Satlaspretrain: A large-scale dataset for remote sensing image understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16772–16782, 2023. 1

[2] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008. 6

[3] Tengda Han, Dilara Gokay, Joseph Heyward, Chuhan Zhang, Daniel Zoran, Viorica Patraucean, Joao Carreira, Dima Damen, and Andrew Zisserman. Learning from streaming video with orthogonal gradients. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 13651–13660, 2025. 2, 3

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 5

[5] Kaiyu Li, Ruixun Liu, Xiangyong Cao, Xueru Bai, Feng Zhou, Deyu Meng, and Zhi Wang. Segearth-ov: Towards training-free open-vocabulary segmentation for remote sensing images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10545–10556, 2025. 1

[6] Fan Liu, Delong Chen, Zhangqingyun Guan, Xiaocong Zhou, Jiale Zhu, Qiaolin Ye, Liyong Fu, and Jun Zhou. Remoteclip: A vision language foundation model for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–16, 2024. 4

[7] Srikumar Sastry, Subash Khanal, Aayush Dhakal, and Nathan Jacobs. Geosynth: Contextually-aware high-resolution satellite image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 460–470, 2024. 1