WalkGPT: Grounded Vision–Language Conversation with Depth-Aware Segmentation for Pedestrian Navigation

Supplementary Material

1. Implementation Details

1.1. Hyperparameter Settings.

Training configuration. WalkGPT is trained for 10 epochs with a batch size of 16 and a gradient accumulation factor of 10, resulting in an effective batch size of 160 samples. The optimizer is AdamW with a learning rate of 2×10^{-4}. All experiments use `bfloat16` precision and a maximum sequence length of 2048 tokens. Images are resized to a resolution of 448×448 before being processed by the vision encoder. Each epoch consists of 54 optimization steps, corresponding to the SANPO training split used in our setup.

Segmentation and alignment objectives. The segmentation branch optimizes a combination of Dice and BCE losses over the predicted masks. In addition, WalkGPT employs a contrastive alignment objective that pairs text-side `<SEG>` token embeddings with local SAM features. SAM produces 256-dimensional visual tokens, which are flattened and projected into the LLM hidden space using the Multi-Scale Query Projector (MSQP), configured with a 6×6 target token grid.

Loss weighting and contrastive settings. The overall objective follows the formulation described in the main paper, with loss weights $\alpha_1 = 0.1$ for the CE loss, $\alpha_2 = 0.05$ and 0.35 for the Dice and BCE segmentation losses respectively, and $\alpha_3 = 0.3$ for the InfoNCE alignment term. The InfoNCE loss uses a temperature of 0.07 and top-8 hard negative selection when computing contrastive similarities.

Query and projection modules. MSQP operates in a 1024-dimensional hidden space and uses two cross-attention layers per scale (8 attention heads), with a total of 32 queries allocated as 12/8/8/4 across $1\times$, $2\times$, $4\times$, and global scales. The resulting tokens are padded to a 6×6 grid before projection into the LLM embedding space. CTP is implemented as a calibrated MLP projector with widen factor 2 and LayerNorm, and applies a learned temperature (logit scale) to normalized text embeddings.

1.2. Computational Statistics.

We report the computational characteristics of WalkGPT to provide transparency regarding training and inference costs. All statistics correspond to the final configuration used in our experiments and are not intended as comparative benchmarks. The model contains approximately 14.1B parameters in total. Training was performed for 10 epochs on the 8.5k-sample SANPO training split, requiring approximately 6 hours on 8 GPUs. Inference throughput was measured independently, with 1k queries processed in approximately 1 hour under the same hardware configuration.

1.3. Structured Token Design

We introduce four categories of structured tokens to extend the language model vocabulary and enable multimodal grounding and spatial reasoning for the navigation task.

- **`<assessment>` and `</assessment>` Tokens:** These tags enclose a concise qualitative summary of scene accessibility, encouraging the model to generate natural language evaluations of how walkable or obstructed the environment appears.
- **`<SEG>` Tokens:** These tokens indicate objects referenced in the response that correspond to pixel-level segmentation regions. During training, each `<SEG>` token is aligned with its ground-truth mask to provide spatial grounding and interpretable visual–text associations.
- **`<p>` and `</p>` Tokens:** These tags wrap short descriptive phrases associated with specific visual elements, enabling phrase-level grounding by linking textual mentions to the corresponding regions in the image.
- **`<distance>` and `</distance>` Tokens:** These tags encode relative distances derived from SANPO depth maps, allowing the model to associate textual references with spatial proximity and improving depth-aware reasoning.

1.4. Depth Estimation Metrics.

To evaluate the numerical depth predictions produced during conversation, we introduce two complementary metrics: **Depth Accuracy (Depth Acc.)** and **Absolute Relative Error (AbsRel)**. Let d_i^{pred} and d_i^{gt} denote the predicted and ground-truth depths for object i, respectively, and let N be the total number of evaluated objects.

Depth Acc. measures the proportion of predictions that fall within a multiplicative tolerance of the ground-truth depth. Specifically, a prediction is considered correct if

$$0.5 \times d_i^{\text{gt}} \;\le\; d_i^{\text{pred}} \;\le\; 2 \times d_i^{\text{gt}}. \tag{1}$$

The metric is computed as

$$\text{Depth Acc.} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\Big(0.5\, d_i^{\text{gt}} \le d_i^{\text{pred}} \le 2\, d_i^{\text{gt}}\Big), \tag{2}$$

where $\mathbf{1}(\cdot)$ denotes the indicator function.

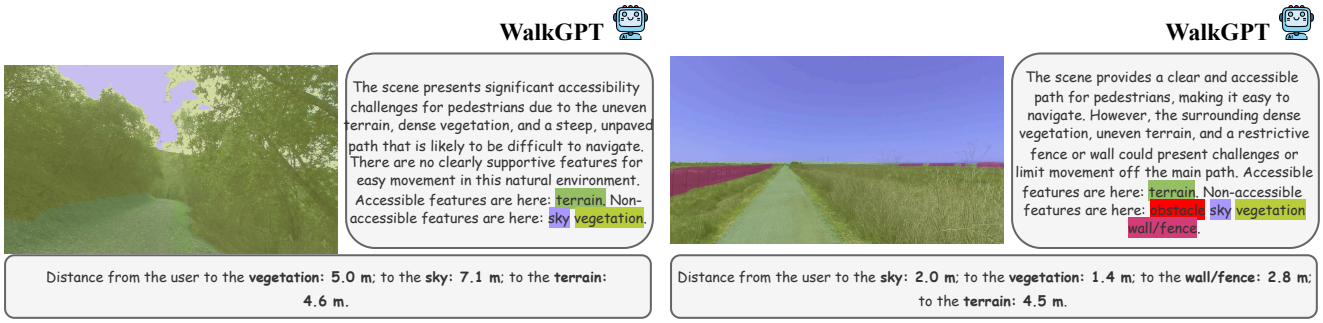

Figure 1. Additional qualitative results of WalkGPT on the PAVE validation set for off-road scenes. Examples illustrate the model's ability to handle unstructured outdoor environments with uneven terrain, dense vegetation, and limited walkable surfaces.

Absolute Relative Error (AbsRel) provides a scale-normalized measure of depth discrepancy and is defined as

$$\text{AbsRel} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|d_i^{\text{pred}} - d_i^{\text{gt}}\right|}{d_i^{\text{gt}}}. \tag{3}$$

Together, Depth Acc. captures coarse correctness within a reasonable interval, while AbsRel measures the relative magnitude of depth error with respect to the ground-truth value.

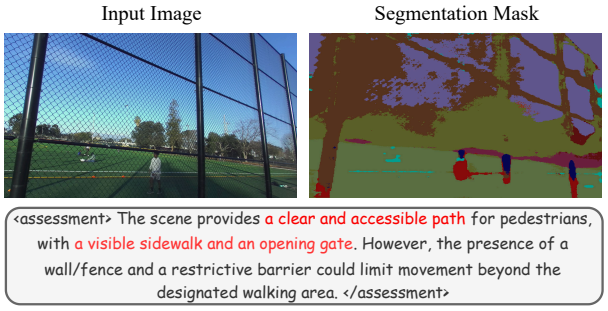

Figure 2. Another failure case on PAVE. WalkGPT incorrectly infers that the fenced area provides an open and accessible path, misled by the transparency of the fence and the clear view of the space behind it.

2. Additional Qualitative Results

Figure 1 presents additional qualitative examples from the PAVE dataset, highlighting diverse off-road scenes and their corresponding accessibility annotations. Figure 2 shows a representative failure case where WalkGPT misinterprets a fenced boundary as an open, walkable passage due to the fence's transparency.

3. Dataset Details

3.1. PAVE Dataset

SANPO: Summary. The source imagery dataset, SANPO [2], provides large-scale egocentric video captured from eye-level and chest-level viewpoints using stereo cameras mounted on real volunteer runners. Each session contains synchronized left–right video streams, associated camera poses, and both sparse depth (from the ZED sensor) and dense depth estimated with CREstereo. SANPO also includes temporally consistent panoptic segmentation for a subset of frames, high-level session attributes (e.g., environment type, visibility, motion), and hardware/IMU metadata. In addition to real captures, the dataset provides 113K synthetic frames generated under similar conditions, enabling controlled comparisons between real and simulated environments. All recordings follow strict privacy and legal guidelines, including participant review and automatic blurring of personally identifiable information.

Figure 3. Qualitative examples illustrating varied capture conditions in SANPO. (a) Motion blur and imaging artifacts. (b) Diverse outdoor environments spanning urban streets, parks, and natural trails.

SANPO: Geographic and environmental coverage. SANPO-Real consists of 701 real-world egocentric recording sessions collected across four geographically distinct locations in the United States: San Francisco (CA), Mountain View (CA), Boulder (CO), and New York City (NY).

These regions were selected to capture a diverse mix of urban cores, suburban neighborhoods, public parks, road junctions, and open pedestrian spaces. Recordings span a wide range of environmental conditions, including sunny, cloudy, rainy, and snowy weather, as well as variations in visibility, elevation change (flat, uphill, downhill, stairs), ground appearance (e.g., asphalt, pavers, gravel, terrain), and pedestrian and vehicular traffic density. Sessions also vary in time of day and motion patterns, covering walking, jogging, and running with different levels of motion blur. Figure 3 illustrates representative scenes across urban streets, park pathways, and narrow dirt trails in vegetation-dense environments.

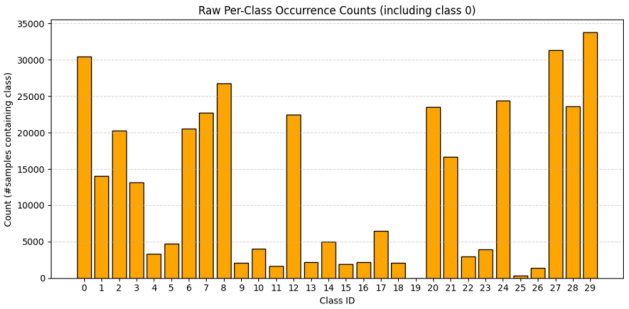

Figure 4. Per-class sample occurrence counts across all semantic categories (including background class 0). The x-axis denotes class IDs and the y-axis indicates the number of samples containing each class.

Labels. SANPO defines 30 categories spanning both semantic and panoptic annotation types, including *road* (1, semantic), *curb* (2, semantic), *sidewalk* (3, semantic), *crosswalk* (5, panoptic), *building* (7, semantic), *pedestrian* (12, panoptic), *vehicle* (21, panoptic), *tree* (28, panoptic), and additional walkability-relevant classes such as *stairs* (15, panoptic), *obstacle* (20, panoptic), and *other walkable surface* (17, semantic). Figure 4 shows the per-class sample occurrence counts, highlighting strong class imbalance across labels. Many walkability-critical classes appear far less frequently than dominant background and surface categories, making dense segmentation particularly challenging.

Processing. We use only SANPO-Real frames that include human-annotated masks, covering both semantic-only and panoptic-encoded categories. For classes annotated in panoptic format, we convert the 3-channel PNG masks into single-channel semantic masks by extracting the semantic ID from the first channel and ignoring instance identifiers. Semantic-only masks are retained as provided. When resizing is required, we apply nearest-neighbor interpolation to preserve label integrity and clamp values to the valid ID range $\{0, \ldots, 30\}$. This yields a unified semantic representation suited for accessibility reasoning, which relies on class-level occupancy rather than instance differentiation.

Depth Estimation. For each SANPO-Real frame, we use the corresponding dense depth map to compute a per-class distance from the camera and store it as ground truth for dataset construction. Because each semantic region may contain many scattered pixels, we derive a single representative depth value by taking the minimum depth among all pixels belonging to that class. This choice emphasizes the closest visible surface of each object, which is most relevant for accessibility reasoning and near-field obstacle assessment.

3.2. Prompt to Generate PAVE

To construct consistent natural-language annotations for pedestrian accessibility, we employ a large language model (LLM) to generate both the user-facing question and the structured answer associated with each scene. The generation pipeline operates in two stages. In the first stage, the LLM receives the RGB image (encoded in base64 format) through the vision-enabled GPT-5-nano API, together with a system prompt and a single formatted example. The model generates (i) a natural conversational question a pedestrian might ask about the environment and (ii) a short answer whose first block is a qualitative `<assessment>` describing overall walkability based solely on visual cues. All internal metadata (class labels, IDs, and distances) are explicitly withheld from the LLM. The JSON output is automatically validated, and malformed responses trigger a regeneration attempt.

In the second stage, the automatically generated assessment is augmented using ground-truth semantic and geometric information. Each object present in the frame is assigned to either the supportive (accessible) or harmful (non-accessible) category according to a fixed label-to-type mapping defined by the PAVE ontology. Depth values are derived from SANPO-Real dense depth maps; for each object, a representative distance is computed as the minimum depth across its pixels, corresponding to the closest visible surface and reducing occlusion bias. These elements are inserted into a fixed template to produce the final question–answer pair.

Prompt Specification for Accessibility Question-Answer Generation. The LLM is instructed to behave as a navigation assistant that generates a natural question and a structured answer in a predefined format. The question must reference helpful and harmful scene elements in general terms, remain user-facing and conversational, and avoid any internal metadata. The answer must follow a strict structure beginning with a concise `<assessment>` tag. The exact prompt used during generation is shown below.

```
You are a navigation assistant that generates
    VQA-style accessibility
question-answer pairs.
```

```
Task:
Write ONE natural-language question a
    pedestrian might ask about the
accessibility of the scene. The question must:
- refer to both helpful and harmful scene
    elements,
- be conversational and user-facing,
- NOT mention lists, ids, distances, or
    metadata,
- vary in tone and structure across samples.
Write ONE structured answer beginning with a
    short <assessment> block
using the following fixed format.
Output:
Return ONLY a valid JSON object:
{
  "question": "...",
  "answer": "..."
}
Inputs provided at runtime:
- The RGB image (base64-encoded)
```

After the qualitative <assessment> is produced by the LLM, we incorporate ground-truth semantic and geometric information to complete the structured answer. Each supportive and harmful feature is listed, followed by per-class distances computed from metric depth. The fixed template used for augmentation is shown below.

```
<assessment> ...your 1-2 sentence qualitative
    assessment... </assessment>

Accessible features are here:
<p>OBJECT_A</p><SEG><p>OBJECT_B</p><SEG> ...

Non-accessible features are here:
<p>OBJECT_C</p><SEG><p>OBJECT_D</p><SEG> ...

<distance>
Distance from the user to OBJECT_A: D_A m;
to OBJECT_B: D_B m;
to OBJECT_C: D_C m;
to OBJECT_D: D_D m;
...
</distance>
```

4. Rationale for Autoregressive Depth Learning

Although WalkGPT does not use an explicit depth regression head or a dedicated metric-depth loss, it can still learn object-level depth reasoning through the autoregressive next-token objective over structured language tokens. Depth information is provided through target <distance> tokens derived from sensor-based depth maps, but supervision occurs only at the level of object-level language tokens rather than dense depth regression. The model therefore learns to predict depth-related information jointly with grounded segmentation-aware text.

Autoregressive Factorization Couples Grounding and Depth. Let the model generate a token sequence

$$\mathbf{y} = (y_1, \ldots, y_T),$$

where some tokens correspond to grounded visual entities (<SEG> tokens) and others encode their associated natural-language distance expressions (<distance> tokens). Under the standard next-token objective, the conditional probability factorizes as

$$p(\mathbf{y} \mid \mathbf{V}_{\text{proj}}) = \prod_{t=1}^{T} p(y_t \mid y_{<t}, \mathbf{V}_{\text{proj}}), \tag{4}$$

where $\mathbf{V}_{\text{proj}}$ denotes the MSQP-projected image tokens. Because depth tokens are generated in the same sequence as grounded region references and navigation-related text, the model is trained to maintain compatibility between segmentation structure, contextual language, and depth expressions.

Local Geometry Provides a Useful Inductive Signal. MSQP produces

$$\mathbf{V}_{\text{proj}} \in \mathbb{R}^{B \times Q \times H},$$

which preserves multi-scale spatial information. These embeddings encode cues such as object extent, occlusion patterns, boundary layout, and relative scale, all of which correlate with ordinal or relative depth. Prior work has shown that vision-language models can exploit such cues for spatial reasoning even without direct metric-depth regression [1]. WalkGPT leverages the same inductive signal while grounding responses through structured tokens.

Structured Tokens Link Regions and Depth Expressions. When the model predicts a depth token for a region referenced by a preceding <SEG> token, the prediction is conditioned on the grounded context established earlier in the sequence. For example,

$$\texttt{<SEG>}_A \rightarrow \texttt{<distance>}_A,$$

requires the model to associate the referenced region A with a natural-language distance expression that is compatible with both the visual evidence and the surrounding generated text. Through self-attention over the partially generated sequence, depth prediction is therefore coupled to region identity, scene context, and previously mentioned objects.

Depth Learning Emerges as Part of the Cross-Entropy Objective. Let $z_A^\star$ denote the target <distance> token sequence associated with object A. The contribution of these positions to the autoregressive training objective can be written as

$$\mathcal{L}_{\text{dist}} = -\sum_{t \in \mathcal{T}_A} \log p(y_t^\star \mid y_{<t}^\star, \mathbf{V}_{\text{proj}}), \tag{5}$$

where $\mathcal{T}_A$ indexes the token positions corresponding to the distance expression for object A. Since these tokens are embedded in longer grounded responses, incorrect depth predictions may also weaken consistency for subsequent

grounded tokens through autoregressive conditioning. Consequently, the cross-entropy objective encourages the model to generate distance expressions that are not only locally correct but also coherent with the overall grounded description of the scene.

The overall training objective minimizes the expectation of the autoregressive loss over the dataset,

$$\mathcal{L} = \mathbb{E}_{(\mathbf{V}_{\text{proj}}, \mathbf{y}^\star)} \left[-\sum_{t=1}^{T} \log p(y_t^\star \mid y_{<t}^\star, \mathbf{V}_{\text{proj}}) \right], \quad (6)$$

of which $\mathcal{L}_{\text{dist}}$ represents the subset of terms corresponding to depth expressions.

References

[1] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024. 5

[2] Sagar M Waghmare, Kimberly Wilber, Dave Hawkey, Xuan Yang, Matthew Wilson, Stephanie Debats, Cattalyya Nuengsigkapian, Astuti Sharma, Lars Pandikow, Huisheng Wang, et al. Sanpo: A scene understanding, accessibility and human navigation dataset. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7866–7875. IEEE, 2025. 3