2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

OpenAnnotate3D: Open-Vocabulary Auto-Labeling System for Multi-modal 3D Data

Yijie Zhou[1], Likun Cai[2], Xianhui Cheng[1,3], Zhongxue Gan[1], Xiangyang Xue[1], and Wenchao Ding[1]

***Abstract*— In the era of big data and large models, automatic annotating functions for multi-modal data are of great significance for real-world AI-driven applications, such as autonomous driving and embodied AI. Unlike traditional closed-set annotation, open-vocabulary annotation is essential to achieve human-level cognition capability. However, there are few open-vocabulary auto-labeling systems for multi-modal 3D data. In this paper, we introduce OpenAnnotate3D, an open-source open-vocabulary auto-labeling system that can automatically generate 2D masks, 3D masks, and 3D bounding box annotations for vision and point cloud data. Our system integrates the chain-of-thought capabilities of Large Language Models (LLMs) and the cross-modality capabilities of vision-language models (VLMs). To the best of our knowledge, OpenAnnotate3D is one of the pioneering works for open-vocabulary multi-modal 3D auto-labeling. We conduct comprehensive evaluations on both public and in-house real-world datasets, which demonstrate that the system significantly improves annotation efficiency compared to manual annotation while providing accurate open-vocabulary auto-annotating results.**

***Source*— The source code will be released at https://github.com/Fudan-ProjectTitan/OpenAnnotate3D**

I. INTRODUCTION

The landscape of machine learning has been dominated by a paradigm where closed-set datasets are manually annotated for subsequent training and evaluation of learning models. A well-annotated benchmark can profoundly enhance the performance of corresponding tasks for both research and practical applications, exemplified by well-known datasets like ImageNet [1], COCO [2], KITTI [3], and SemanticKITTI [4].

Data and annotations are undoubtedly the cornerstone of machine learning and deep learning tasks. Particularly, with the advent of Large Language Models (LLMs) [5], [6], [7], massive amounts of data have proven to lead to breakthrough improvements in model capabilities, as demonstrated by the emergence abilities of LLMs [8]. Compared with easily obtained textual corpora on the internet, which are used to train LLMs, acquiring well-annotated multi-modal (2D & 3D) data is still a pending challenge.

Recently, the emergence of vision and language foundation models has underscored the urgency to develop an efficient annotation process for generating diverse and extensive multi-modal 3D datasets. Especially for applications like embodied AI and autonomous driving, huge amounts of annotations (2D & 3D segmentation, 3D bounding boxes) are required. Moreover, unlike traditional closed-set data annotations, open-vocabulary scene understanding [9] is the common trend to enable human-level reasoning capability. Manual annotation generation is significantly time-consuming and cannot satisfy the need for annotating open-vocabulary multi-modal 3D data. Consequently, there is a pressing need for *an open-vocabulary auto-labeling tool that can automatically generate accurate 3D annotations for multi-modal data based on various user prompts*.

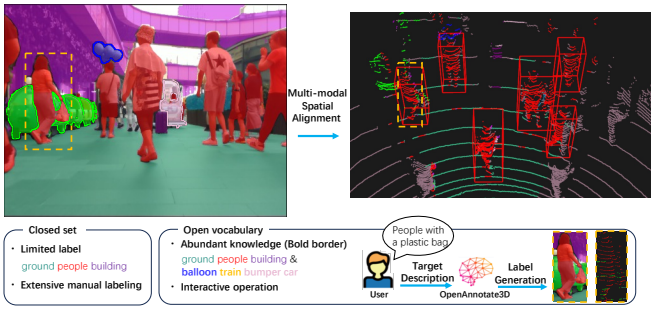

Fig. 1: An illustration of open-vocabulary multi-modal 3D annotations. Compared to closed-set annotation systems which can provide labels for known categories such as "*ground*", "*people*" and "*building*", OpenAnnotate3D can provide open-vocabulary 3D annotations for rare objects such as "*balloon*" and "*bumper car*". Moreover, OpenAnnotate3D can even understand high-level labeling commands such as "labeling the people with plastic bag".

Regarding auto-labeling methods for multi-modal 3D data, there has not been an extensive amount of research in both academia and industry. Currently, one of the most advanced and effective approaches is the Auto-Labeling Machine showcased at Tesla's AI Day 2022 [10], which is based on pre-trained models with closed-set taxonomy (predefined categories such as vehicles, pedestrians, lane topology, etc.). However, these pre-trained models struggle to effectively perform auto-labeling in open-vocabulary settings and fail to adapt to flexible labeling requirements.

Recently, LLMs have demonstrated remarkable few-shot, zero-shot, and text reasoning capabilities across a range of natural language tasks, with the most notable application being ChatGPT [11]. Taking inspiration from this, we propose a novel data annotation system called OpenAnnotate3D, which consists of an LLM-based interpreter module, a promptable vision module, and a spatio-temporal 3D auto-labeling process. Our annotation system, upon receiving multi-modal 3D data (vision and point clouds) and high-level labeling requests, such as "*labeling the balloon aside the road*" and "*labeling the rightmost cyclist with a strange payload*". The system explicitly reasons the request using the LLM-

Corresponding author: Wenchao Ding and Xiangyang Xue. [1]Fudan University, China. [2]University of Toronto, Canada. [3]Huawei Technologies Co., Ltd.
Email: dingwenchao@fudan.edu.cn
xyxue@fudan.edu.cn

interpreter, automatically matches the textual information with specific objects in the semantic 3D world, and generates 2D mask, 3D mask, and 3D bounding box annotations as shown in Fig. 1. There are two highlights for this system. First, the LLM-based interpreter module combines the LLM and promptable vision models (VLMs) in a closed-loop iterative manner, to interpret high-level user commands more precisely. Second, a spatio-temporal fusion and correction module is incorporated to overcome the imperfectness in single-frame results from VLMs.

Our contributions can be summarized as follows:

- A pioneering open-source open-vocabulary auto-labeling system for multi-modal 3D data.
- An LLM-based interpreter that interacts with promptable vision modules in a closed-loop iterative manner enabling effective reasoning of high-level commands.
- A spatio-temporal fusion and correction method that overcomes imperfectness in single-frame auto-labeling.
- Extensive experiments to validate the superior efficiency and open-vocabulary scene understanding capability of the proposed system.

II. RELATED WORK

A. 2D Annotation

For annotating 2D RGB data, numerous tools have been developed like LabelMe [12], Vatic [13], Label Studio [14], VIA [15], DEXTR [16], PolygonRNN++ [17], and CVAT [18]. These annotation tools cover most RGB-based vision tasks from basic image classification to video annotations. Since manual labeling is extremely time-consuming, most of these tools support model-assisted auto-labeling functions. For example, DEXTR [16] and PolygonRNN++ [17] can be used to obtain precise dense annotations with manually provided coarse information, such as bounding boxes and extreme points. Several open-sourced annotation tools like CVAT as well as commercial tools (Roboflow [19], Labelbox [20]) support SAM [21] to boost the efficiency of annotating. However, most of these labeling tools remain in 2D and fail to handle multi-modal 3D data.

B. 3D Annotation

Compared to annotating intuitive 2D RGB data, annotating 3D point clouds is inherently more complicated due to the sparsity and irregularity of 3D data. In open-sourced annotation tools mentioned above, only CVAT supports manual annotation of 3D bounding boxes on point clouds. In [22], a min-cut base method was presented to segment a single object from the background in 3D point clouds. To expand to multi-object segmentation, [23] developed an interactive method based on the shortest path tree, requiring the user to select sparse control points in a 3D scene. In [24], a deep network is introduced for 3D instance segmentation, which generalizes well to previously unknown objects with little manual annotation effort. If the input data includes both 2D RGB and 3D point clouds, LATTE [25] and LiLaNet [26] support 3D point cloud segmentation guided by 2D masks. PALF [27] uses a pre-trained 3D object detection model to generate 3D bounding boxes and calibrate them using 2D bounding boxes.

These annotation tools for 3D point clouds mentioned above generally either require users to annotate within the point cloud data space directly or have complex and intricate operational logic. All of these conditions significantly raise the threshold and workload for annotating 3D point clouds. Moreover, few of these annotation tools support open-vocabulary annotations. In contrast, OpenAnnotate3D provides a systematic solution for open-vocabulary auto-labeling for multi-modal 3D data.

III. SYSTEM ARCHITECTURE

In this section, we introduce the workflow of OpenAnnotate3D, as well as its implemented components in detail. Fig 2 illustrates the whole auto-labeling process of our system, which takes a text description $T \in \mathbb{R}^N$, RGB image $I \in \mathbb{R}^{M \times N \times 3}$, and 3D point clouds $P \in \mathbb{R}^{N \times 3}$ as input. To further reduce the frequency of physical interaction for users, our system also supports voice input. These voice signals are automatically transcribed to text using a speech recognition model, Whisper [28]. Our system accomplishes the generation of precise 2D mask, 3D mask, and 3D bounding box annotations based on any user-provided descriptive text.

A. LLM-based Interpreter Module

Our system is designed to annotate one or multiple open-vocabulary instances based on flexible user-provided text descriptions. The labeling request can be high-level and abstract, such as "labeling the balloon on the road". To this end, an LLM is employed as a semantic interpreter to transform the user-provided prompt into plain text outputs that can be understood by VLMs. The reason is that even the recent state-of-the-art promptable vision modules (VLMs) suffer from limited textual reasoning capabilities compared to LLMs, which may result in poor visual recognition results if we directly feed raw user text commands to the promptable vision module. Given the LLM-based interpreter, users only need to provide a high-level text phrase for labeling commands rather than elaborately design a segmentation algorithm before the annotation process.

1) ***Prompt Engineering****:* For direct prompts such as "*garbage bin with trash on it*", we augment the text with a pre-defined prompt, which is shown in Fig. 3. The prompt template primarily includes three components: 1) the fundamental role of the LLM interpreter and its basic task description; 2) several important rules for the interpreter regarding the format of outputs; 3) the conversation history of the last 5 text inputs from users. This prompt engineering enables LLMs to better interpret user-provided text, minimizing any prior knowledge, and thus allowing the subsequent vision module to achieve a higher hit rate.

For high-level prompts such as "*generate 3D bounding box for the third car from the left*", we first parse user input using the LLM, extracting all relevant information (even from internet). Subsequently, we conduct a coarse query using the promptable vision module, deriving 2D segmentation

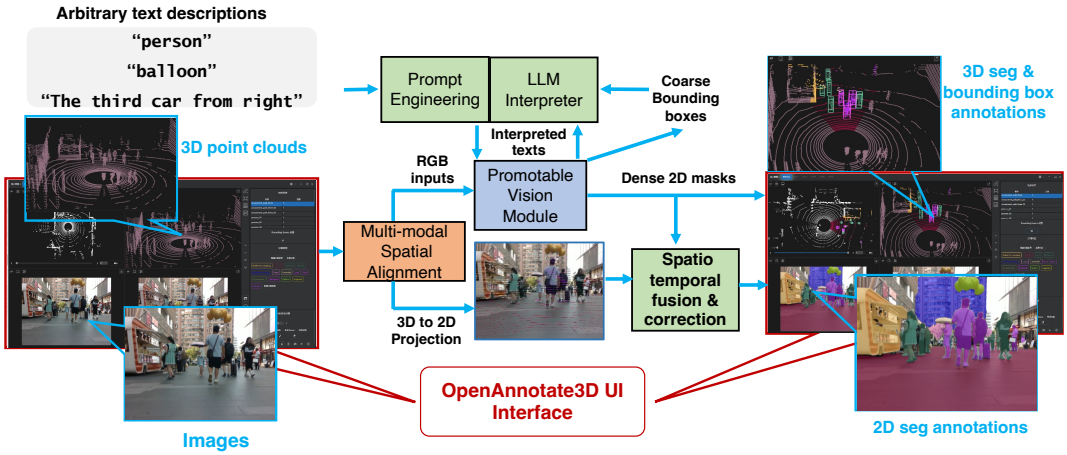

Fig. 2: Workflow of OpenAnnotate3D. Upon receiving a user's labeling request, the system first reasons about the request through the LLM interpreter and proper prompt engineering. Note that the interpreter may interact with the promptable vision module for several rounds so that the interpreted texts fit the reasoning capability of the promptable vision module. Then dense 2D masks are produced and 3D masks are further calculated through multi-modal spatial alignment. To overcome the imperfectness in 2D masks, spatio-temporal fusion and correction are carried out to refine the 3D labels.

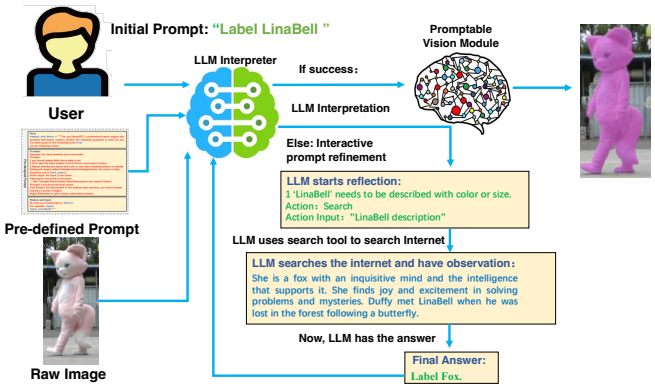

Fig. 3: Illustration of the process of interpretation based on a pre-defined prompt. Using the pre-defined prompt template, a role can be assigned to the LLM, specifying available tools. Furthermore, interaction history with the promptable vision module is memorized and incorporated.

data and feeding the segmentation quality back to the LLM. Drawing on its scene comprehension, the LLM can then accurately discern the user's intention and target, slightly adjust the output interpreted texts, and prompt the vision module again. A toy example is depicted in Fig. 4. This process can be iterative as elaborated as follows.

*2) **Iterative Text Interpretation**:* We devise an iterative text interpretation strategy as outlined in Algo. 12, which is designed to better connect open-vocabulary user prompts to a downstream promptable vision module. Initially, we feed the original user-provided text to the promptable vision module. If the vision module cannot establish a match between the text description and image, it provides feedback to the LLM interpreter. The prompt history is memorized and further incorporated into the next prompt. Then the LLM interpreter adjusts its outputs leveraging language understanding and reasoning abilities embedded in LLMs until the promptable vision module can understand its instructions well.

Suppose the visual module still cannot generate a valid

Algorithm 1: Iterative Text Interpretation

Inputs: RGB image: I, user-provided text: T_0;
Outputs: A set of bounding boxes: $\mathcal{B}$;
$\mathcal{B} \leftarrow \varnothing$;
for $i \leftarrow 0$ *to* L **do**
  $\mathcal{B} \leftarrow \texttt{VisionModule}(T_i, I)$;
  if $\mathcal{B} == \varnothing$ **then**
    $T_{i+1} \leftarrow \texttt{LLMInterpreter}(\texttt{PromptEng}(T_i))$;
  else
    End this for loop;
  end
end
return $\mathcal{B}$

output after L iterations, our annotation system interrupts and provides feedback to the user, requesting them to refine their text input to describe desired objects. Additionally, when mask annotations are generated, our system also allows users to assess these annotations. If they are dissatisfied with the results, this feedback is also conveyed to the interpreter, enabling the system to continue iterating for better annotations.

B. Promptable Vision Module and 3D Auto Labeling

Following the LLM-based interpreter, we build a labeling process that can automatically annotate 3D multi-modal data. Current off-the-shelf cross-modality vision-language models are based on 2D images, such as CLIP [29] and SAM [21]. In this section, we will elaborate on how to annotate 3D multi-modal data based on off-the-shelf VLMs.

*1) **Multi-Modal Spatial Alignment**:* As aforementioned, our OpenAnnotate3D is designed to perform object-level labeling on RGB and 3D point cloud data. There are few open-vocabulary models directly operating on multi-modal 3D data. To this end, we conduct multi-modal spatial alignment so that the reasoning capability of 2D VLMs can be better utilized.

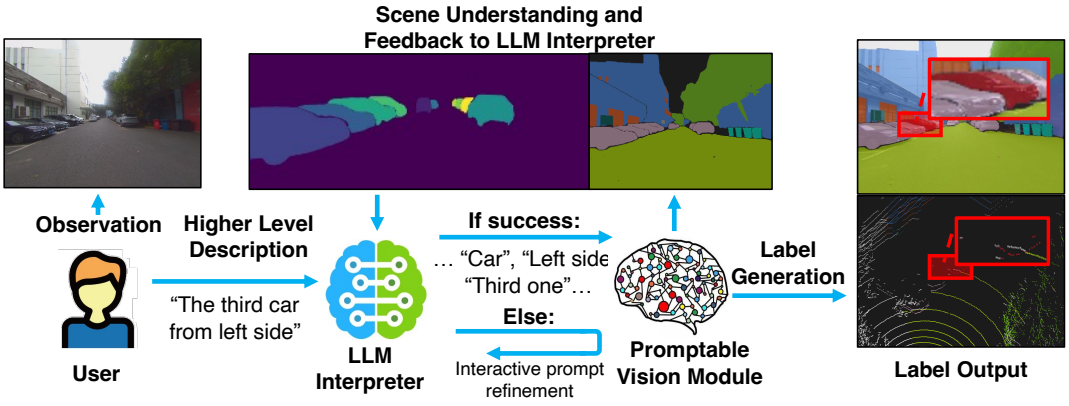

Fig. 4: Pipeline for iterative text interpretation. LLM first interprets the user's goal prompt, extracts the core content, and then conducts an initial query to the promptable vision module to obtain scene features. The prompt for the vision module is then constantly refined by the LLM interpreter based on the scene understanding results from the promptable vision module. This enhances the reasoning capability and segmentation accuracy significantly.

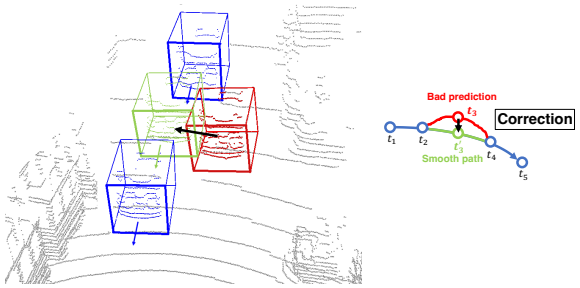

Fig. 5: Illustration of spatio-temporal fusion and correction.

When RGB and 3D point clouds are spatially aligned, precise 2D masks can be directly projected to the 3D space to serve as 3D segmentation annotations. Now that 2D mask annotations are produced from the vision module, to obtain 3D annotations, we need to spatially align RGB camera images with 3D LiDAR point clouds.

We directly transform 3D point clouds in world coordinates into 2D image coordinates using the extrinsic and intrinsic parameters as follows:

$$\mathbf{s}[u_i, v_i, 1]^\top = \overrightarrow{P}[x_i, y_i, z_i, 1]^\top \tag{1}$$

where $[u_i, v_i, 1]^\top$ and $[x_i, y_i, z_i, 1]^\top$ are 2D homogeneous image coordinates and 3D homogeneous world coordinates, respectively. $\overrightarrow{P} = K \cdot [R \mid \mathbf{t}]$ represents the projection matrix, where $K \in \mathbb{R}^{3\times3}$ is the intrinsic matrix, and $[R \mid \mathbf{t}] \in \mathbb{R}^{3\times4}$ is the extrinsic matrix, consisting of the rotation matrix $\mathbb{R}$ and translation vector $\mathbf{t}$. $\mathbf{s}$ is a scaling factor.

Given well-aligned RGB images and 3D point clouds, we can establish an accurate point-to-pixel correspondence. In Sec. III-B, we obtain 2D masks through the promptable visual module, which is implemented using VLMs such as SAM. Interested readers may refer to Sec. IV for implementation details. Based on the semantic object annotated in 2D image coordinates, we can label corresponding points within the same area as the same semantic object. When these point clouds are projected back to 3D world coordinates, we can directly obtain 3D mask annotations for different objects. Additionally, our system also supports labeling 3D bounding boxes by fitting 3D bounding boxes to segmented and clustered 3D point clouds.

*2) **Spatio-temporal Fusion and Correction**:* When dealing with multi-frame video data, we offer two optional solutions enabling continuous frame annotation. In the first approach, users can explicitly specify the starting and ending frames within a video segment. Once the system automatically labels the two frames, an interpolation algorithm is employed to annotate the remaining frames in this video. This approach is highly efficient but may not guarantee the accuracy of annotations for intermediate frames.

Therefore, our system also supports frame-by-frame auto-labeling for videos. However, the issue is that the VLMs may occasionally mislabel or miss certain objects for particular frames, which may result in poor 3D annotation quality, especially for difficult cases such as occlusion.

To this end, we propose a fusion and correction method based on the observation that it is essential to utilize spatial and temporal information across frames. If we consider time as an additional axis, a moving object will generate a 3-dimensional volume over time. A cross-section of this volume represents the object's instantaneous pose in time. Given that the majority of objects in the physical world adhere to kinematic laws, maintaining geometric and spatial consistency, we can evaluate and correct the trajectory of an object. Fig 5 demonstrates how the spatio-temporal fusion and correction fix the result of an incorrect annotation.

IV. EXPERIMENT

To evaluate our OpenAnnotate3D system, we conducted experiments on both public benchmark and in-house multi-modal datasets.

A. Implementation Details

For the LLM interpreter module in our system, we utilized `gpt-3.5-turbo` API from OpenAI [30] and Langchain API from Langchain [31]. In the promptable vision module, we integrate two off-the-shelf pre-trained models, Grounding DINO [32] and SAM [21], without any training or finetuning. Both of these models are robust foundational models that

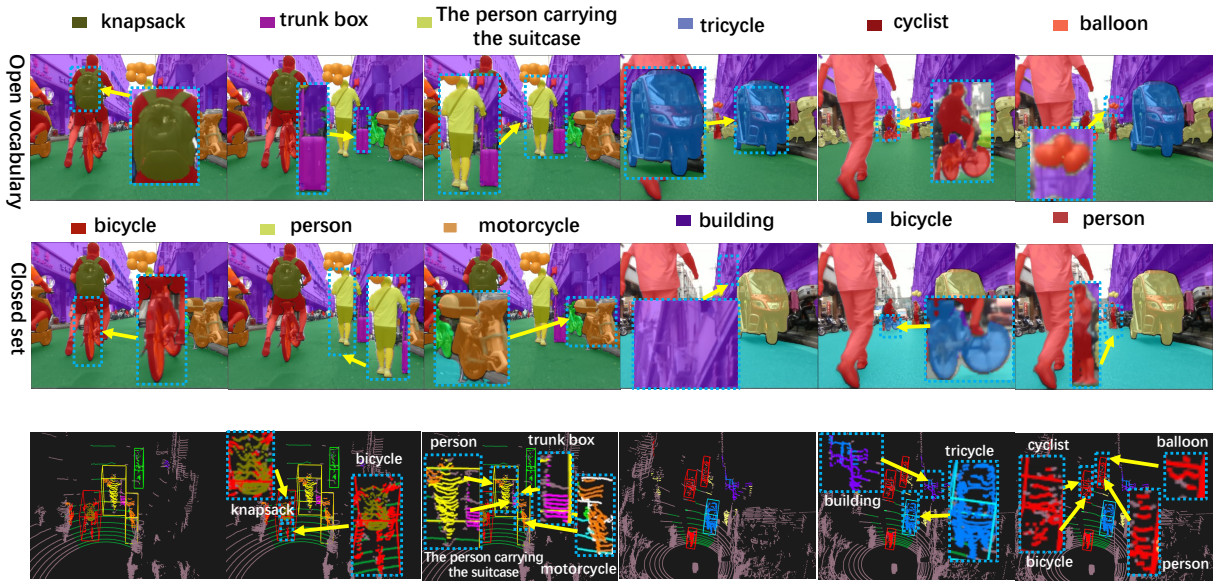

Fig. 6: Visualization of open-vocabulary annotations generated by our OpenAnnotate3D in in-house datasets.

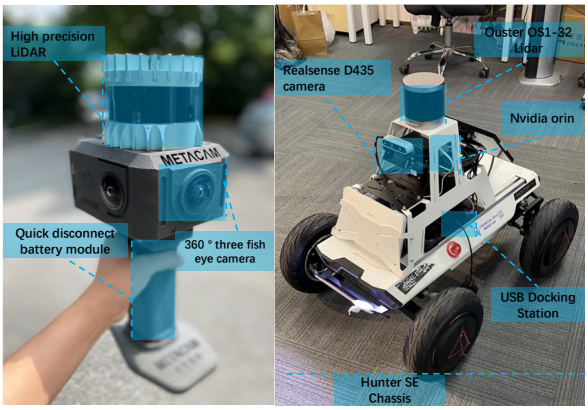

Fig. 7: Two devices utilized to record in-house datasets.

support vision-language inputs. Given user-provided texts and 2D images, we employ Grounding DINO to generate 2D proposal bounding boxes for images based on text feature matching. Subsequently, these bounding boxes are fed to the prompt encoder of SAM, serving as segmentation prompts. SAM generates the final 2D mask based on these bounding boxes. It is worth noting that Grounding DINO and SAM are plug-and-play components and can be replaced with other models that have similar functionalities. All experiments are implemented on a single machine with a RTX-4090.

We primarily conduct experiments on two types of data:

- **Public Dataset** To assess the labeling accuracy, we use our OpenAnnotate3D system to generate annotations on SemanticKITTI [4] and compare the auto-labeling results with ground truth annotations provided by SemanticKITTI. SemanticKITTI was collected using a 3D LiDAR and dual cameras, capturing 3D point clouds and RGB images from both left and right perspectives. Since our annotation system relies on 2D images to annotate 3D data, when evaluating it on SemanticKITTI, we choose the left-view images and filter out the point clouds that were not covered by the left-view images in the 3D scene. In addition, SemanticKITTI consists of 22 sequences, each containing thousands of frames. For ease of comparison, we only selected the 08 *val* sequence as the evaluation subset.
- **In-house Dataset** Public datasets only contain closed-set annotations of driving scenes, which are monotonous. To comprehensively evaluate open-vocabulary reasoning, we record a series of complex open-scene data using two devices. The first device is a handheld 3D reality scanner called Metacam, which is equipped with a 32-line LiDAR and three fish-eye cameras. The LiDAR has a field of view of 45° × 360°, while the cameras output 4032×3040 px full-color images. In addition, we also utilized an autonomous grounded vehicle with a 32-line LiDAR and a 640×480 camera. The two devices are shown in Fig. 7.

B. Metrics

To evaluate the annotation performance of OpenAnnotate3D on the 3D semantic segmentation task of the SemanticKITTI dataset, we employed IoU (intersection-over-union) for each class as the evaluation metric, i.e.,

$$\text{IOU}_c = \frac{TP_c}{TP_c + FP_c + FN_c} \tag{2}$$

where TP_c, FP_c, and FN_c correspond to the number of true positive, false positive, and false negative point predictions for class c.

Note that evaluating annotation results on closed-set datasets using IOU metrics is because closed-set annotations have ground truth. However, apart from annotating these closed-set objects, OpenAnnotate3D is capable of labeling various open-set objects, which is not reflected in the quantitative experiments due to the lack of ground truth.

C. Quantitative Analysis of Accuracy and Efficiency

Apart from labeling accuracy, we also evaluate labeling efficiency. To this end, we recruited two human annotators

TABLE I: Evaluation of 3D semantic segmentation results on selected SemanticKITTI *val* subset.

Annotation Methods	road	car	person	vegetation	building	pole	motorcycle
Junior User IOU(%)	88.6	63.1	35.4	67.2	59.1	18.1	59.8
Senior User IOU(%)	91.5	95.3	67.8	69.9	88.1	45.3	84.7
OpenAnnotate3D IOU(%)	94.2	92.3	75.3	81.4	85.7	58.2	93.8
OpenAnnotate3D w.o. spatio-temporal fusion(%)	94.2	87.4 (-4.9)	72.3 (-3.0)	81.4	85.7	58.2	88.5 (-5.3)

TABLE II: 3D semantic segmentation time costs on selected SemanticKITTI *val* subset (30 frames).

Annotation Methods	road	car	person	vegetation	building	pole	motorcycle
Junior User (Sec)	168	110	95	226	183	75	134
Senior User (Sec)	152	98	87	200	162	65	120
OpenAnnotate3D (Sec)	2	2	2	2	2	2	2

(one junior and one senior) who had received training in the annotating process. In the baseline comparison experiment, we manually and automatically annotated 10 objects for 30 frames from SemanticKITTI, respectively.

OpenAnnotate3D supports a manual fine-tuning interface for users. In practice, we find that through slight manual corrections based on OpenAnnotate3D, the whole system is even more powerful thanks to the iterative process in the LLM interpreter and spatio-temporal fusion and correction. However, in this part of the experiment, we only tested the automatic annotation components of OpenAnnotate3D to represent pure auto-labeling accuracy.

We record the precision of the annotations compared to the ground truth and the time taken by different annotators to complete the tasks. The annotation results are shown in Tab. I, where we present the IoU for each category. Especially for objects with complex shapes like "*person*", "*vegetation*", or relatively small objects like "*pole*", even senior human annotators achieve IoU of only 67.8%, 69.9%, and 45.3%, respectively. In contrast, our OpenAnnotate achieves IoU of 75.3%, 81.4%, and 58.2%, respectively, without any manual fine-tuning. For objects that are challenging for the human eye to precisely identify, our automatic system demonstrates a more pronounced advantage. The time costs are shown in Tab. II. As we can see, our OpenAnnotate3D incurs significantly lower time consumption compared to manual annotation, especially for objects with irregular shapes and large areas such as "*vegetation*" and "*motorcycle*". Furthermore, our OpenAnnotate3D, with consistent program execution speed (dependent mainly on GPU performance), can be quantified in terms of time. In contrast, manual annotation not only exhibits low efficiency but also varies among different users on their level of expertise.

D. Qualitative Analysis of Open Vocabulary Reasoning

In addition, as shown in Fig. 6, we further demonstrate the annotation capabilities of our OpenAnnotate3D on real-world scene data. Our annotation system not only consistently and automatically annotates several common closed-set objects such as "*bicycle*", "*person*", "*building*", and "*motorcycle*", but also accurately identifies numerous open-vocabulary objects that were not previously annotated in the closed-set dataset. These open-vocabulary objects include "*balloon*", "*knapsack*", "*trunk box*, as well as long descriptions like "*the person carrying the suitcase*". These examples highlight the powerful open-vocabulary annotation capabilities of our annotation system.

E. Ablation Studies

We also conduct an ablation study to evaluate the automatic correction function of our OpenAnnotate3D, namely the spatio-temporal confusion and correction module. Using the same setup as the previous experiments, we first allowed OpenAnnotate3D to perform automatic correction. Then, we conducted another round of annotation with the automatic correction disabled. The results are shown in rows 3 and 4 of Tab. I. It can be observed that after undergoing automatic correction with the spatio-temporal confusion module, the annotation system's precision is further improved, especially for moving objects like "*motorcycle*" and "*car*".

F. Limitations

Our OpenAnnotate3D tool still exhibits some degree of dependency on user inputs. For prompts that are ambiguous or overly abstract, such as "other-vehicle," the annotation capabilities of the tool may be somewhat limited. Additionally, the performance of OpenAnnotate3D is subject to hardware-specific parameters, including camera resolution, frame rate, and laser scanner resolution. In cases where camera resolution is sub-optimal, annotation results for distant or unclear objects may not meet the desired standards.

V. CONCLUSION

In this paper, we propose OpenAnnotate3D, an open-source open-vocabulary auto-labeling system for multi-modal 3D data, which includes an LLM-based interpreter module, a promptable vision module, and a spatial-temporal 3D auto-labeling process. OpenAnnotate3D integrates the chain-of-thought capabilities of Large Language Models (LLMs) and the cross-modality capabilities of vision-language models. To the best of our knowledge, OpenAnnotate3D is one of the pioneering works for open-vocabulary multi-modal 3D auto-labeling.

VI. ACKNOWLEDGMENT

This work is supported in part by Shanghai Pujiang Program (23PJ1400900), NSFC Project(62176061) and Shanghai Technology Development and Entrepreneurship Platform for Neuromorphic and AI SoC.

REFERENCES

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[3] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[4] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9297–9307.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[6] OpenAI, "Gpt-4 technical report," 2023.

[7] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[8] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, *et al.*, "Emergent abilities of large language models," *arXiv preprint arXiv:2206.07682*, 2022.

[9] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, *et al.*, "Openscene: 3d scene understanding with open vocabularies," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 815–824.

[10] Tesla, "Tesla ai day 2022," 2000. [Online]. Available: https://www.youtube.com/watch?v=ODSJsviD_SU&ab_channel=Tesla

[11] OpenAI, "Chatgpt (september 17 version)," https://chat.openai.com/chat, 2023.

[12] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, pp. 157–173, 2008.

[13] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation: A set of best practices for high quality, economical video labeling," *International journal of computer vision*, vol. 101, pp. 184–204, 2013.

[14] M. Tkachenko, M. Malyuk, A. Holmanyuk, and N. Liubimov, "Label Studio: Data labeling software," 2020-2022, open source software available from https://github.com/heartexlabs/label-studio. [Online]. Available: https://github.com/heartexlabs/label-studio

[15] A. Dutta and A. Zisserman, "The VIA annotation software for images, audio and video," in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM '19. New York, NY, USA: ACM, 2019. [Online]. Available: https://doi.org/10.1145/3343031.3350535

[16] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 616–625.

[17] D. Acuna, H. Ling, A. Kar, and S. Fidler, "Efficient interactive annotation of segmentation datasets with polygon-rnn++," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 859–868.

[18] B. Sekachev, M. Nikita, and Z. Andrey, "Computer vision annotation tool: A universal approach to data annotation. 2019," *URL https://github.com/opencv/cvat*, 2019.

[19] B. Dwyer, J. Nelson, J. Solawetz, and et al., "Roboflow (version 1.0)," Available from https://roboflow.com, 2022, computer Vision.

[20] Labelbox. (2023) Labelbox. [Online]. [Online]. Available: https://labelbox.com

[21] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[22] A. Golovinskiy and T. Funkhouser, "Min-cut based segmentation of point clouds," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, 2009, pp. 39–46.

[23] R. Monica, J. Aleotti, M. Zillich, and M. Vincze, "Multi-label point cloud annotation by selection of sparse control points," in *2017 International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 301–308.

[24] T. Kontogianni, E. Celikkan, S. Tang, and K. Schindler, "Interactive object segmentation in 3d point clouds," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2891–2897.

[25] B. Wang, V. Wu, B. Wu, and K. Keutzer, "Latte: accelerating lidar point cloud annotation via sensor fusion, one-click annotation, and tracking," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 265–272.

[26] F. Piewak, P. Pinggera, M. Schafer, D. Peter, B. Schwarz, N. Schneider, M. Enzweiler, D. Pfeiffer, and M. Zollner, "Boosting lidar-based semantic labeling by cross-modal training data generation," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[27] Y. Zhang, M. Fukuda, Y. Ishii, K. Ohshima, and T. Yamashita, "Palf: Pre-annotation and camera-lidar late fusion for the easy annotation of point clouds," *arXiv preprint arXiv:2304.08591*, 2023.

[28] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[29] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[30] OpenAI. (2023) OpenAI. [Online]. Available: https://www.openai.com

[31] H. Chase, "Langchain," Available at https://github.com/hwchase17/langchain, 2022, cff-version: 1.2.0.

[32] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.