Aerial Vision-and-Dialog Navigation

Yue Fan, Winson Chen, Tongzhou Jiang, Chun Zhou, Yi Zhang, Xin Eric Wang

University of California, Santa Cruz

{yfan71, wchen157, tojiang, czhou43, yiz, xwang366}@ucsc.edu

Abstract

The ability to converse with humans and follow natural language commands is crucial for intelligent unmanned aerial vehicles (a.k.a. drones). It can relieve people's burden of holding a controller all the time, allow multitasking, and make drone control more accessible for people with disabilities or with their hands occupied. To this end, we introduce Aerial Vision-and-Dialog Navigation (AVDN), to navigate a drone via natural language conversation. We build a drone simulator with a continuous photorealistic environment and collect a new AVDN dataset of over 3k recorded navigation trajectories with asynchronous human-human dialogs between commanders and followers. The commander provides initial navigation instruction and further guidance by request, while the follower navigates the drone in the simulator and asks questions when needed. During data collection, followers' attention on the drone's visual observation is also recorded. Based on the AVDN dataset, we study the tasks of aerial navigation from (full) dialog history and propose an effective Human Attention Aided Transformer model (HAA-Transformer), which learns to predict both navigation waypoints and human attention. Dataset and code are released: https://sites.google.com/view/aerial-vision-and-dialog/home.

1 Introduction

Drones have been widely adopted for many applications in our daily life, from personal entertainment to professional use. It has the advantage of mobility and observing large areas over ground robots. However, compared with ground robots, the control of the aerial robot is more complex because an extra degree of freedom, altitude, is involved. To control a drone, people often need to hold a controller all the time, so it is essential to create a hands-free control experience for drone users and develop an intelligent drone that can complete tasks simply by talking to humans. It can lower the barrier of
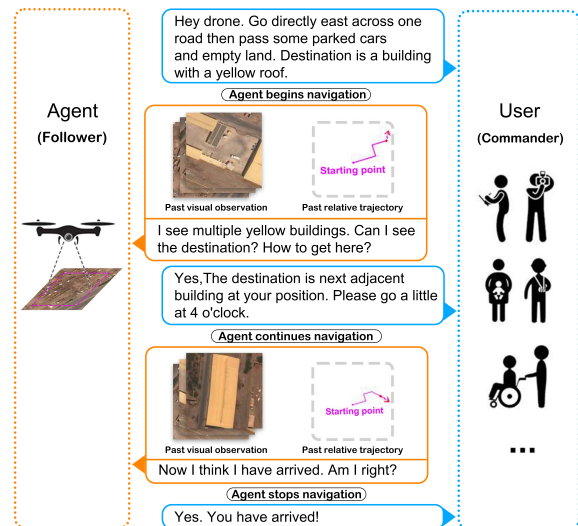

Figure 1: An example of Aerial Vision-and-Dialog Navigation (AVDN). The user instructs the agent to fly to a destination. During the navigation, the agent can ask questions while showing the images of past visual observations and relative trajectory. The user will talk back at a convenient time to provide further guidance to the agent without having to monitor the agent all the time.

drone control for users with some disabilities and who have their hands occupied by activities such as taking photos, writing, etc.

Therefore, this work introduces Aerial Vision-and-Dialog Navigation (AVDN), aiming to develop an intelligent drone that can converse with its user to fly to the expected destination. As shown in Figure 1, the user (commander) provides instructions, and the aerial agent (follower) follows the instruction and asks questions when needed. The past visual trajectories are also provided along with the question, which frees the commander from monitoring the drone all the time and minimizes the burden of drone control. In this free-form dialog, potential ambiguities in the instruction can be gradually resolved through the further instructions provided by the commanders upon request.

To implement and evaluate the AVDN task, we

build a photorealistic simulator with continuous state space to simulate a drone flying with its on-board camera pointing straight downward. Then we collect an AVDN dataset of 3,064 aerial navigation trajectories with human-human dialogs, where crowd-sourcing workers play the commander role and drone experts play the follower role, as illustrated in Figure 1. Moreover, we also collect the attention of human followers over the aerial scenes for a better understanding of where humans ground navigation instructions.

Based on our AVDN dataset, we introduce two challenging navigation tasks, Aerial Navigation from Dialog History (ANDH) and Aerial Navigation from Full Dialog History (ANDH-Full). Both tasks focus on predicting navigation actions that can lead the agent to the destination area, whereas the difference is that ANDH-Full presents the agent with full dialog and requires it to reach the final destination (Kim et al., 2021), while ANDH evaluates the agent's completion of the sub-trajectory within a dialog round given the previous dialog information (Thomason et al., 2020).

The proposed tasks open new challenges of sequential action prediction in a large continuous space and natural language grounding on photo-realistic aerial scenes. We propose a sequence-to-sequence Human Attention Aided Transformer model (HAA-Transformer) for both tasks. The HAA-Transformer model predicts waypoints to reduce the complexity of the search space and learns to stop at the desired location. More importantly, it is jointly trained to predict human attention from the input dialog and visual observations and learns where to look during inference. Experiments on our AVDN dataset show that multitask learning is beneficial and human attention prediction improves navigation performance. The main contributions are concluded as follows:

- We create a new dataset and simulator for aerial vision-and-dialog navigation. The dataset includes over 3K aerial navigation trajectories with human-human dialogs.

- We introduce ANDH and ANDH-Full tasks to evaluate the agent's ability to understand natural language dialog, reason about aerial scenes, and navigate to the target location in a continuous photorealistic aerial environment.

- We propose an HAA-Transformer model as the baseline for ANDH and ANDH-Full. Be-

sides predicting the waypoint navigation actions, HAA-Transformer also learns to predict the attention of the human follower along the navigation trajectory. Experiments on our AVDN dataset validate the effectiveness of the HAA-Transformer model.

2 Related work

Vision-and-Language Navigation Vision-and-Language Navigation (VLN) is an emerging multimodal task that studies the problem of using both language instructions and visual observation to predict navigation actions. We compare some of the works with our AVDN dataset in Table 1. Early VLN datasets such as Anderson et al. (2018); Ku et al. (2020) start with the indoor house environments in the Matterport3D simulator (Chang et al., 2017), where the visual scenes are connected on a navigation graph. To simulate continuous state change as in the real world, Krantz et al. (2020) built a 3D continuous environment by reconstructing the scene based on topological connections where the agent uses continuous actions during the navigation. Some other VLN studies focus on language instructions. Nguyen et al. (2019); Nguyen and Daumé III (2019); Thomason et al. (2020) created datasets where the agent can interact with the user by sending fixed signals or having dialogs. There are also works on synthetic indoor environments, such as Shridhar et al. (2020b); Padmakumar et al. (2021) that use an interactive simulation environment with synthetic views named ALFRED, where the agent needs to follow language instructions or dialogs to finish household tasks. Besides the indoor environment, some VLN datasets work on the more complex outdoor environment, such as the Touchdown dataset (Chen et al., 2019) and the modified LANI dataset (Misra et al., 2018). Blukis et al. (2019) is similar to ours for both using drones. However, the synthetic environment used has a gap from the realistic scene, and they ignored the control of the drone's altitude, where such navigation is oversimplified and has a large gap towards navigation in the real world in terms of language and vision aspects. Our work absorbs the advantage from previous works where we have continuous environments and dialog instructions to better approximate the real-world scenario.

Aerial Navigation Using both vision and language for aerial navigation is a less studied topic, whereas vision-only aerial navigation for drones is

Dataset	Env	Photorealistic	Continuous Space	Dialog	Free Form
R2R, RxR	indoor	✓	✗	✗	✓
VNLA, HANNA	indoor	✓	✗	✓	✗
VLN-CE	indoor	✗	✓	✗	✓
CVDN	indoor	✓	✗	✓	✓
ALFRED	indoor	✗	✗	✗	✓
DialFRED	indoor	✗	✗	✓	✗
TEACh	indoor	✗	✓	✓	✓
TouchDown	street-view	✓	✗	✗	✓
modified LANI	aerial	✗	✓	✗	✓
AVDN (ours)	aerial	✓	✓	✓	✓

Table 1: Example Vision-and-Language Navigation Datasets. R2R (Anderson et al., 2018), RxR (Ku et al., 2020), VNLA (Nguyen et al., 2019), HANNA (Nguyen and Daumé III, 2019), VLN-CE (Krantz et al., 2020), CVDN (Thomason et al., 2020), ALFRED (Shridhar et al., 2020a), DialFRED (Gao et al., 2022), TEACh (Padmakumar et al., 2021), TouchDown (Chen et al., 2019), modified LANI (Blukis et al., 2019).

already an active topic in the field. Some inspiring works (Loquercio et al., 2018; Giusti et al., 2015; Smolyanskiy et al., 2017; Fan et al., 2020; Bozcan and Kayacan, 2020; Majdik et al., 2017; Kang et al., 2019) worked on using pre-collected real-world drone data to tackle aerial vision navigation problems. Due to the hardness of collecting data and the risk of crashes, some other works applied simulation for aerial navigation (Chen et al., 2018; Shah et al., 2017; Chen et al., 2020), where rich ground truths are provided without the need for annotation. However, the modality of language is missing in these prior works and as a result, the navigation tasks only contain simple goals. As for the aerial vision-and-language navigation task in this work, the navigation is guided by natural dialog. As a result, it allows more diverse and complex navigation and also resolves ambiguities during complicated navigation.

3 Dataset

The AVDN dataset includes dialogs, navigation trajectories, and the drone's visual observation with human attention, where an example is shown in Figure 2. With the help of a newly proposed simulator, we record the AVDN trajectories created by two groups of humans interacting with each other, playing either the commander role or the follower role. Our AVDN dataset is the first aerial navigation dataset based on dialogs to the best of our knowledge.
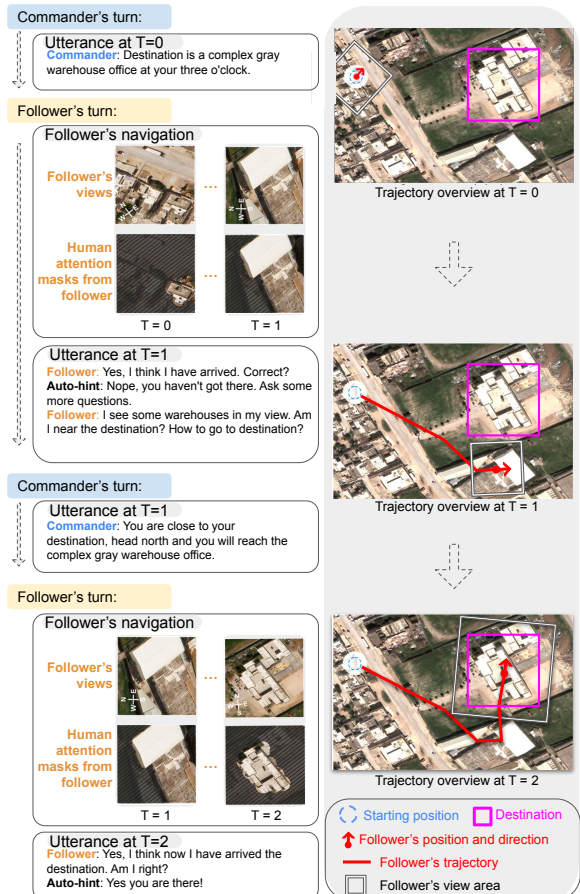

Figure 2: Example of a trajectory in our AVDN dataset. On the left, the commander's turn and the follower's turn alternate in chronological order. In each turn, dialog utterances are shown, and the follower's turn also shows the navigation process that spans from time step T to $T+1$, including the follower's observation and attention. On the right, there are trajectory overviews at different time steps. More examples can be found in the Appendix.

3.1 Simulator

We build a simulator to simulate the drone with a top-down view area. Our simulation environment is a continuous space so that the simulated drone can move continuously to any point within the environment. The drone's visual observations are square images generated corresponding to the drone's view area by cropping from high-resolution satellite images in the xView dataset (Lam et al., 2018), an open-source large-scale satellite image object detection dataset. In this way, our simulator is capable of providing continuous frames with rich visual features. We also design an interface for our simulator, where the simulated drone can be controlled with a keyboard and the drone's visual observation will be displayed in real-time with a digital compass. During the control, users can also

provide their attention over the displayed images on the interface by clicking the region they attend to. Last but not least, our simulator is capable of generating trajectory overviews, i.e. commander's view, showing the starting positions, destination areas, current view area and past trajectory (if exists) as in Figure 2.

3.2 Dataset Structure

In our AVDN dataset, each navigation trajectory includes time steps $T = 0, 1, \ldots, M$, where $M \geq 1$. At $T = 0$, an initial instruction is provided by the commander. Between adjacent time steps, there is a corresponding navigation sub-trajectory. At every time step of $0 < T < M$, there are questions from the follower and the corresponding answers from the commander. At $T = M$, the navigation trajectory ends because the destination area Des is reached and claimed by the follower. For details about when a trajectory ends, please refer to Section 3.3 Success Condition.

There are M follower's view area sequences $< u_0^T, u_1^T, \ldots, u_{N_T}^T >$, N_T is the length of T-th sequence, where the view area's center coordinate c_i^T always falls on the trajectory. Therefore, based on each view area, we could retrieve not only the simulated drone's location c_i, but also direction d_i and altitude h_i. Last but not least, for each view area u, there is a corresponding binary human attention mask with the same size. The area in u that corresponds to the white area on the mask is where the follower attended.

3.3 Dataset Collection

We collect our dataset with the help of Amazon Mechanical Turk (AMT) workers and drone experts, where AMT workers play the commander role to provide instructions and drone experts play the follower role to control a simulated drone and carry out the instruction. We pay the workers with wages no less than $15/h, and the data collection lasts for 90 days. We adopt an asynchronous data collection method, where the followers and commanders work in turns rather than simultaneously. This not only lowers the cost of data collection but also simulates how aerial vision-and-dialog navigation would work in practice, where the commanders will not monitor the follower's actions all the time.

Pipeline Before the start of data collection, we first sample objects in the xView dataset (Lam et al., 2018) as the destination areas and pair them with randomly selected initial follower's view areas within 1.5km distance. Then, using our simulator, we generate the trajectory overview at time step $T = 0$, as shown in Figure 2, which becomes the initial commander's view.

During data collection, the initial commander's view is presented to AMT workers for creating the initial instructions. We instruct the AMT workers to write instructions as if they are talking to a drone pilot based on the marked satellite images. Next, we let human drone experts play the follower role, i.e. controlling the simulated drone through our simulator interface, following the instructions and asking questions if they cannot find the destination area. When the experts stop the current navigation, they can either enter questions into a chatbox, claim the destination with a template sentence or reject the instruction for bad quality. If the destination is falsely claimed, the simulator will generate an auto-hint to let the follower ask some questions. For questions asked, AMT workers will provide further instructions accordingly based on given navigation information and dialog history. Then, the same drone experts will continue playing the follower role and again asking questions when necessary. We iterate the process until the destination is successfully reached and claimed by the follower.

Success Condition The navigation trajectory is successful only when the destination is reached at the time the follower claims it. We determine that the destination is reached in view area u_j by checking the center c_j and computing the Intersection over Union (IoU) between u_j and Des. If c_j is inside Des and the IoU of u_j and Des is larger than 0.4, the destination is regarded in u_j.

3.4 Data Analysis

Our AVDN dataset includes 3,064 aerial navigation trajectories, each with multi-round natural language dialog. There are two rounds of dialog on average per trajectory, where the number of dialog rounds in a trajectory equals to the maximum time step M. The most frequent words are shown in Figure 3a. The recorded AVDN trajectory path length has an average of $287m$, and its distribution is shown in Figure 3b. The trajectories and dialogs can be further separated into 6,269 sub-trajectories corresponding to the dialog rounds.

We split our dataset into *training*, *seen-validation*, *unseen-validation*, and *unseen-testing*

Split	#dialogs	#words per dialog	#areas	destination area-dim	#sub-paths	sub-path length
Training	2221	90	350	126m	4591	145m
Seen-val	197	79	197	120m	370	148m
Unseen-val	214	83	30	131m	411	144m
Unseen-test	432	91	65	117m	897	142m

Table 2: Dataset statistics. #dialogs is the number of dialogs, and #words per dialog is the average number of words in each dialog. #areas refers to the number of non-overlapped satellite images used. Destination area-dim is the average dimension of the sampled destination areas. #sub-paths is the number of sub-trajectories where each sub-trajectory corresponds to one round of dialog. Sub-path length is the average sub-trajectory length.

sets, where *seen* and *unseen* sets are pre-separated by making sure the area locations of the visual scenes are over $100km$ apart from each other. We show some statistical analysis across the dataset splits in Table 2. The visual scenes in our dataset come from the xView dataset (Lam et al., 2018), which covers both urban and rural scenes. The average covered area of the satellite images is $1.2km^2$.

Rather than providing a target hint in the beginning as in Thomason et al. (2020), the destination must be inferred from the human instructions given by the commander. For example, the commander may give a detailed description of the destination initially or write a rough instruction first and then describe the destination later in the dialog. We also find that there are two ways of describing the directions for navigation: egocentric direction description, such as "turn right", and allocentric direction description, such as "turn south". By filtering and categorizing words related to directions, we find that 82% of the dialog rounds use egocentric direction description and 30% of the dialog rounds include allocentric direction description. There are 17% dialog rounds that have mixed direction deceptions, making the instruction complex. This opens a new challenge for developing a language understanding module that can ground both the egocentric and allocentric descriptions to navigation actions.

4 Task

Following indoor dialog navigation (Thomason et al., 2020; Kim et al., 2021), we introduce an Aerial Navigation from Dialog History (ANDH) task and an Aerial Navigation from Full Dialog History (ANDH-Full) task based on our AVDN dataset and simulator.

(a) Frequent words   (b) Path length distribution

Figure 3: (a) displays the frequent words that appear in the dialogs and (b) shows the path length distribution of our AVDN dataset.

4.1 Aerial Navigation from Dialog History

The goal of the task is to let the agent predict aerial navigation actions that lead to goal areas G, following the instructions in the dialog history. Specifically, to predict one action $\hat{a}_j$ of an action sequence between navigation time step T_i and T_{i+1}, the inputs are dialogs from navigation time step 0 to T_i and images from a sequence of view areas $< \hat{u}_0, \hat{u}_1, \ldots, \hat{u}_{j-1} >$. A new view area $\hat{u}_j$ will be generated after $\hat{a}_j$ takes place.[1] The goal area G depends on the current navigation time step,

$$ G = \begin{cases} u_0^{T_{i+1}}, & \text{if } T_{i+1} \neq M \\ Des, & \text{otherwise} \end{cases}, \quad (1) $$

The predicted view area sequence will be recorded for evaluation with regard to the ground truth view area sequence $< u_0^{T_i}, \ldots, u_{N_{T_i}}^{T_i} >$.

4.2 Aerial Navigation from Full Dialog History

Compared with the ANDH task, the major difference of the ANDH-Full task is that it adopts the complete dialog history from navigation time step $T = 0, 1, \ldots, M$ as input. With the full dialog and visual observation, the agent needs to predict the full navigation trajectory from the starting view area u_0^0 to the destination area Des. ANDH-Full provides complete supervision for agents on a navigation trajectory with a more precise destination description and includes longer utterances and more complex vision grounding challenges.

4.3 Evaluation

Since the agent in both tasks, ANDH and ANDH-Full, needs to generate predicted view area sequences, the evaluation metrics for both tasks are the same. In the evaluation, the center points of every view area are connected to form the navigation trajectory, and the last view area is used to determine whether the predicted navigation successfully

---

[1]$\hat{u}_0$ is known as it is the initial view area at time step T_i.

leads to the destination area. The predicted navigation is successful if the IoU between the predicted final view area and the destination area is greater than 0.4. We apply several metrics for evaluation.

Success Rate (SR): the number of the predicted trajectory being regarded as successful, i.e., the final view area of the predicted trajectory satisfies the IoU requirement, over the number of total trajectories predicted.

Success weighted by inverse Path Length (SPL) (Anderson et al., 2018): measuring the Success Rate weighted by the total length of the navigation trajectory.

Goal Progress (GP) (Thomason et al., 2020): evaluating the distance of the progress made towards the destination area. It is computed as the Euclidean distance of the trajectory, deducted by the remaining distance from the center of the predicted final view area $\hat{c}_N$ to the center of goal area G.

5 Model

We proposed a Human Attention Aided (HAA-Transformer) model for the ANDH and ANDH-Full tasks as shown in Figure 4, where it takes as input multimodal information and generates multimodal predictions, including human attention prediction and navigation prediction.

Multimodal Encoding The input has three modalities, the drone's direction, images from the drone's visual observation, and history dialogs. At the start of a prediction series, our model uses a BERT encoder (Devlin et al., 2018) to get the language embeddings of the input dialog history, $h^l_{1:L}$, where special language tokens such as [INS] and [QUE] are added in front of each instruction and question in the dialog. Then, at every time step, all previous drone directions and images from the drone's visual observation are input to the model. A fully connected direction encoder is used to generate direction embeddings $h^x_{1:t}$ and an xView-pretrained Darknet-53[2] (Redmon and Farhadi, 2018) with an attention module is used to extract and flatten the visual features to get visual embeddings $h^v_{1:t}$. Finally, similar to the Episodic Transformer (Pashevich et al., 2021), all embeddings from the languages, images and directions, are concatenated and input into a multimodal transformer (F_{MT}) to produce output multimodal embeddings $\{z^l_{1:L}, z^v_{1:t}, z^x_{1:t}\}$ as in Equation 2.

$$\{z^l_{1:L}, z^v_{1:t}, z^x_{1:t}\} = F_{MT}(\{h^l_{1:L}, h^v_{1:t}, h^x_{1:t}\}) \quad (2)$$

Navigation Prediction and Waypoint Control The navigation outputs from our model come from a fully connected navigation decoder (F_{ND}) taking as input the transformer's output embeddings $\{z^l_{1:L}, z^v_{1:t}, z^x_{1:t}\}$ and generating predicted waypoint actions $\hat{w}$ and predicted navigation progress $\hat{g}$ as in Equation 3.

$$(\hat{w}, \hat{g}) = F_{ND}(\{z^l_{1:L}, z^v_{1:t}, z^x_{1:t}\}) \quad (3)$$

The predicted waypoint action $\hat{w}$ is a 3-D coordinate $(\hat{x}, \hat{y}, \hat{h})$, where $(\hat{x}, \hat{y})$ corresponds to an position in the current view area u and $\hat{h}$ corresponds to an altitude. The predicted waypoint also controls the drone's direction, where the direction is kept towards the direction of movement. Therefore, $\hat{w}$ controls the drone's movement, and as a result, the center, width and rotation of the next view area center are determined by $\hat{w}$. As for the navigation progress prediction $\hat{g}$, it is to generate a one-dimension navigation progress indicator for deciding when to stop (Xiang et al., 2019). If the predicted navigation progress is larger than a threshold, the drone navigation will be ended without executing the predicted waypoint action.

Human Attention Prediction A human attention decoder is proposed to predict the human attention mask using the output embeddings, $z^v_{1:t}$, from the multi-layer transformer that corresponds to the visual inputs. We build the decoder based on He et al. (2019), where the input to the decoder will be decoded to an $8 * 8$ representation through a fully connected layer and then linearly interpolated to a mask with the same shape as the input image. The greater the values in the mask means more likely the human follower attends the corresponding pixels.

Training We first train our HAA-Transformer model on the ANDH task and then fine-tuned it on the ANDH-Full task because the ANDH task is relatively easier with a shorter path length. For each task, we conduct the training alternately in teacher-forcing (Williams and Zipser, 1989) and student-forcing modes, where the main difference is whether the model interacts with the simulator using ground truth actions or the predicted actions. Our model is trained with a sum of losses from both navigation prediction and human attention prediction. First, the predicted waypoint action $\hat{w}$ and predicted navigation progress $\hat{g}$ are trained with Mean Square Error (MSE) loss, supervised by the ground truth w and g computed based on the recorded trajectory in our dataset. The naviga-
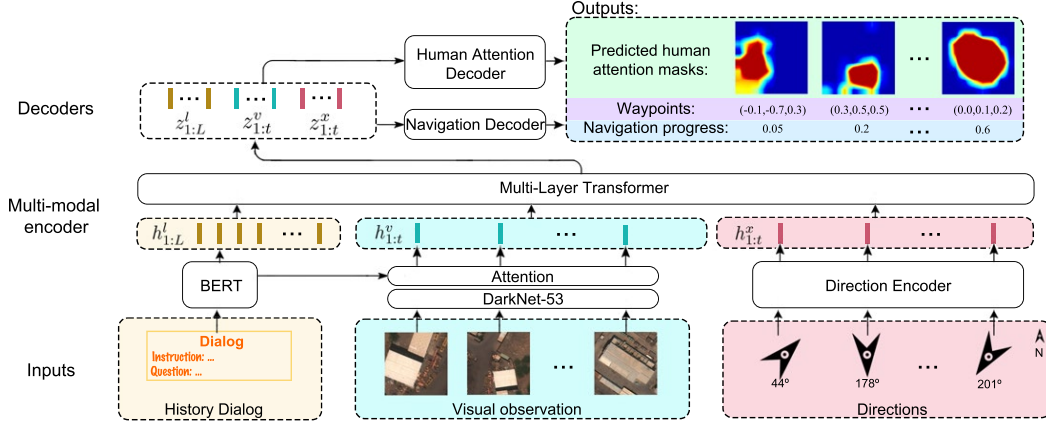
Figure 4: Our Human Attention Aided (HAA) model. The output of the model will interact with our simulator for generating the input for next time step.

tion prediction loss (L_{nav}) is shown in Equation 4, where $Rot(.)$ is computing the rotation change as a result of the waypoint action.

$$L_{nav} = MSE(Rot(\hat{w}), Rot(w)) \\ + MSE(\hat{w}, w) + MSE(\hat{g}, g) \quad (4)$$

Second, for human attention prediction training, we apply the modified Normalized Scanpath Score loss (NSS) (He et al., 2019). Given a predicted human attention mask P and a ground-truth human attention mask Q,

$$NSS(P, Q) = \frac{1}{N} \sum_i \overline{P_i} \times Q_i,$$

$$\text{where } N = \sum_i Q_i \text{ and } \bar{P} = \frac{P - \mu(P)}{\sigma(P)} \quad (5)$$

Since human attention may not exist in certain view areas, the human attention loss is only computed for view areas with recorded human attention.

6 Results

We conduct experiments to study our AVDN dataset and our HAA-Transformer model on the ANDH and ANDH-Full tasks.

Results on the ANDH task and ANDH-Full task As shown in Table 3, we evaluate our HAA-Transformer model along with multiple baseline models on both ANDH and ANDH-Full tasks. We first create a multimodal Episodic Transformer (E.T.) model (Pashevich et al., 2021) by removing the human attention decoder from our HAA-Transformer, and then build vision-only and language-only uni-modal models by ablating on the multimodal E.T. model. For uni-modal models, direction inputs are maintained while either vision input or language input is discarded. A multimodal LSTM-based model is also included as a sequence-

to-sequence baseline model, which has the same input and output as the multimodal E.T.model. All models, including our HAA-Transformer model are trained with random initialization. The batch size is 4 for the ANDH task, while for the ANDH-Full task. Based on the result, our HAA-Transformer model outperforms the baseline models in both tasks by a large margin. Also, compared with uni-modal baseline models and a random model outputting random waypoint actions, the multimodal E.T. model achieves overall higher performance, which indicates the importance of learning multimodal information in order to succeed in the ANDH task. Last but not least, we find that the language-only uni-modal model achieves much better performance than the vision-only uni-modal model showing that the language instructions play a more important role in guiding the navigation in our AVDN dataset.

Impact of Human Attention Prediction Training We then evaluate the impact of human attention prediction training for multimodal learning by ablation not only on our HAA-Transformer model but also on a Human Attention Aided Multimodal LSTM-based (HAA-LSTM) model developed by adding the human attention decoder module to the multimodal LSTM-based model (detailed in Appendix B). We apply the same human attention prediction training process and training loss as in our HAA-Transformer model. As the result shown in Table. 3, we find that the human attention prediction training significantly boosts both transformer-based and LSTM-based models across all evaluated metrics.

We further evaluate the benefit of human attention prediction training on different trajectory lengths. The sub-trajectories in the validation set

| | ANDH | | | | | | | | | ANDH-Full | | | | | | | | |
| | Seen Validation | | | Unseen Validation | | | Unseen Testing | | | Seen Validation | | | Unseen Validation | | | Unseen Testing | | |
Model	SPL↑	SR↑	GP↑	SPL↑	SR↑	GP↑	SPL↑	SR↑	GP↑	SPL↑	SR↑	GP↑	SPL↑	SR↑	GP↑	SPL↑	SR↑	GP↑
Random	0.5	1.6	-84.1	0.2	1.0	-81.4	0.5	1.1	-86.6	0.0	0.0	-176.6	0.0	0.0	-162.4	0.2	0.1	-158.4
Vision-only E.T.	3.1	3.8	0.2	6.5	7.3	0.2	3.2	3.9	0.2	0.0	0.0	-1.1	1.9	1.9	-5.2	0.2	0.2	-1.6
Language-only E.T.	9.2	11.6	48.9	13.9	17.0	50.6	9.7	12.7	49.1	1.4	2.0	54.8	2.3	3.7	54.5	1.8	2.2	58.2
E.T.	12.1	14.1	50.1	14.3	16.6	51.9	11.3	13.3	51.7	2.2	3.1	51.3	2.5	3.7	48.9	1.9	2.8	60.7
HAA-Transformer	**14.7**	**17.3**	**56.3**	**16.5**	**20.4**	**55.2**	**12.9**	**15.7**	**53.7**	**3.7**	**5.1**	**54.6**	**3.2**	**4.7**	**50.9**	**4.1**	**6.3**	**63.2**
LSTM	9.0	10.3	31.9	13.3	14.1	35.9	9.7	10.8	40.4	1.0	1.0	43.8	3.2	3.7	48.7	1.8	1.9	56.4
HAA-LSTM	**11.6**	**13.0**	**50.3**	**18.3**	**20.0**	**54.4**	**12.6**	**14.1**	**54.6**	**3.8**	**4.1**	**52.2**	**3.4**	3.7	**56.1**	**1.9**	**2.6**	**66.5**

Table 3: Main results on both ANDH and ANDH-Full tasks including ablation results on human attention prediction training. Both the Human Attention Aided Multi-modal LSTM (HAA-LSTM) model and our HAA-Transformer model are benefited from the human attention prediction training based on the performance comparison.

for ANDH task are split into four subsets based on the ground truth length. In Figure 5, we compare the number of successful sub-trajectory in different subsets among models with and without human attention prediction training. As a result, both our HAA-Transformer model and the HAA-LSTM model achieves significant performance improvements for subsets of longer trajectory. It leads to the conclusion that human attention prediction training benefits navigation prediction, especially for long trajectories for both two models that are based on LSTM and Transformer.

Besides improving task performance, human attention prediction also benefits the interpretability of the model by generating visualizable attention predictions paired with navigation predictions. We evaluate the human attention prediction result using the Normalized Scanpath Saliency (NSS) score, which measures the normalized saliency prediction at the ground truth human attention. Our HAA-Transformer model receives NSS scores of 0.84, 0.62 and 0.68, respectively, in seen validation, unseen validation, and test set, indicating the human attention prediction is effective.

Comparison for Different Input Dialog Length Comparing with the ANDH task, the ANDH-Full task requires the model to predict actions that correspond to longer dialogs with more dialog rounds. As a result, more challenges are involved and longer training time is needed compared with the results in the ANDH task. During training, we add a prompt of the drone's direction that corresponds to the dialog, e.g., "when facing east" to clarify the instructions in dialogs that happened in different time steps, especially when egocentric direction descriptions exist. In Table 4, we show our HAA-Transformer model's performance on trajectories with different dialog lengths, i.e. different numbers of dialog rounds, and we find the model's SR and SPL are diminished for trajectories with the num-
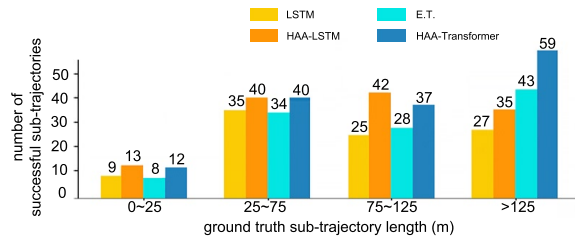

Figure 5: The impact of human attention prediction training on the success of trajectories of different lengths. Human attention prediction significantly improves navigation performance for longer trajectories.

| | Seen Validation | | Unseen Validation | | Unseen Testing | |
Dialog rounds	SPL↑	SR↑	SPL↑	SR↑	SPL↑	SR↑
1	3.7	4.7	1.7	2.2	4.3	7.0
2	**4.3**	**6.2**	**4.8**	**7.3**	**6.8**	**10.3**
≥ 3	2.7	4.3	3.8	6.1	1.6	2.0
All	3.7	5.1	3.2	4.7	4.1	6.3

Table 4: Result of our HAA-Transformer on ANDH-Full task regarding different dialog lengths. The more rounds in dialog, the longer the trajectory is and the more challenging the task is.

ber of dialog rounds less or greater than average, where they either containing too less or too much information. It shows a big room for improvement in understanding dialog with various lengths.

7 Conclusion

In this work, we introduce a dataset and a simulator for Aerial Vision-and-Language Navigation (AVDN). Challenging tasks are proposed based on our dataset focusing on navigation. A Human Attention Aided Multimodal Transformer (HAA-Transformer) model is designed for both tasks. Our work provides the possibilities for further studies to develop stronger models on AVDN that not only focus on navigation prediction but also on question generation. Furthermore, based on our results, future works may investigate using human attention prediction training to help solve VLN problems.

Limitation

This work proposed a dataset, a simulator, tasks, and models for Aerial Vision-and-Language Navigation. Since satellite images are needed to simulate the drone's observation, risks of privacy leaking may exist. By using the open-source satellite dataset xView (Lam et al., 2018), we mitigate the risks while also being able to develop a simulator for training our model. Additionally, using satellite images for simulating top-down visual observation of the drone introduces the shortcoming of having only 2D static scenes while adopting the strength of the satellite images where rich labels and visual features are included.

Broader Impact

We recognize the potential ethical problems during the dataset collection, where human annotators are involved. The data collection of this project is classified as exempt by Human Subject Committee vis IRB protocols. As a result, we utilized the Amazon Mechanical Turk (AMT) website to find workers willing to participate in the project. With AMT, our data collection is constrained by legal terms, and the data collection protocol is under AMT's approval. The agreement signed by both requesters and workers on AMT also ensures a transparent and fair data annotation process and that privacy is well protected.

References

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683.

Valts Blukis, Yannick Terme, Eyvind Niklasson, Ross A Knepper, and Yoav Artzi. 2019. Learning to map natural language instructions to physical quadcopter control using simulated flight. *arXiv preprint arXiv:1910.09664*.

Ilker Bozcan and Erdal Kayacan. 2020. AU-AIR: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance. *arXiv preprint arXiv:2001.11737*.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*.

Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12538–12547.

Lyujie Chen, Feng Liu, Yan Zhao, Wufan Wang, Xiaming Yuan, and Jihong Zhu. 2020. Valid: A comprehensive virtual aerial image dataset. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2009–2016. IEEE.

Lyujie Chen, Wufan Wang, and Jihong Zhu. 2018. Learning transferable uav for forest visual perception. *arXiv preprint arXiv:1806.03626*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yue Fan, Shilei Chu, Wei Zhang, Ran Song, and Yibin Li. 2020. Learn by observation: Imitation learning for drone patrolling from videos of a human navigator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5209–5216.

Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S Sukhatme. 2022. Dialfred: Dialogue-enabled agents for embodied instruction following. *arXiv preprint arXiv:2202.13330*.

Alessandro Giusti, Jérôme Guzzi, Dan C Cireşan, Fang-Lin He, Juan P Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, et al. 2015. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2):661–667.

Sen He, Hamed R Tavakoli, Ali Borji, Yang Mi, and Nicolas Pugeault. 2019. Understanding and visualizing deep visual saliency models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10206–10215.

Katie Kang, Suneel Belkhale, Gregory Kahn, Pieter Abbeel, and Sergey Levine. 2019. Generalization through simulation: Integrating simulated and real data into deep reinforcement learning for vision-based autonomous flight. *arXiv preprint arXiv:1902.03701*.

Hyounghun Kim, Jialu Li, and Mohit Bansal. 2021. Ndh-full: Learning and evaluating navigational agents on full-length dialogue. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6432–6442.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. 2020. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pages 104–120. Springer.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*.

Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. 2018. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*.

Antonio Loquercio, Ana I Maqueda, Carlos R Del-Blanco, and Davide Scaramuzza. 2018. Dronet: Learning to fly by driving. *IEEE Robotics and Automation Letters*, 3(2):1088–1095.

András L Majdik, Charles Till, and Davide Scaramuzza. 2017. The zurich urban micro aerial vehicle dataset. *The International Journal of Robotics Research*, 36(3):269–273.

Dipendra Misra, Andrew Bennett, Valts Blukis, Eyvind Niklasson, Max Shatkhin, and Yoav Artzi. 2018. Mapping instructions to actions in 3d environments with visual goal prediction. *arXiv preprint arXiv:1809.00786*.

Khanh Nguyen and Hal Daumé III. 2019. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. *arXiv preprint arXiv:1909.01871*.

Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. 2019. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12527–12537.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2021. Teach: Task-driven embodied agents that chat. *arXiv preprint arXiv:2110.00534*.

Alexander Pashevich, Cordelia Schmid, and Chen Sun. 2021. Episodic transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15942–15952.

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. 2017. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. *arXiv preprint arXiv:1705.05065*.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020a. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020b. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Nikolai Smolyanskiy, Alexey Kamenev, Jeffrey Smith, and Stan Birchfield. 2017. Toward low-flying autonomous mav trail navigation using deep neural networks for environmental awareness. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4241–4247.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2020. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jiannan Xiang, Xin Wang, and William Yang Wang. 2019. Not all actions are equal: Learning to stop in language-grounded urban navigation. In *ViGIL@ NeurIPS*.

A HAA-Transformer Model Details

There are around 120m parameters in our HAA-Transformer model. Our model uses a BERT$_{BASE}$ encoder (Devlin et al., 2018) with pretrained weights that open-sourced on Hugging Face (Wolf et al., 2020) to extract language feature of the input dialog history. For ANDH task, We extract two sets of language embeddings in ANDH task, where the input is either all the previous and current dialog rounds, or only the current dialog round for the target sub-trajectory. The language embeddings that include all previous dialog are used to attend to the image feature extracted by DarkNet-53 and flatten the feature to only 768 long per frame. The other with only current dialog is passed to the multi-model encoder. Whereas in ANDH-Full task, since the agent starts at an initial position with no previous dialog, only one set of language embeddings is extracted and used.

The attention modules that are used in our HAA-Transformer model and the HAA-LSTM model have the same structure. They generate soft attention based on dot-product attention mechanism. The inputs are context features and attention features. There is a fully connected layer before the output of the attention module. The context features attended by the attention features are concatenated with the attention features to become the input of the fully connected layer, and the output will be the attention module's output which has the same shape as the attention features.

A.1 Navigation Progress Prediction

As for the navigation progress prediction, we adopt the idea of L2Stop (Xiang et al., 2019) and create a navigation progress predictor to help decide when to stop, which overcomes the problem that the model would fail to stop at the desired position. The navigation progress is trained with the supervision of IoU score of the current view area $\hat{u}_{i,j,k}$ and the destination area. When the IoU is larger than 0, it indicates the designation area is seen in $\hat{u}_{i,j,k}$ and the larger the IoU the closer the $\hat{u}_{i,j,k}$ to the $des_{i,j}$. During the inference time, the predicted navigation stops when the generated navigation progress indicator is less than 0.5.

B HAA-LSTM Model

We also design a Human Attention Aided Multi-modal LSTM model for experiments in Section 6 as shown in Figure 6, where it takes the same input
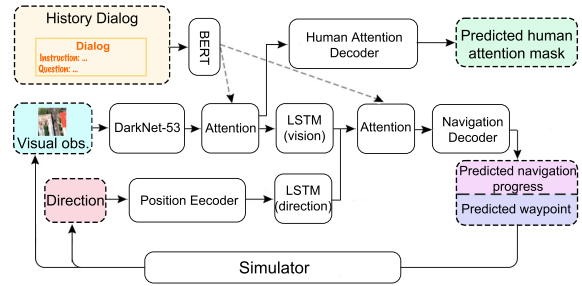

Figure 6: Human Attention Aided Multi-modal LSTM (HAA-LSTM) model which uses the same input and output as our HAA-Transformer.
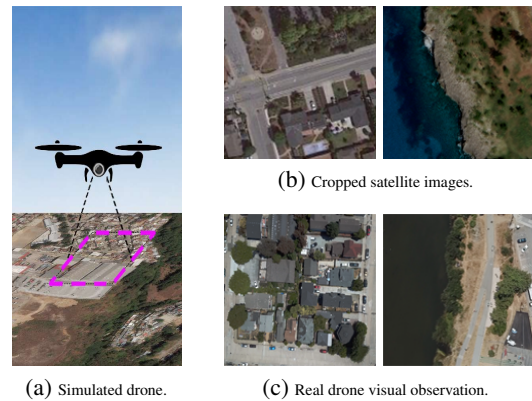

(b) Cropped satellite images.

(a) Simulated drone. (c) Real drone visual observation.

Figure 7: (a) shows how simulated drone's visual observation is generated from satellite images in our simulator. We compare the simulated drone's visual observation from satellite images, (b), with images from a drone's onboard camera at about 200m above ground, (c).

and output as our HAA-Transformer model. We also add the same human attention decoder as in our HAA-Transformer model for human attention prediction training. The language embeddings, visual observation and direction embeddings are also extracted in the same way.

C Training Details

We train all models on one Nvidia RTX A6000 graphic card. We train all baseline models as well as both HAA-Transformer model and HAA-LSTM model for approximately 150k iterations on ANDH task with batch size being 4 and learning rate being 1e-5. For the ANDH-Full task, since it uses full dialog history as input, where more GPU RAM is needed, we use a batch size of 2 and learning rate of 5e-6 and train the model for 200k interactions which take about 48 hours.

D Simulator Details

We design a simulator to simulate a drone flying with its onboard camera facing straight downward, as in Figure 7a. The simulator uses satellite images from xView dataset (Lam et al., 2018) for the drone's visual observation, where the observation is square image patches cropped from the satellite image based on the drone's view area, as in Figure 7b. We argue that by using satellite images, our simulator is capable of providing equally rich visual features as in the real world and some examples are shown in Figure 7c. Additionally, since satellite images have boundaries that are not adjacent with each other, we prevent the drone's view area from moving out of boundary by automatically invalidate the drone's action that will lead to out-of-boundary view areas. Further more, for simplicity, we assume perfect control of the drone's movement, and therefore, the drone's current view area is determined by the previous drone's position and navigation action.

During the dataset collection, the follower controls the simulated drone through the simulator interface with keyboards. We defined 8 keys for the control with a total of four degrees of freedoms (DoFs), where there are 2 DoFs for horizontal movement, 1 DoF for altitude control, and 1 DoF for rotation control. Despite that our simulator environment is continuous, the control through the interface is discrete for an easier control experience. Every time a key is pressed, the simulated drone will move along the DoF for a fixed distance and the higher the simulated drone flies, the faster it moves with one press of the keyboard. Before the follower presses ESC key to stop the control, he/she can also generate the human attention data by using the mouse to left-click on the attended image region shown on the interface. After every left-click, a circle with a radius being 1/10 of the current view area width will become the attended region and be displayed on the interface. Also, a right-click on the circle will remove this region from the attention record.

E Dataset Details and Examples

We provide some details about our dataset with related examples. Each example includes a dialog, sample drone's visual observation with human attention and navigation overviews.
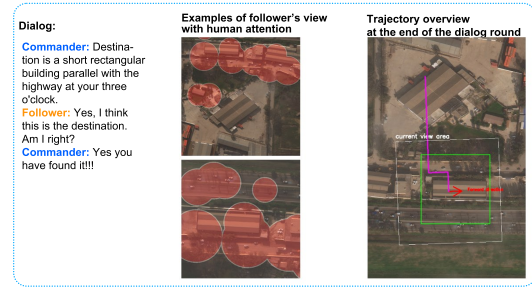

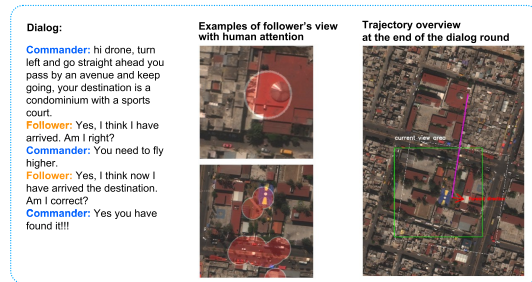Figure 8: Example of a trajectory with one dialog round.

Figure 9: Example of a trajectory that includes auto-instruction about altitude adjustment. There is only one dialog round.

E.1 Human Attention

We record the attention from the follower through our simulator interface when the follower is controlling the simulated drone. In each navigation trajectory collected, the attention are stored in a list where the order of the list is ignored, meaning that the attended areas either recorded earlier or later during the navigation will be retrieved together when using the human attention data. In this way, the human attention data becomes more accurate since the area that followers missed to attend in the current view area is likely to be included in the future time steps. Also, because the previously attended area is kept in later view areas, less effort is needed to annotate the attended areas. We find that 1/7 of the area on average is attended to in the recorded view areas $u_{i,j}$.

E.2 Dialog Structure

The dialogs contained in our AVDN datset have a various number of rounds. Since the dialog rounds are split based on the data collection rounds, each dialog round contains only one instruction written by the commander. Figure 8 shows an example of a simple dialog with only one dialog round. However, when the follower can not follow the initial instruction to find the destination area, questions will be brought up, and therefore more dialog

Dialog:

Commander: Head straight ahead, your destination is in a segregated area.
Follower: Where should I go after I go straight? How does the destination look like?
Commander: Proceed forward direction. The nearest by brown colour building is in your destination.
Follower: Yes, I think this is the destination. Is it correct?
Commander: Nope, you haven't get there. Ask some more questions.
Follower: I have already seen the brown building, how to go to destinaton?
Commander: Go south to next building.
Follower: Yes, I think this is the destination. Am I right?
Commander: Yes you have found it!!!

Figure 10: Example of a trajectory with three dialog rounds. There is an incorrect instruction in the second dialog round, where the destination should be described as the second nearest brown building rather than the nearest one. For this case, since the instruction is clear and can be followed by the follower, we treat it as an inevitable and acceptable type of instruction with mistakes and keep it in our dataset.

rounds will be introduced. Every dialog rounds start with the instruction from human commanders and could include one or more utterance from the follower, depending on if auto-instructions exist. We provide details about auto-instructions in the next sub-section. Also, when followers are writing the questions, we enable them to define some shortcut keys for frequently used general questions such as "could you further explain it?", "where should I go?", etc. To avoid templated dialogs, followers are forbidden to only use the shortcut for the question but need to incorporate their own language.

E.3 Auto-instructions

When the follower claims that the destination is reached, our simulator will check the navigation result automatically using the success condition described in Section 3.3. Then, auto-instructions will be generated based on whether the destination area is reached successfully. Specifically, when the success condition is met, an auto-instruction of "Yes, you have found it!!!" will be added to the dialog as the end; if the destination is in the center of the view area, but the view area is either too large or too small, failing the success condition, the simulator will also provide auto-instructions asking the follower to adjust the drone's altitude and verify again if the success conditions are met or not, as shown is Figure 9.

E.4 Dialog Quality

To ensure the dialogs in our dataset have good quality, we make efforts during the data collection process and conduct extra examination for the dialog data after the data collection.

(a) Words from commander utterances

(b) Words from follower utterances

Figure 11: Counts of top 50 most frequently used words in commander and follower utterances.

3055

During the data collection, online workers from Amazon Mechanical Turk (AMT) are playing as commanders and provide instructions in the dialog, who, compared with the follower that we hired to work on-site and supervised by us in-person, have a higher chance of generating low quality and incorrect language instructions. We develop some strategies to deal with these undesired instructions. First, if the follower, guided by the instruction, lets the drone navigate to a direction that is more than 90 degrees different from the ground truth direction of the destination area, our simulator will automatically label the instruction as incorrect. Those labeled instructions will be discarded and collected again. Then, since the follower needs to read and understand the instructions, they have the chance to report the instructions as being low-quality or incomprehensible and skip them. Finally, in the remaining instructions that are not spotted as low-quality or incorrect, it is still possible that instructions are not accurate or incorrect due to human mistakes from the AMT workers, such as in Figure 10. By manually checking the dialogs and navigation trajectories in randomly selected subsets of our AVDN dataset, we spot only 5 instructions with potential mistakes in 50 dialogs. In those cases, because the follower successfully followed the instruction, we keep those instructions unchanged even if they didn't help guide the follower to find the destination area. In the real world, the user in AVDN could also make mistakes, so this mistake tolerance strategy makes our dataset even closer to real scenarios.

We further examine the dialog quality after the data collection by analyzing the dialogs. The average utterance (human-written instructions and questions) in a dialog is 3.1, with a minimum and maximum being 1 and 7 because each dialog includes at least one instruction written by a human. The average number of words written by commander and follower are 45 and 19, and there are about 15 words from auto-instructions. Also, in Figure 11, we show the distribution of the top 30 most frequent words in the commander's and follower's utterances. The results show a smooth variance across nouns, verbs, adjectives, and prepositions, indicating that our dataset's utterances have rich contents and good variety. Last but not least, we manually checked the dialogs in all validation and test sets by visualizing the corresponding navigation trajectory and the dialog, and we observed no

major issue.

F Interface for workers in dataset collection

We use help from Amazon Mechanical Turk (AMT) workers and human drone experts during the collection of our Aerial Vision-and-Language Navigation (AVDN) dataset, where the AMT workers play the commander role providing instructions the drone experts play the follower role asking questions and controlling the drone. In this section, we demonstrate the interface for both groups of workers with all the information they receive in the data collection procedure.

F.1 Interfaces for commanders

There are two interfaces for commanders (AMT workers) depending on which data collection round it is. The interface includes one trajectory each time and contains all the information needed for the commander to create the instruction. Detailed and step-by-step instructions for what needs to be done as a commander are introduced at the beginning of the interface. The AMT workers need to write sentences in the *Answer* according to the provided information.

In the first round of data collection, the commander needs to write the initial instruction based on an overview of the AVDN trajectory. As shown in Fig. 12 the satellite image shows the trajectory overview marked with a predefined staring position (the red point with an arrow showing the drone's direction at the starting position) and a destination area (purple bounding box).

In the data collection round after the first round, the commander is required to give follow-up instructions, i.e., answers, to the questions from the follower. The user interface for the second and following rounds is shown in Fig. 13. Besides all the information shown to the commander in the first round, the follower is also provided with previous dialog, past trajectories (broken purple line), and the view area corresponding to the most recent time step (named current view area marked with white bounding box).

F.2 Interface for followers

The follower uses an interface to interact with our simulator. In our simulator, they receive instructions from the commander and control the simulated drone. The keyboard is used to simulate

Please read and follow the following requirements carefully, if your answer will be rejected if it is wrong or has poor quality (cannot be followed).

Background:
- Please describe **how** to get to the destination (purple box) from the current view center (a red arrow pointing in the forward direction), and **what** the destination looks like.

Steps:
Step 1: find the destination (purple box) in the image.

Step 2: figure out the <u>orientation</u>. The image is taken by a flying drone facing towards the direction as the red arrow.
-- You may use the clock position of the <u>drone's perspective</u>, where 12 o'clock is the "forward direction".
-- You may use the compass direction, and the up is North.
-- You may use words such as left, right, forward, and backward.

Step 3: write some sentences as the instruction for the drone. It should include:
-- <u>What direction to go? How to get to the destination? What the destination is?</u>
-- <u>What landmarks will you pass by, eg. roads, trees, building blocks.</u>
-- **(important)** What is the destination (maybe a house, a building, a hut, etc.) and what does it look like (color, shape, etc.).

Some examples: Here are some examples of how the instructions may look like. Your answer may include but is not limited to them.
- Hi drone, head forward towards your ... o'clock direction. After passing ... the destination is a building surrounded by ...
- Could you please go northeast for ... blocks until seeing ... Then go ... a little, the destination is a ...
- You need to get to the nearest road at 6 o'clock for ... Follow the road ... for ... the destination is ...

Attention
- Up is North.
- Give instruction from the drone's perspective, so "Forward direction" is 12 o'clock direction.
- Do not mention the image boundary.
- Do not mention the annotation marks (such as purple box)

Answer:

Figure 12: Interface for AMT workers (commanders) in first round of data collection.

the drone controller with eight keys representing four channels in the controller, where key *w* and *s* represent the channel controlling forward and backward movement, key *a* and *d* represent the channel controlling left and right movement, key *q* and *e* represent the channel controlling rotating clockwise and anti-clockwise movement and key *1* and *2* represent the channel controlling altitude change. After the experts finish the control, the commander can either claim the destination is reached or ask questions for more instruction. As in Fig. 14, the interface is an image window showing the simulated drone's visual observation and a text window for displaying the previous dialogs and inputting questions from the follower. There is a compass on the top left of the image window, showing the orientation of the simulated drone. The red cross in the image window shows the center of the view, helping the follower control the drone to right above the destination area, and the red corners in the window show the area of 0.4 IoU with the view area. The follower is instructed to make the destination area larger than the area indicated by the red corners in order to finish successful navigation.

Please read and follow the following requirements carefully, if your answer will be rejected if it is wrong or has poor quality (cannot be followed).

Background:
- The image below is a map showing the flying trajectory (purple line) of a drone. The drone was guided by a conversation (shown below).
- Assume the drone **now can only see the area inside the "current view area" (white bounding box)**.
- The conversation is unfinished. You need to answer some questions, as part of the conversation.
- Describe the how the destination (purple box) looks like and guide the drone to it.

Steps:
Step 1: In the image below, find:
- the destination (purple circle).
- the current drone location (a red arrow). The red arrow is pointing in the forward direction of the drone.
- the drone's current view area (white bounding box).

Step 2: figure out the orientation. The forward direction of the helicopter is where the red arrow is pointing.
-- You may use clock position, forward direction is the drone's 12 o'clock direction.
-- You may use the compass to tell the orientation. Up is north by default.
-- You may use words such as left, right, forward and backward from the drone's perspective.

Step 3: Read the previous conversations and continue the conversation by answering the following questions.

Example of your goal: (help to make the situation change from left to right)

Unfinished drone flying path   Finished drone flying path
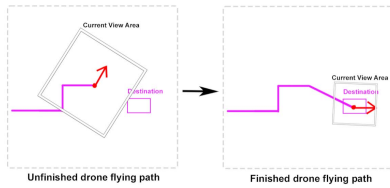
Coversation:

Previous Dialog:
- Instruction: Destination is a small gray-roofed building at your seven o'clock.
- Question:Yes, I think I have arrived.
- Auto-instruction: Nope, you haven't got there. Ask some more questions.
- Question: I am on top of a grey building. Am I near the destination? Where should I go now?
- Instruction: Proceed forward direction and turn right in cross the trees and nearest by grey colour building in your destination
- Question:Yes, I think I have arrived.
- Auto-instruction: Nope, you haven't got there. Ask some more questions.
- Question: I am on the grey building after passing the trees, How does the destination look like? In what direction is our destination?
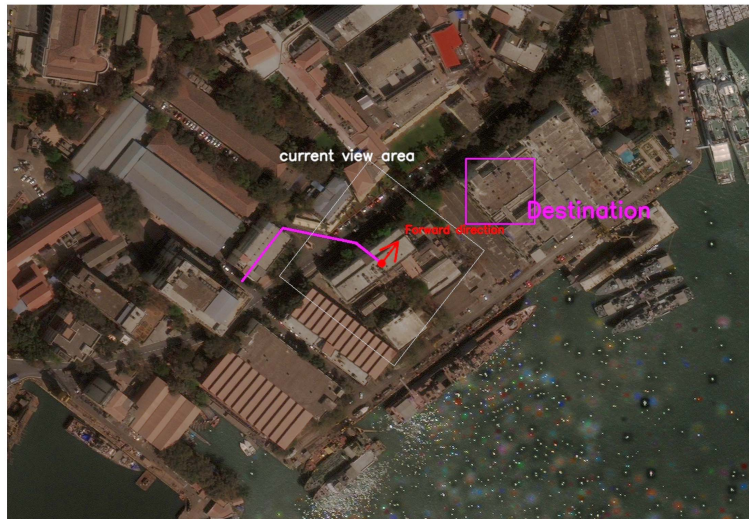
Question:
- I am on the grey building after pass the trees, How does the destination look like? In what direction is our destination?

Answer:

Attention
- Answer every question if multiple questions are asked.
- Describe what the destination looks like.
- Do not mention the annotation marks (such as purple box), instead, use "destination" or "goal", etc.

Submit

Figure 13: Interface for AMT workers (commanders) in following rounds of data collection.

Figure 14: Interface for human drone experts (follower). The upper window shows the simulated drone's visual observation and the lower window shows the previous dialog.

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section of Limitation.

☑ A2. Did you discuss any potential risks of your work?
Section of Broader Impact.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 3, Section 5 and Appendix A.1

☑ B1. Did you cite the creators of artifacts you used?
Section 3, Section 5 and Appendix A.1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 3, Section 5 and Appendix A.1

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 3, Section 5 and Appendix A.1

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Appendix E.4

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3.2.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3.3

C ☑ Did you run computational experiments?

Section 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5 and Appendix C

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 6

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 6

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4.3 and Appendix A

D ☑ **Did you use human annotators (e.g., crowdworkers) or research with human participants?**

Appendix 3.3

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Section F

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section 3.3

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Section 3.3 and Appendix F

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. The data collection part of this project is classified as exempt by Human Subject Committee vis IRB protocols.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Amazon Mechanical Turk website for collecting commander data, we have no access to the workers' information.