

Seeing is Believing (and Predicting): Context-Aware Multi-Human Behavior Prediction with Vision Language Models

Supplementary Material

A. Metric Computation

A.1. Accuracy Score (Acc)

$$\text{Acc} = \frac{2 \times |P \cap G|}{|P| + |G|} \quad (1)$$

where P and G are the sets of predicted and ground-truth interaction labels at the same future time step, respectively. It reports the label overlap between P and G , considering both exact and partial matches, and returning a value between 0 and 1, where 1 means fully matched.

A.2. Cosine Similarity (CS)

$$\text{CS} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \quad (2)$$

where \mathbf{A} and \mathbf{B} denote the vector representations of the text strings, correspondingly. It reveals the semantic similarity of two phrases and produces a value between -1 and 1 , where 1 infers identical meaning, and -1 the opposite.

A.3. Edit Distance (ED)

$$d_{\text{ed}}(P, G) = \min \begin{cases} d_{\text{ed}}(P_{1:m-1}, G_{1:n}) + 1 & (\text{del.}) \\ d_{\text{ed}}(P_{1:m}, G_{1:n-1}) + 1 & (\text{ins.}) \\ d_{\text{ed}}(P_{1:m-1}, G_{1:n-1}) + 1_{P_m \neq G_n} & (\text{sub.}) \end{cases} \quad (3)$$

where $P_{1:m}$ and $G_{1:n}$ refer to the first m and n characters of the prediction P and the ground truth G , respectively. The value varies from 0 to 1, where 0 indicates a full match.

B. Training Hyperparameter

In this paper, we mainly focus on fine-tuning the *Qwen2-VL-72B-Instruct* variant. The hyperparameters of the Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) stages are listed below in Table 1 and 2:

C. Action Scripts for Dataset Generation

This section describes the action programs used to generate the synthetic human behavior videos using the VirtualHome Simulator [1]. The scripts are programmed using the format pre-defined by VirtualHome, i.e., $\langle \text{char_id} \rangle [\text{action}] \langle \text{object} \rangle \langle \text{object_id} \rangle$, where:

Hyperparameter	Value
Epochs	30
Learning Rate	3e-4
Learning Rate Scheduler	Constant
Optimizer	AdamW (Torch Fused)
LoRA Rank (r)	64
LoRA Alpha (α)	128
Maximum New Tokens	512
Temperature	1.0
Gradient Accumulation Steps	8
Precision	bfloat16 (bf16)
Hardware	3× Nvidia H100

Table 1. Training hyperparameters for SFT stage

Hyperparameter	Value
Epochs	5
Learning Rate	1e-6
Optimizer	AdamW
Maximum New Tokens	512
Temperature	1.0
Gradient Accumulation Steps	8
Precision	bfloat16 (bf16)
Hardware	2× Nvidia H100

Table 2. Training hyperparameters for DPO stage

- $\langle \text{char_id} \rangle$ corresponds to the id of the supported character [3], e.g., *male1*, *female1*, etc.
- $[\text{action}]$ can be chosen from the supported actions that fulfill the preconditions [2].
- $\langle \text{object} \rangle$ can be chosen from the supported objects pre-defined in the simulator [4].
- $\langle \text{object_id} \rangle$ specifies the object instance in the environment to interact with. It may vary in different VirtualHome environments.

The following scripts are used across 6 different VirtualHome environments, spanning three primary domestic scenarios: kitchen, living room, and bedroom, each with unique spatial configurations and object layouts. For each environment, the $\langle \text{object_id} \rangle$ in the script should be replaced by the corresponding id, which can be queried from the environment graph [5]. The scene graphs used in the dataset can also be derived using the



Figure 1. Example frames of kitchen script 1



Figure 2. Example frames of kitchen script 2



Figure 3. Example frames of kitchen script 3

environment_graph() API and converted into the previously described format.

C.1. Scripts for Kitchen Scenarios

Script 1: This is an example of two humans in a kitchen, one heats up food using a microwave, and the other drinks a milkshake, as shown in Fig. 1.

```
-<char0> [walk] <salmon> (salmon.id) | <char1>
[walk] <kitchentable> (kitchentable.id),
-<char0> [grab] <salmon> (salmon.id) | <char1>
[grab] <milkshake> (milkshake.id),
-<char0> [open] <microwave> (microwave.id) |
<char1> [drink] <milkshake> (milkshake.id)",
-<char0> [putin] <salmon> (salmon.id)
<microwave> (microwave.id) | <char1> [walk]
<sink> (sink.id)",
-<char0> [close] <microwave> (microwave.id)
| <char1> [putback] <milkshake> (milkshake.id)
<sink> (sink.id)",
-<char0> [open] <microwave> (microwave.id) |
<char1> [switchon] <faucet> (faucet.id)",
-<char0> [grab] <salmon> (salmon.id) | <char1>
[switchoff] <faucet> (faucet.id)",
-<char0> [close] <microwave> (microwave.id) |
<char1> [grab] <milkshake> (milkshake.id)",
-<char0> [walk] <kitchentable>
(kitchentable.id) | <char1> [walk]
<kitchentable> (kitchentable.id)",
-<char0> [put] <salmon> (salmon.id)
```

```
<kitchentable> (kitchentable.id) | <char1>
[put] <milkshake> (milkshake.id) <kitchentable>
(kitchentable.id)",
-<char0> [grab] <glass> (glass.id)",
-<char0> [walk] <sink> (sink.id)",
-<char0> [putback] <glass> (glass.id) <sink>
(sink.id)",
-<char0> [switchon] <faucet> (faucet.id)",
-<char0> [switchoff] <faucet> (faucet.id)",
-<char0> [grab] <glass> (glass.id)",
-<char0> [walk] <kitchentable>
(kitchentable.id)",
-<char0> [put] <glass> (glass.id)
<kitchentable> (kitchentable.id) "
```

Script 2: This is an example of two humans in a kitchen, where one is cooking and another is interacting with objects on the kitchen table, as shown in Fig. 2.

```
-<char0> [walk] <fryingpan> (fryingpan.id)
| <char1> [walk] <cutleryknife>
(curleryknife.id)",
-<char0> [grab] <fryingpan> (fryingpan.id)
| <char1> [grab] <cutleryknife>
(curleryknife.id)",
-<char0> [put] <fryingpan> (fryingpan.id)
<stove> (stove.id) | <char1> [walk] <sink>
(sink.id)",
-<char0> [walk] <salmon> (salmon.id) | <char1>
```



Figure 4. Example frames of living room script 1



Figure 5. Example frames of living room script 2

```
[put] <cutleryknife> (curleryknife.id) <sink>
(sink.id)",
-"<char0> [grab] <salmon> (salmon.id) | <char1>
[walk] <cutleryfork> (curleryfork.id)",
-"<char0> [walk] <fryingpan> (fryingpan.id) |
<char1> [grab] <cutleryfork> (curleryfork.id)",
-"<char0> [put] <salmon> (salmon.id)
<fryingpan> (fryingpan.id) | <char1> [walk]
<sink> (sink.id)",
-"<char0> [switchon] <stove> (stove.id) |
<char1> [put] <cutleryfork> (curleryfork.id)
<sink> (sink.id)",
-"<char0> [switchoff] <stove> (stove.id) |
<char1> [walk] <plate> (plate.id)",
-"<char1> [grab] <plate> (plate.id)",
-"<char1> [walk] <sink> (sink.id)",
-"<char1> [put] <plate> (plate.id) <sink>
(sink.id)",
-"<char1> [switchon] <faucet> (faucet.id)",
-"<char1> [grab] <dishwashingliquid>
(dishwashingliquid.id)",
-"<char1> [put] <dishwashingliquid>
(dishwashingliquid.id) <sink> (sink.id)",
-"<char1> [switchoff] <faucet> (faucet.id)",
-"<char1> [open] <kitchencabinet>
(kitchencabinet.id)",
-"<char1> [grab] <plate> (plate.id)",
-"<char1> [put] <plate> (plate.id)
<kitchencabinet> (kitchencabinet.id)",
-"<char1> [grab] <cutleryknife>
(curleryknife.id)",
-"<char1> [put] <cutleryknife>
(curleryknife.id) <kitchencabinet>
(kitchencabinet.id)",
-"<char1> [grab] <cutleryfork>
(curleryfork.id)",
-"<char1> [put] <cutleryfork> (curleryfork.id)
<kitchencabinet> (kitchencabinet.id)"
```

Script 3: This is an example of three humans in a kitchen, where one human is washing the dishes using a dishwasher, one is making coffee, and another human is putting items

inside the refrigerator, as shown in Fig. 3.

```
-"<char0> [walk] <kitchentable>
(kitchentable.id)",
-"<char0> [grab] <plate> (plate.id) | <char1>
[grab] <mug> (mug.id)",
-"<char0> [walk] <dishwasher> (dishwasher.id) |
<char1> [walk] <coffeemaker> (coffeemaker.id)
| <char2> [walk] <kitchentable>
(kitchentable.id)",
-"<char0> [open] <dishwasher> (dishwasher.id)
| <char1> [put] <mug> (mug.id) <coffeemaker>
(coffeemaker.id) | <char2> [grab] <cupcake>
(cupcake.id)",
-"<char0> [putin] <plate> (plate.id)
<dishwasher> (dishwasher.id) | <char1> [lookat]
<coffeemaker> (coffeemaker.id) | <char2> [walk]
<fridge> (fridge.id)",
-"<char0> [walk] <stove> (stove.id) | <char1>
[lookat] <coffeemaker> (coffeemaker.id) |
<char2> [open] <fridge> (fridge.id)",
-"<char0> [grab] <cookingpot> (cookingpot.id) |
<char1> [lookat] <coffeemaker> (coffeemaker.id)
| <char2> [putin] <cupcake> (cupcake.id)
<fridge> (fridge.id)",
-"<char0> [putin] <cookingpot> (cookingpot.id)
<dishwasher> (dishwasher.id) | <char2> [walk]
<kitchentable> (kitchentable.id)",
-"<char0> [close] <dishwasher> (dishwasher.id)
| <char2> [grab] <cupcake> (cupcake.id)",
-"<char0> [switchon] <dishwasher>
(dishwasher.id) | <char2> [putin] <cupcake>
(cupcake.id) <fridge> (fridge.id)",
-"<char0> [switchoff] <dishwasher>
(dishwasher.id) | <char1> [grab] <mug> (mug.id)
| <char2> [close] <fridge> (fridge.id)",
-"<char0> [open] <dishwasher> (dishwasher.id)
| <char1> [walk] <kitchentable>
(kitchentable.id)",
-"<char0> [grab] <plate> (plate.id) | <char1>
[drink] <mug> (mug.id)",
-"<char0> [open] <kitchencabinet>
(kitchencabinet.id) | <char1> [walk] <sink>
```




Figure 6. Example frames of bedroom script 1



Figure 7. Example frames of bedroom script 2

```
(sink.id)",
-<char0> [putin] <plate> (plate.id)
<kitchencabinet> (kitchencabinet.id) | <char1>
[put] <mug> (mug.id) <sink> (sink.id)",
-<char0> [grab] <cookingpot> (cookingpot.id) |
<char1> [switchon] <faucet> (faucet.id)",
-<char0> [close] <dishwasher>
(dishwasher.id)",
-<char0> [putin] <cookingpot> (cookingpot.id)
<kitchencabinet> (kitchencabinet.id)",
-<char0> [close] <kitchencabinet>
(kitchencabinet.id) "
```

C.2. Scripts for Living Room Scenarios

Script 1: This is an example of a living room scenario, where one person is watching television and another person is working on a computer, as shown in Fig. 4.

```
-<char0> [walk] <tv> (tv.id) | <char1> [walk]
<computer> (computer.id)",
-<char0> [switchon] <tv> (tv.id) | <char1>
[switchon] <computer> (computer.id)",
-<char0> [walk] <coffeetable>
(coffeetable.id)",
-<char0> [grab] <milkshake> (milkshake.id) |
<char1> [walk] <chair> (chairs[-1].id)",
-<char0> [walk] <sofa> (sofa.id) | <char1>
[sit] <chair> (chairs[-1].id)",
-<char0> [sit] <sofa> (sofa.id)",
-<char0> [drink] <milkshake> (milkshake.id) "
```

Script 2: This is an example script of three humans in a living room scenario, where one human is arranging folders on a bookshelf, one human is working at a computer, and one human is watching television, as shown in Fig. 5.

```
-<char0> [walk] <coffeetable>
(livingroomcoffeetable.id) | <char1> [walk]
<coffeetable> (livingroomcoffeetable.id) |
```

```
<char2> [walk] <tv> (tv)",
-<char0> [grab] <folder> (folder.id) | <char1>
[grab] <mug> (mug.id)",
-<char0> [walk] <bookshelf> (bookshelf.id)
| <char1> [walk] <computer> (computer.id) |
<char2> [switchon] <tv> (tv)",
-<char0> [putin] <folder> (folder.id)
<bookshelf> (bookshelf.id) | <char1> [switchon]
<computer> (computer.id) | <char2> [walk]
<sofa> (sofa.id)",
-<char0> [walk] <coffeetable>
(livingroomcoffeetable.id) | <char1> [walk]
<chair> (chair.id) | <char2> [sit] <sofa>
(sofa.id)",
-<char0> [grab] <book> (book.id) | <char1>
[sit] <chair> (chair.id)",
-<char0> [walk] <sofa> (sofa.id)",
-<char0> [sit] <sofa> (sofa.id) "
```

C.3. Scripts for Bedroom Scenarios

Script 1: This is an example of two humans in a bedroom scene, where one human is arranging clothes in a closet and another human reads a book, as shown in Fig. 6.

```
-<char0> [walk] <bed> (bed.id)",
-<char0> [grab] <clothespile> (clothpile.id) |
<char1> [walk] <tablelamp> (tablelamp1.id)",
-<char0> [walk] <closet> (closet.id) | <char1>
[switchon] <tablelamp> (tablelamp1.id)",
-<char0> [putin] <clothespile> (clothpile.id)
<closet> (closet.id) | <char1> [walk]
<tablelamp> (tablelamp2.id)",
-<char0> [walk] <bed> (bed.id) | <char1>
[switchon] <tablelamp> (tablelamp2.id)",
-<char0> [grab] <clothespants>
(clothespant.id) | <char1> [grab] <book>
(book.id)",
-<char0> [walk] <closet> (closet.id) | <char1>
[walk] <chair> (chair.id)",
-<char0> [putin] <clothespants>
(clothespant.id) <closet> (closet.id) | <char1>
```

```
[sit] <chair> (chair_id)",
-<char0> [walk] <bed> (bed_id)",
-<char0> [grab] <clothesshirt>
(clothesshirt_id)",
-<char0> [walk] <closet> (closet_id)",
-<char0> [putin] <clothesshirt>
(clothesshirt_id) <closet> (closet_id)"
```

Script 2: This is an example script of two humans in a bedroom, where one human is arranging stationery on a bookshelf and another is eating food, as shown in Fig. 7.

```
-<char0> [walk] <bed> (bed_id) | <char1>
[walk] <coffeetable> (bedroomcoffeetable_id)",
-<char0> [grab] <book> (book_id) | <char1>
[grab] <chinesefood> (chinesebox)",
-<char0> [walk] <bookshelf> (bookshelf_id) |
<char1> [walk] <bed> (bed_id)",
-<char0> [putin] <book> (book_id) <bookshelf>
(bookshelf_id) | <char1> [sit] <bed> (bed_id)",
-<char0> [walk] <bed> (bed_id) | <char1> [sit]
<bed> (bed_id)",
-<char0> [grab] <magazine> (magazine_id)",
-<char0> [walk] <bookshelf> (bookshelf_id)",
-<char0> [putin] <magazine> (magazine_id)
<bookshelf> (bookshelf_id)",
-<char0> [walk] <bed> (bed_id)",
-<char0> [grab] <folder> (folder_id)",
-<char0> [walk] <bookshelf> (bookshelf_id)",
-<char0> [putin] <folder> (folder_id)
<bookshelf> (bookshelf_id)",
-<char0> [walk] <bed> (bed_id)",
-<char0> [grab] <journal> (journal_id)",
-<char0> [walk] <bookshelf> (bookshelf_id)",
-<char0> [putin] <journal> (journal_id)
<bookshelf> (bookshelf_id)"
```

References

- [1] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8494–8502, 2018. 1
- [2] VirtualHome. Knowledge base: Actions. <http://virtual-home.org/documentation/master/kb/actions.html>, 2022. Accessed: 2025-07-18. 1
- [3] VirtualHome. Knowledge base: Agents. <http://virtual-home.org/documentation/master/kb/agents.html>, 2022. Accessed: 2025-07-18. 1
- [4] VirtualHome. Knowledge base: Objects. <http://virtual-home.org/documentation/master/kb/objects.html>, 2022. Accessed: 2025-07-18. 1
- [5] VirtualHome. Knowledge base: Environment graph. <http://virtual-home.org/documentation/master/kb/environments.html#environment-graph>, 2022. Accessed: 2025-07-18. 1