A. Model Architecture

The overall architecture of @MODEL is a generic encoder-decoder design as shown in main paper. We follow X-Decoder [51] to adapt Focal-T [43] as image encoder $\mathbf{Enc_I}$ and use a number of transformer layers as text encoder $\mathbf{Enc_T}$. Decoder is a common Transformer [51] decoder structure with self- and cross-attention layers.

A.1. Formulation

First, we use image encoder $\mathbf{Enc_I}$ to extract multi-scale features $\mathbf{Z}$ from input image $\mathbf{I}\in\mathcal{R}^{H\times W\times 3}$:

$$\mathbf{Z} = \mathbf{Enc_I}(\mathbf{I}) = \langle \mathbf{z}_l \rangle_{l=1}^{L} \tag{1}$$

where $\mathbf{z}_l \in \mathcal{R}^{H_l\times W_l\times d}$ and $\{H_l, W_l\}$ is the size of feature map at level l and d is the feature dimension. Then, we use the text encoder $\mathbf{Enc_T}$ to encode a task-specific prompt into $\mathbf{P} = \langle p_1, \cdots, p_n \rangle$ of length n. Afterwards, we use the same text encoder $\mathbf{Enc_T}$ to encode a textual label into $\mathbf{Q^t} = \langle {q_1}^t, \cdots, {q_n}^t \rangle$ and create a latent queries $\mathbf{Q^l} = \langle {q_1}^l, \cdots, {q_m}^l \rangle$ as inputs of decoder. All these features are fed into @MODEL to predict the outputs:

$$\langle \mathbf{O^p}, \mathbf{O^s} \rangle = \mathbf{@Model}(\langle \mathbf{P}, \mathbf{Z} \rangle ; \langle \mathbf{Q^l}, \mathbf{Q^t} \rangle), \tag{2}$$

where $\mathbf{O^p}$ and $\mathbf{O^s}$ are the pixel-level outputs and token-level semantic outputs, respectively.

A.2. Tasks

Based on the aforementioned designs, @MODEL can be effectively employed to integrate various vision and vision-language tasks by utilizing different input combinations.

Pixel-level Output Tasks. For these tasks, such as panoptic segmentation and depth estimation, there is no textual label as input for decoder:

$$\mathbf{O^p} = \mathbf{@Model}(\langle \mathbf{P}, \mathbf{Z} \rangle ; \mathbf{Q^l}), \tag{3}$$

where $\mathbf{O^p}$ has the same size of $\mathbf{Q^l}$.

Token-level Output Tasks. For OCR, captioning and VQA, they require both latent and text queries as inputs. Hence, Eq. (2) is adapted to:

$$\mathbf{O^s} = \mathbf{@Model}(\langle \mathbf{P}, \mathbf{Z} \rangle ; \langle \mathbf{Q^l}, \mathbf{Q^t} \rangle), \tag{4}$$

where $\mathbf{O^s}$ correspondingly has equal size of $\mathbf{Q^t}$, and no pixel-level output are predicted. All predictions follow an auto-regressive strategy.

B. Loss Functions

B.1. Pixel-level Output Loss

Segmentation Loss. There are two losses on the segmentation corresponding to two tasks. For mask classification, we use text encoder $\mathbf{Enc_T}$ to encode all N class names including "background" into N text embeddings $\mathbf{E}_{cls}\in\mathcal{R}^{N\times C}$ and take it to represent the concept. Afterward, we take the first $(m-1)$ latent queries and compute the dot-product between these outputs and concept embeddings to obtain an affinity matrix $\mathbf{S}_{cls}\in\mathcal{R}^{(m-1)\times N}$ and compute $\mathcal{L}_{cls}$=$\mathbf{CE}(\mathbf{S}_{cls}, \mathbf{y}_{cls})$, with the ground-truth class $\mathbf{y}_{cls}$. For mask prediction, we use Hungarian matching [4, 6] to find the matched entries of first $(m-1)$ outputs to ground-truth annotations. Afterward, we to use binary cross-entropy loss $\mathcal{L}_{bce}$ and dice loss $\mathcal{L}_{dice}$ to compute the loss. Thus, the overall training loss function of panoptic segmentation is:

$$\mathcal{L}_{ps} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{bce}\mathcal{L}_{bce} + \lambda_{dice}\mathcal{L}_{dice}, \tag{5}$$

where λ_{cls}, λ_{bce} and λ_{dice} are coefficient weights to control different losses

Depth Estimation Loss. Given the prediction $\mathbf{O^p}$ derived from m latent queries, we use the last (m-th) latent querie to make depth prediction. In order to calculate the distance between predicted output $\widehat{\mathbf{Y}}_{de}$ and ground truth $\mathbf{Y}_{de}$, we use scale-invariant log scale loss [8, 19]. The equation of training loss is as follows:

$$\mathcal{L}_{de} = \frac{1}{n}\sum_i {d_i}^2 - \frac{1}{2}(\frac{1}{n}\sum_i d_i)^2, \tag{6}$$

where $d_i = log(y_i) - log(\widehat{y}_i)$, y_i and $\widehat{y}_i$ are ith pixel-value of $\mathbf{Y}_{de}$ and $\widehat{\mathbf{Y}}_{de}$, respectively.

B.2. Token-level Output Loss

For token-level tasks, we begin by extracting embeddings for all tokens in the vocabulary, which has a size of V, from the text encoder. Using the last n semantic token-level outputs from @MODEL, we calculate the dot product with all token embeddings to generate an affinity matrix $\mathbf{S}_{token}\in\mathcal{R}^{n\times V}$. Subsequently, we compute the cross-entropy loss $\mathcal{L}_{token}$=$\mathbf{CE}(\mathbf{S}_{token}, \mathbf{y}_{token})$, where $\mathbf{y}_{token}$ represents the ground-truth next-token id.

B.3. Multi-task Training Loss

During multi-task training, we calculate losses on the top decoder layers for each task to guide the model to converge faster in the early training stage and accelerate the overall training process. The overall training loss function is:

$$\sum_{task\in\{ps,de,ocr,ic,vqa\}} \sum_{i=1}^{nl_{task}} \lambda_{task}\mathcal{L}_{task}, \tag{7}$$

where nl_{task} represents the number of decoder layers that need to calculate the loss for different task, λ_{task} and $\mathcal{L}_{task}$ are loss weights and losses for different task, respectively.

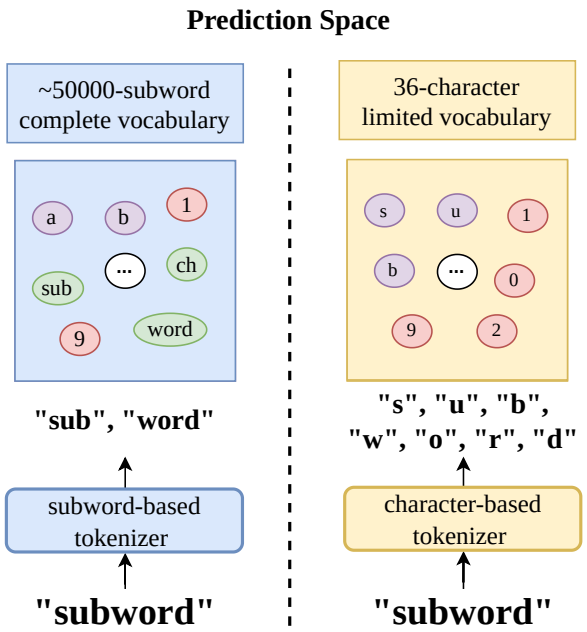

Figure 6. Comparison between subword-based tokenizer and character-based word tokenizer in our proposed @MODEL

C. Implementation Details

C.1. Multi-task Training

Training Setting. Since the number of images in OCR training dataset is much larger than datasets for the other tasks, we define OCR training dataset as the major dataset for multi-task training. It means that the total number of iterations is calculated based on the number of images in the OCR dataset. The batch sizes for panoptic segmentation, depth estimation, OCR, captioning, and VQA are 4, 4, 768, 8, and 4, respectively, to accommodate datasets of different sizes for various tasks. The model is trained with 15 epochs based on OCR datasets on 4 A100 (40G). The AdamW optimizer is used with the initial learning rate 1^e-5. A step-wise scheduler is used to decay the learning rate by 0.1 on the fraction [0.6, 0.8] of training steps.

Hyperparameter Choice. In @MODEL, the decoder has 7 decoder layers. Due to segmentation and OCR are the top two scored tasks (from user study), and the OCR datasets are very large, the amount of data for each task is unbalanced, we set nl_{task} as 6, 3, 6, 3 and 3 for panoptic segmentation, depth estimation, OCR, captioning and VQA, respectively, to allow the model to focus more on segmentation and OCR during training. We set loss weights λ_{task} as 1, 10, 10, 2 and 2, respectively. Because the loss values of OCR and depth estimation in the later stage of training are very small, in order to minimize significant differences in the loss magnitude for each task as much as possible, we have made such a setting. And in Eq. (5), we set λ_{cls} = 2, λ_{bce} = 5 and λ_{dice} = 5 as default, following the settings of X-Decoder.

C.2. Single-task Training

All tasks are trained with AdamW as the optimizer on 4 1080Ti (11G), except OCR. The initial learning rate is 1^e-5 and reduced by 10 times after 60% and 80%.

Panoptic Segmentation. We train the model for 50 epochs. We set the image resolution to 640×640 and the batch size to 4.

Depth Estimation. We train the model for 50 epochs. We set the image resolution to 480×640 and the batch size to 16. Note that this task is very unstable and requires careful hyperparameter tuning. If you encounter training errors, you can increase the batch size, reduce the learning rate or training with single precision (FP32).

Optical Character Recognition. We train the model for 10 epochs on 4 A100 (40G). We set the image resolution to 64×200 and the batch size to 1024.

Image Captioning. We train the model for 50 epochs. We set the image resolution to 480×640 and the batch size to 16. We use all captions for training and do not use beam search and CIDEr optimization.

Visual Question Answering. We train the model for 30 epochs. We set the image resolution to 480×640 and the batch size to 16.

C.3. Character-based Tokenizer with Limited Vocabulary for OCR

In our main paper, we observed that subword-based tokenizer with complete vocabulary hurts the performance of OCR task. In Fig. 6, we show how to use character-based tokenizer and much smaller limited vocabulary to perform OCR task. Using a character-based word tokenizer to divide the text that needs to be recognized into characters one by one, model only needs to predict token from the limited vocabulary space, and do not need to select candidate subword from the complete vocabulary. This reduces the prediction space and improves the accuracy of prediction.

D. User Study

D.1. Comments on Generalist Assistance Systems

By conducting the questionnaire survey, we communicated with visually impaired individuals to comprehend the functionalities they expect a generalist assistive system should possess. We got some thoughts like: *"It should find the door, look for stairs in an open area, read the*

house/room number, read signs/plates, describe the environment, warn me of obstacles, and can navigate the corridors with a floor plan.". Some participants also described specific usage scenarios, *"I would use navigation and obstacle detection systems outside. It should warn me of obstacles or describe something I'm about to encounter. For example, if I'm navigating outside and there's a road ahead, then it should say if it has a roundabout or an intersection. Or, if there is a railroad crossing, announce something similar. It would be cool if there was an all around view. The system says, the front of you is street and the back is a building, left is bike racks, etc. If there is a name of the store, read it out. The most important thing is to have a general navigation ability based on this all around view. If I then say navigate to the store (name of the store) recognized by this system, then it should navigate me there."*. Based on these thoughts and comments, essential functions identified by People with Visual Impairments (PVIs) that a generalist assistive system should include are:

(1) **Navigation and Obstacle Avoidance.** A critical component is a navigation system integrated with obstacle detection capabilities. PVIs desire a system that allow for interactive navigation, where users can request directions to specific locations identified by the system.
(2) **Text Recognition and Environmental Description.** The ability to recognize and verbally relay textual information is also important. This includes identifying and reading door labels, room numbers, and signs. Furthermore, recognizing the names of stores, significant landmarks or other text contributes to better environmental understanding and orientation.
(3) **Comprehensive Scene Interpretation.** PVIs expressed a desire for a system that provides a holistic view of their surroundings. This "all-around vision" function should describe streets, buildings, and other elements in the vicinity.
(4) **Integration of Text-to-Speech Technology.** Incorporating text-to-speech technology for dynamic interaction is also valuable.

D.2. More Comments

Navigation. The majority of participants (5 P: 5 participants) prioritize outdoor navigation, noting its greater complexity and risk. They highlight that outdoor environments pose larger obstacles, longer and more complex routes, and a higher likelihood of getting lost compared to indoor scenarios. One participant emphasized, *"Outdoor navigation is much more important. Indoors, the reach of a cane is much more likely to adequately capture the surroundings. The distances are shorter and the density of people is higher."*. Another added, *"Definitely outdoors. If I have to go into a building I don't know, it will probably only be for once. It's not worth learning a way to do that."*. The unpredictability of outdoor spaces, such as traffic, was also mentioned as a significant factor. Conversely, a minority (2 P) believes that indoor navigation is more important. They mention the challenges of navigating within large unfamiliar buildings, locating specific rooms, stairs, elevators, or exits, walking across a large open-area, and walking in rooms with highly differentiated structures, such as restrooms. Importantly, they spend most of their time indoors.

Text Recognition. Today, PVIs mainly use screen readers to recognize digital texts and usually use smartphones, or smartphones Apps to read non-digital texts. However, they find non-digital text reading is difficult and cumbersome, like *"Everything I receive on paper in the post annoys me. I use apps like Seeing AI and Be My Eyes or the iPhone's magnifying glass to read non-digital texts, but using a smart glass to read these text directly would be better."*. They also pointed out that it is also important for them to read signs to find the right floor or hallway and read door numbers to enter the right room.

Other Functions. About depth estimation, *"This function helps one develop a mental map of an environment. You get the proportions well."*. About object location, *"In my personal environment I am always very sure where all the things I am looking for are. However, locating a true one in a larger shelf section of 3-4 meters would be very useful. A function that detects objects that don't belong in that space would also be very helpful to check a room for overlooked clutter. The glasses could use a reference photo of the tidy room and then report any anomalies, such as dirty dishes on the table or socks on the floor."* and *"If I only need it if I can't find something in my apartment, it could make the search easier, but I would need it pretty rarely."*. About surroundings understanding, *"It would be important to me that the description be highly efficient. The short form is always first"* and *"Most of the time we are not interested in the scene because it is too much information for us. But descriptions of photos, environments, etc. are very exciting. ChatGPT is really great."*. About scene recognition, *"Perhaps interesting for recognize different scenarios, but a correspondingly efficient description of the image would serve the same purpose. I can't imagine a situation where I would need room detection. I usually know which room I'm going to or being led into."*. About visual Q&A, *"This function would make it possible to expand a short initial description of an image dynamically and according to your own needs. That would improve the overall function enormously."*.

Interaction. If there were such a general system, PVIs prefer interacting with system through discrete button presses or subtle gestures (6 P), rather than voice commands (1 P) for privacy reasons, when inputting instructions. For receiving system feedback, they show a preference for auditory feedback (for general purpose) and vibrations (for special

Task			ADE-150	VizWiz_Cap		VizWiz_VQA				
PS	IC	VQA	PQ	B@1	CIDEr	Other	Unans	Yes/No	Number	Acc(%)
✓			39.2	–	–	–	–	–	–	–
	✓		–	60.0	45.1	–	–	–	–	–
		✓	–	–	–	30.5	92.1	70.1	13.7	49.1
✓	✓		37.7	57.8	46.8	–	–	–	–	–
	✓	✓	–	59.8	46.3	32.2	86.5	73.4	16.4	48.8
✓	✓	✓	38.5	61.0	52.5	39.4	88.2	70.1	10.8	53.7

Table 7. **Comparison of results of mixed training for different tasks.** Note:"Other", "Unanswerable", "Yes/No", "Number" are 4 different answer types for VQA. (PS = panoptic segmentation, IC = image captioning).

purpose such as obstacle avoidance).

Based on these comments and ideas, it becomes evident that for PVIs, navigation and quick, direct recognition of non-digital text are the two most critical functionalities. Meanwhile, the multifaceted nature of navigation encompasses functions like environmental comprehension, obstacle avoidance, path planning, voice guidance and *etc.* These insights serve as valuable guidance for our work. Furthermore, the analysis of participants' relevant feedback has provided us with an initial understanding of creating a universal assistive system.

E. More Experiments

E.1. Complementariness in Multitasking

As shown in the experiments section, our @MODEL exhibits a strong performance in captioning and VQA under multi-task training. Here, we further study the role of segmentation objectives in vision-language (VL) understanding, as well as the role of different vision-language understanding tasks on each other. To investigate, we mix different tasks for training. In Table 7, for captioning, when jointly trained with VQA or PS, or all tasks, CIDEr improved by 1.2, 1.7 and 7.4 respectively. For VQA, we report 5 numbers for better analysis, namely the accuracy for 4 Q&A types: *other*, *unanswerable*, *yes/no*, *number*, and the overall accuracy. From the comparison of these numbers, when training VQA alone, the model tends to predict "unanswerable" to improve the accuracy. Because in the dataset, the *unanswerable* type of Q&A is the most common. For other types of Q&A, the accuracy is relatively lower because a deeper or more granular understanding of the semantic information of image is required to predict the correct answers. After joint training with captioning, the accuracy of *unanswerable* type Q&A decreased, and the accuracy of other types increased. The model does not just return "unanswerable" blindly but understands more semantic information of the image and then make predictions. When all tasks are trained together, the accuracy of *other* type Q&A is greatly improved (+8.9%). We analyze that it is because the question of this type of Q&A is usually "*what is this?*", and the segmentation task naturally has a very good assisting effect in answering this question. Segmentation data can help models to learn more fine-grained visual understanding and consequently benefit vision-language tasks. We also give some examples to show these improvements in Fig. 7. Along with our findings in the main paper, we conclude that segmentation has clear benefits to VL learning and different VL tasks are complementary to each other.

F. More Visualization

F.1. Visualization on Test Datasets

We present a comprehensive visualization of our model's performance on the test datasets in Fig. 8. For segmentation, we show some results in outdoor scene, indoor scene, multi-person scene, especially the open-area mentioned by the PVIs. For OCR, various types of text recognition results can show the robustness and generalization of @MODEL. For other task, @MODEL can also perform well.

F.2. Zero Shot

Finally, we apply the 5 tasks in a zero-shot manner to show the generalization ability of @MODEL. @MODEL performs well on three tasks: segmentation, depth estimation, and OCR, as shown in Fig. 9. However, for open-ended tasks, captioning and VQA , the performance on out-of-dataset data can sometimes be less satisfactory (Fig. 9 (B)). Therefore, it may be necessary to perform large-scale pre-training to enhance the model's capability for handling these tasks well in zero-shot.

G. More Discussion

This section discusses the limitations and future work of this work for more insights on the research in this track.

Pre-training. In the main paper, we did not perform pre-training. This has a certain impact on the capability of zero shot, especially for open-ended tasks. In the future, we plan to conduct pre-training on large-scale corpora to enhance the model's zero-shot capability. Additionally, we use a unified language encoder to encode text in @MODEL. Pre-training can enrich the vocabulary size, thereby improving the model's ability to open-vocabulary segmentation. The importance of this open-vocabulary capability for practical applications is self-evident, especially for blind users. As mentioned by blind users in user study, they require systems with high object recognition accuracy. When the model has seen a greater variety of objects and can distinguish between them, the recognition accuracy also increases. Additionally, this open-vocabulary capability allows the model to handle previously unseen objects. In sum, after pre-training, the model can better handle the diversity, complexity and unpredictability of usage scenarios.

Multi-task Training. As shown in the main submission, @MODEL performs well on the OCR task during single-

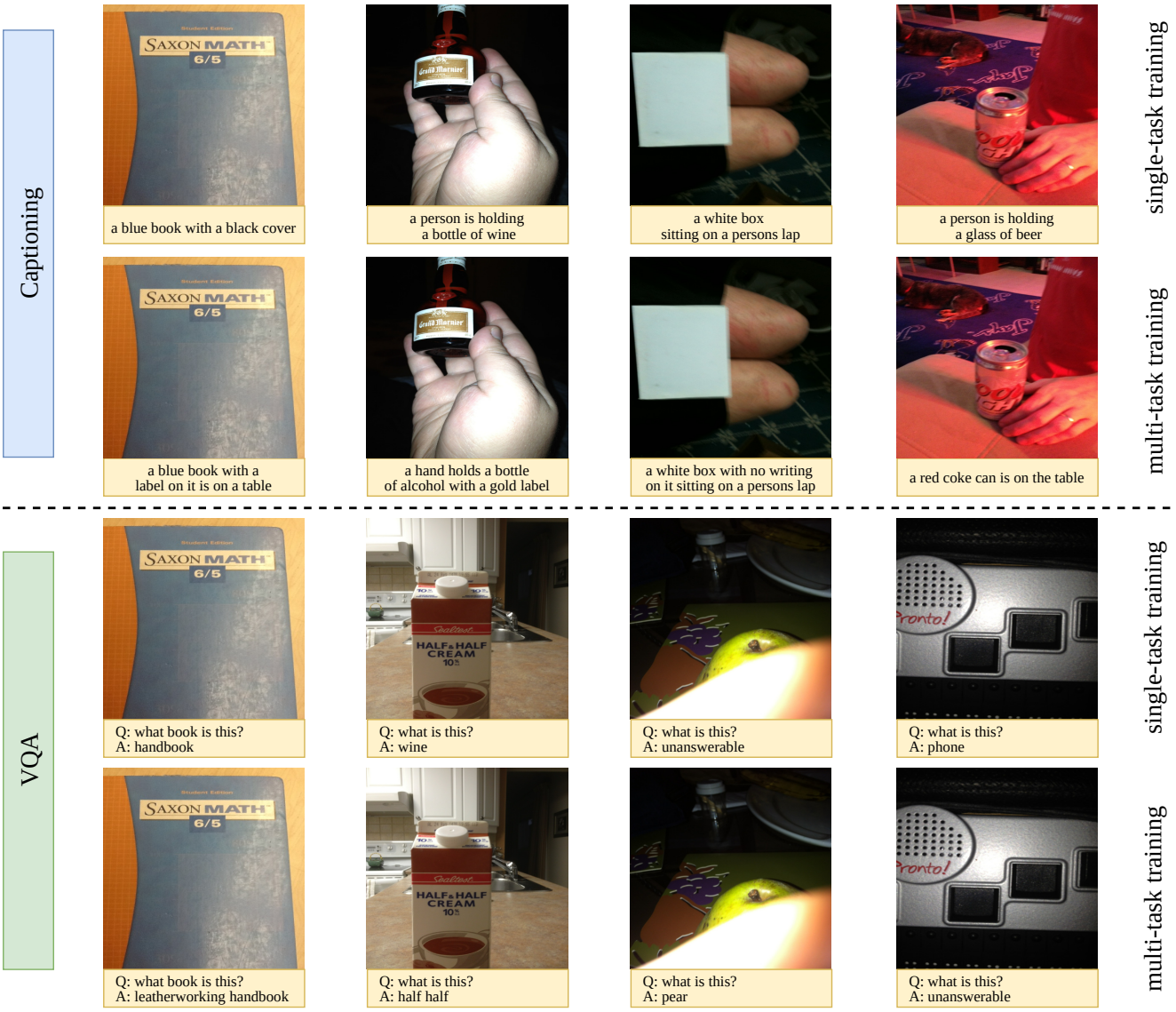

Figure 7. Examples to show the promotion of vision for vision-language and complementariness between different vision-language tasks.

task training, but there is a certain gap in performance during multi-task training. Our analysis suggests that the OCR dataset is too large, and the model does not balance multiple tasks during training. When dealing with multi-task training with extremely imbalanced dataset sizes, it is not enough to merely adjust loss weights differently. In the future, we may try more optimization methods for multi-task learning to ensure performance without greatly increasing the training time.

Functions Development and Model Deployment. In our user study, we have identified several potential and crucial functions that received unanimous agreement from participants. Furthermore, it's important to note that @MODEL is not limited to these five tasks alone; it can be extended to more uni-modal or multi-modal tasks to provide more functionalities. Our future research direction will focus on building a PVIs-Centred generalist assistive system, leveraging @BENCH and @MODEL as cornerstones, to develop a wide range of practical functions and services. As for model deployment, although @MODEL achieves high performance on multiple datasets, since the model is based on Transformer, its costs are larger than the non-Transformer models. Additionally, though @MODEL only has 62M parameters, it is still difficult to deploy such a model in the portable device used by PVIs. Therefore, in our future work we will discover how to extract or compress @MODEL into an efficient light-weight model.

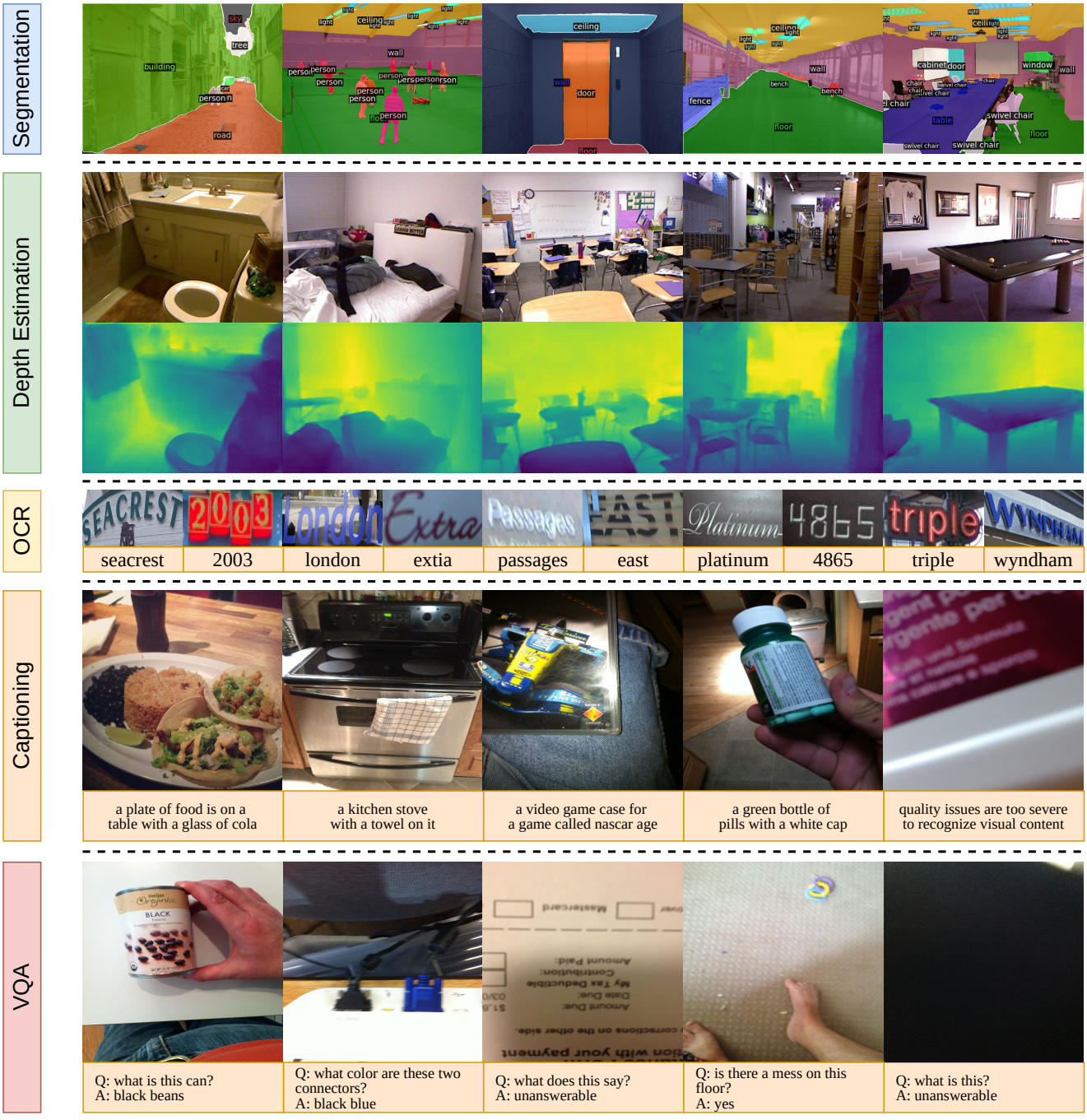

Figure 8. Examples on different test datasets. These images cover a diversity of visual domains and concepts in the daily life of PVIs.

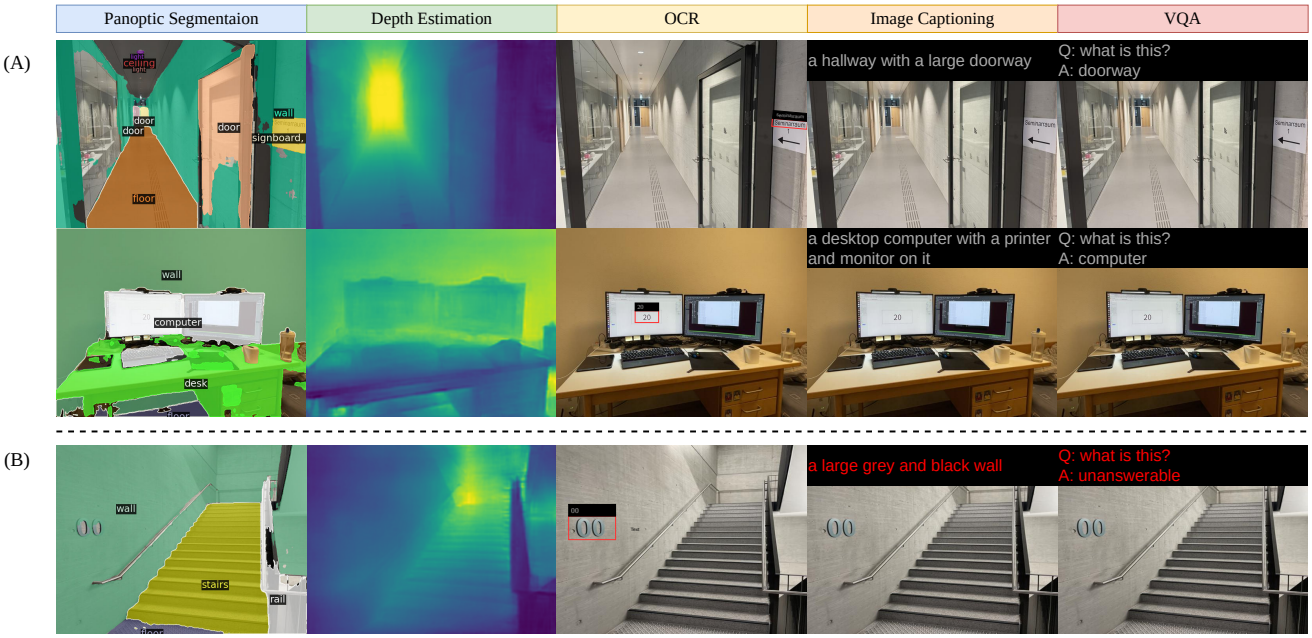

Figure 9. Examples on real-world scenes. These images were randomly collected by using mobile phone.