# Visual Item Selection With Voice Assistants

## A systems perspective

Prashan Wanigasekara[1†*], Rafid Al-Humaimidi[2♤*], Turan Gojayev[3†*]
Niloofar Gheissari[4†], Achal Dave[5◆], Stephen Rawls[5†], Fan Yang[1†], Kechen Qin[1†], Nalin Gupta[6†]
Spurthi Sandiri[1†], Chevanthie Dissanayake[6♥], Zeynab Raeesy[6†], Emre Barut[5†], Chengwei Su[1†]
{wprasha,rafidka,tgojayev,gniloo,achald,sterawls,fyaamz,qinkeche,nalgupta,spurthi,raeesyzr,ebarut,chengwes}@amazon.com
Amazon Alexa AI[†], Amazon Web Services[♤], Amazon Physical Stores[◆], Amazon Prime Air[♥]
Cambridge, MA, USA[1], Vancouver, Canada[2], Berlin, Germany[3], San Jose, CA, USA[4], New York, NY, USA[5], Seattle, WA, USA[6]

## ABSTRACT

Interacting with voice assistants, such as Amazon Alexa to aid in day-to-day tasks has become a ubiquitous phenomenon in modern-day households. These voice assistants often have screens to provide visual content (e.g., images, videos) to their users. There is an increasing trend of users shopping or searching for products using these devices, yet, these voice assistants do not support commands or queries that contain visual references to the content shown on screen (e.g., *"blue one", "red dress"*). We introduce a novel multi-modal visual shopping experience where the voice assistant is aware of the visual content shown on the screen and assists the user in item selection using natural language multi-modal interactions. We detail a practical, lightweight end-to-end system architecture spanning from model fine-tuning, deployment, to skill invocation on an Amazon Echo family device with a screen. We also define a niche "Visual Item Selection" task and evaluate whether we can effectively leverage publicly available multi-modal models, and embeddings produced from these models for the task. We show that open source contrastive embeddings like CLIP [30] and ALBEF [24] have zero-shot accuracy above 70% for the "Visual Item Selection" task on an internally collected visual shopping dataset. By further fine-tuning the embeddings, we obtain further gains of 8.6% to 24.0% in relative accuracy improvement over a baseline. The technology that enables our visual shopping assistant is available as an Alexa Skill in the Alexa Skills store.

## CCS CONCEPTS

• **Applied computing** → **Service-oriented architectures**; **Online shopping**.

## KEYWORDS

Multi-modality, voice assistants, systems architecture, visual shopping, deployed system, app based skills

∗ Equal Contribution.


*WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA*

ACM ISBN 978-1-4503-9419-2/23/04.
https://doi.org/10.1145/3543873.3584655

**ACM Reference Format:**
Prashan Wanigasekara[1†][1], Rafid Al-Humaimidi[2♤][1], Turan Gojayev[3†][1] and Niloofar Gheissari[4†], Achal Dave[5◆], Stephen Rawls[5†], Fan Yang[1†], Kechen Qin[1†], Nalin Gupta[6†], Spurthi Sandiri[1†], Chevanthie Dissanayake[6♥], Zeynab Raeesy[6†], Emre Barut[5†], Chengwei Su[1†]. 2023. Visual Item Selection With Voice Assistants: A systems perspective. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30-May 4, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3543873.3584655

## 1 INTRODUCTION

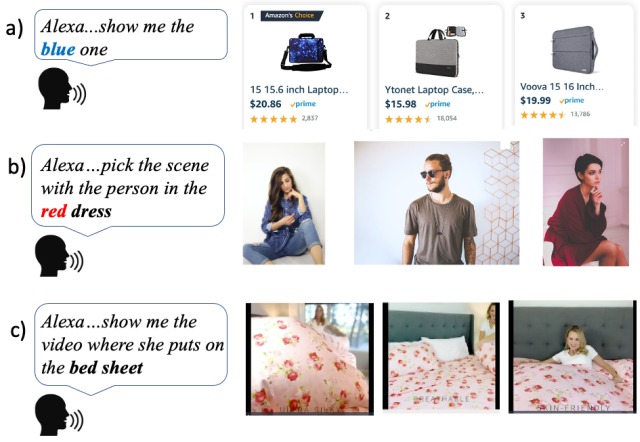


**Figure 1: Natural language multi-modal interactions (audiovisual sensory input) with a voice assistant for Visual Shopping. The images are taken from unsplash.com and used here only for illustrative purposes.**

We study the problem of "Visual Item Selection" with a voice assistant with co-observed voice *and* visual content. Here, the term "co-observed" refers to the fact that the displayed items are observed by both the user and the voice assistant. Some sample multi-modal user interactions of this novel shopping experience are shown in Fig 1. In Fig.1 a) the user intends to select a specific laptop case out of 3 laptop cases shown with a color reference (*"blue one"*), in Fig.1 b) the user intends to select a scene that contains a social media influencer with a color and product category reference (*"person in the red dress"*), and in Fig.1 c) the user intends to select a product demonstration video from 3 videos shown with a product category

reference (*"bed sheet"*). To the best of the authors' knowledge, our offering is the first instance where a commercial voice assistant enables the user to select visual items shown on the screen by referring to their visual attributes, where the selection is powered by a multi-modal model artifact and a multi-modal interaction (audio-visual sensory input). These user referring expressions select one item out of $n$ items that are on display. All items on display can be termed as the co-observed visual context. The item that the user selected is the positive example ($label = 1$). The items that were shown and not selected are the negative examples ($label = 0$). The main contributions of this paper are:

- **Novel visual shopping experience**: We introduce a novel commercially deployed visual shopping experience where the voice assistant is aware of the co-observed voice *and* visual context.
- **End-to-end system architecture**: We detail a low latency, highly scalable, service oriented, app based (a skill in Alexa Skills store) end-to-end system architecture.
- **Niche "Visual Item Selection" task definition**: We define a niche "Visual Item Selection" task within the context of the user-voice assistant interaction and show that open source contrastive embeddings (e.g., CLIP, ALBEF) can be effective in a zero-shot manner (accuracy above 70%).
- **Fine-tuning on a custom dataset**: We show that further fine-tuning the contrastive embeddings on an internal visual shopping dataset collected via Amazon Mechanical Turk (MTurk) [3] can improve the performance of the contrastive embeddings on the "Visual Item Selection" task.

Due to space constraints, the discussion on related work is relegated to the Appendix A.3.

## 2 END-TO-END SYSTEM ARCHITECTURE

We show the end-to-end architecture of this system in Fig. 2. We focused on designing a system with the following key features; (1) low latency, (2) high scalability, (3) service oriented client-server architecture, (4) app based (a skill in Alexa Skills store), (5) ability to be easily incorporated into existing deployed systems with (6) automated training, evaluation and model hosting orchestration.

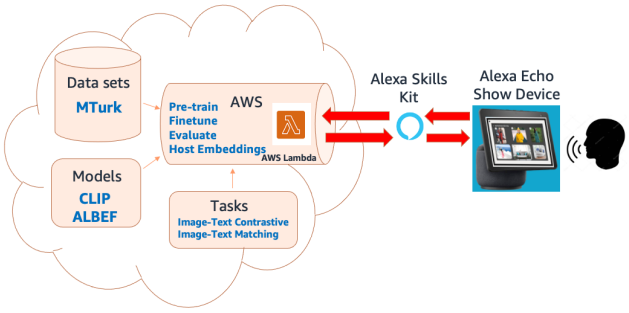


**Figure 2: High-level systems design.**

The process of training, fine-tuning and model deployment is performed using Amazon Sagemaker [32]. The pipeline shown as "AWS" in Fig. 2 takes the data sets, models, and tasks as input and provides a fine-tuned contrastive image and text embedding as output. Once the embeddings are fine-tuned, we use Amazon SageMaker Endpoint as a hosting service to host the embeddings and an AWS Lambda [33] to handle incoming queries that request access to the embeddings. This system resides in the AWS cloud and follows service-oriented architecture design principles. Using AWS enables us to dynamically scale based on incoming user traffic to the visual shopping skill.

The Alexa Skills Kit [4, 20] and its Alexa Skills Interaction Model (Appendix A.4) reside on the client side in the Amazon Echo Show device. The users opt-in to use the visual shopping skill on their Alexa Echo Show devices, similar to the installation of a mobile application. It handles the user's incoming voice requests, on-device user interface (UI) flow, the user's tactile input and makes calls (shown with red left & right arrows in Fig. 2) to the AWS Lambda which in turn calls a Sagemaker Endpoint that is hosting the contrastive embeddings.

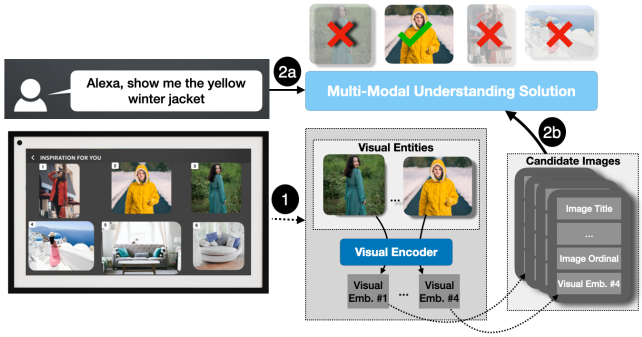


**Figure 3: Runtime user interaction flow. The images are taken from unsplash.com and used here only for illustrative purposes.**

Contrastive embeddings bring vision and text modalities into a shared embeddings space. By taking a simple dot product between the vision and text embeddings, we can gauge the alignment between a text that resulted from a voice command and the images on display with very low latency. The image that has the highest contrastive embedding dot product with the referring text can be selected as the item the user is referring to. As seen in Fig. 3 at step (1) the images are displayed to the user. The visual embeddings for these images can be pre-computed and cached beforehand. During runtime in step (2a) the user makes a referring utterance to select a visual item from a list of shown items. The user utterance text is then routed to the cloud, the text encoder that resides in the cloud hosted by AWS computes the text embedding and the contrastive text embedding is retrieved to the device via the Alexa Skills Kit and AWS Lambda. In parallel, as shown in step (2b) the Alexa Skills Kit retrieves the pre-computed and cached visual embedding candidate images from the visual list items. Once visual and text embeddings are ready they are given as input to the Alexa Skills Interaction Model (shown in Fig. 3 as the "Multi-modal Understanding Solution") and the model picks the image that has the best match via the highest dot product similarity. All this happens in under 100 ms which is a latency that cannot be perceived by a human. This simple, low-latency design, app-based opt-in installability and the service-oriented architecture

powered by AWS enables our system to be easily incorporated into existing systems (e.g, non-multi-modal voice only systems).

## 3 VISUAL ITEM SELECTION

In a conversation between the voice assistant and the user, this visual item selection step can be a small piece of the overall user-system dialogue. On the surface, this looks like a simple text-to-image retrieval task. It can also resemble a multi-modal ranking task, where each image is ranked on how much it matches the referring text. Depending on the use case, it can also be formulated as visual grounding task. There is definitely an intersection of subfields, as shown in Figure 4.

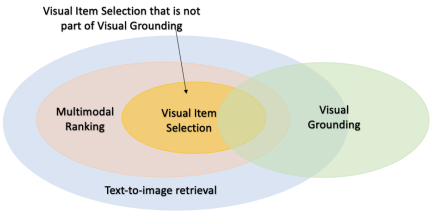


**Figure 4: Intersection of subfields of text-to-image retrieval, multi-modal ranking, visual item selection and visual grounding.**

We formulate our "Visual Item Selection" task to be a subset of text-to-image retrieval and multimodal ranking, as shown in Figure 4. It also intersects with visual grounding. In this paper, we are interested in "Visual Item Selection" that is not part of visual grounding. The task we describe as "Visual Item Selection" differs from already existing subfields of text-to-image retrieval, multimodal ranking and visual grounding in the following manner.


**Figure 5: Visual Item selection conditioned on co-observed context**

In "Visual Item Selection" the co-observed visual context has $n$ items shown to the user beforehand and the user selects a few (usually 1) out of $n$. The referring text the customer uses is not a full description of the desired item and can be a partial shortcut. For example, in Figure 5, Case A, if the Voice Assistant is showing 3 handbags on the screen and the user wants to select one particular handbag the user would say *"I want the pink one"* where all the other handbags are not pink. In Case B where the handbags that are on display are all pink, the user would use a different referring attribute like *"the one with the leather strap"*. In both cases the user wanted the same handbag but had to adjust the referring utterance based on other items that were on display. Thus, the referring utterance is a partial sentence that is dependent on the co-observed context. This is slightly more nuanced than a generic text-to-image retrieval or multimodal ranking task, where each image would be independent of others and the text would either be a well-formulated caption or a full description of the image independent of other images. Additionally, for our use case we do not explore a visual grounding based formulation because grounding models tend to be more complex and sub-optimal for a latency constrained environment.

After defining the Visual Item Selection task we create a suitable multimodal visual item selection dataset that match the task. Then, we focus on the visual item selection step in the conversational user-system interaction, and evaluate the effectiveness of using contrastive embeddings.

## 4 EXPERIMENTS

In this section, we first introduce the contrastive models we use in section 4.1. We then give a brief description of the datasets in section 4.2. We define our fine-tuning tasks in section 4.3. Fine-tuning task evaluation is in section 4.4. The results and analysis is discussed in section 4.5.

### 4.1 Contrastive Models

This paper uses embeddings from 2 models trained with contrastive loss [10, 35, 36, 40]; CLIP [30] and ALBEF [24]. The authors of these papers have pretrained the models on paired image-text datasets that were crawled from the internet. In this paper, we use these models in a zero-shot manner, as well as after fine-tuning them on a custom multi-modal dataset.

### 4.2 Datasets

The multi-modal MTurk dataset is created by showing Mechanical Turkers (MTurkers) product images, scene images, video thumbnails and asking them to pick one out of the many products, scenes or movies using a single referring utterance. We define one such referring act that includes multiple images and a single referring utterance as a single "instance". The newly collected MTurk dataset has a Train:Dev:Test split sizes of 33k:4k:4k instances respectively.

### 4.3 Fine-tuning Tasks

We define 3 different tasks to further fine-tune the embeddings. The objective of these tasks together with the input they require are listed in Table 1.

**Task 1** in Table 1, involves following the original pre-training procedures described in [30] and [24]. **Task 2** is defined only for the ALBEF architecture and refers to training with Image-Text Matching (ITM) loss only. Because we have formulated our Visual Item Selection as a subset of image retrieval (Fig. 4) we can proceed this way, namely: given an image-text pair, decide if they match each other. Both these tasks use the user utterance together with the positive image only. To take advantage of supervised signals from data, we define **Task 3**. In Task 3, for CLIP we sum the contrastive loss and cross-entropy (CE) loss, whereas for ALBEF we do adaptations to the ITM head and use only CE loss as seen in the last row (Task 3) of Table 1. The reason for summing the contrastive and CE losses

| Task | Loss Formulation | Input for fine-tuning |
|---|---|---|
| Task1 | CLIP $\mathcal{L}_{contrastive} = \frac{1}{2N}\sum_{i=1}^{N}\left(\mathcal{L}_i^{(image\rightarrow text)} + \mathcal{L}_i^{(text\rightarrow image)}\right)$<br><br>ALBEF $\mathcal{L} = \mathcal{L}_{ITC} + \mathcal{L}_{ITM} + \mathcal{L}_{MLM}$ | utterance and positive image |
| Task2 | ALBEF $\mathcal{L} = \mathcal{L}_{ITM}$ | utterance and positive image |
| Task3 | CLIP $\mathcal{L} = \mathcal{L}_{contrastive} + \mathcal{L}_{cross-entropy}$<br><br>ALBEF $\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_c^i \log\left(p\left(c\middle\vert \left\langle \mathbf{I_i^1}, .., \mathbf{I_i^C}\right\rangle, \mathbf{T_i}\right)\right)$ | utterance and all images |

**Table 1: Fine-tuning tasks. Please see Appendix [A.5, A.6] for more details on definitions and notation.**

| # | Experiment | Embedding | Task | # instances | MTurk Test set Performance | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | Instance Level | | | | Relative % Improv. |
| | | | | | P | R | F1 | Accuracy | |
| 1 | baseline | CLIP | zero-shot | 4,152 | 70.17 | 70.11 | 70.12 | 70.11 | - |
| 2 | + fine-tune on | | Task 1 | | 75.87 | 75.84 | 75.84 | 75.84 | 8.17% |
| 3 | {MTurk train} | | Task 3 | | 76.19 | 76.16 | 76.15 | 76.16 | 8.63% |
| 4 | baseline | ALBEF | zero-shot | 4,152 | 71.29 | 71.29 | 71.29 | 71.29 | - |
| 5 | + fine-tune on {MTurk train} | | Task 1 | | 79.63 | 79.6 | 79.60 | 79.60 | 11.66% |
| 6 | | | Task 2 | | 79.78 | 79.72 | 79.72 | 79.72 | 11.82% |
| 7 | | | Task 3 | | 88.42 | 88.41 | 88.41 | 88.42 | 24.03% |

**Table 2: Performance on MTurk test set.**

for CLIP in task 3 is that the CLIP architecture doesn't have a specific block modeling the matching objective, but rather relies purely on the contrastive representations. When we feed negative images together with the positive image and calculate the probabilities, we are going to do it via a non-parametric *softmax*, which might not perform well when used as the only objective for CLIP. In contrast to that, ALBEF has a specific ITM head that is designed to match the image-text pairs and hence we feel safer to not add the contrastive loss to ALBEF. The objectives used in these fine-tuning tasks are described in Appendix A.6 in more detail.

## 4.4 Fine-tuning Task Evaluation

To assess which of the fine-tuning tasks has larger impact on our system, we execute each of them and then do the evaluation as described in Appendix A.7. We use "instance level" precision (P), recall (R), F1, and accuracy as our evaluation metrics and report results in Table 2.

## 4.5 Results and Analysis

We show the results on the MTurk test set for various experimental setups in Table 2. Table 2, row 1 and row 4 shows the zero-shot performance of CLIP and ALBEF embeddings on the "Visual Item Selection" task. We see that both public CLIP and ALBEF have accuracy above 70% which is above our stated threshold. We could interpret it as when the user asks for an item displayed on the screen in a conversation, the voice assistant gets the item right 70% of the time. We consider this zero-shot performance of the embeddings as our baseline. The rest of the table shows the results brought by fine-tuning tasks. In Row 2 and Row 5 of Table 2 we see 8.17% and 11.66% relative accuracy improvement for Task1, respectively for CLIP and ALBEF. This shows that the contrastive embeddings can be further fine-tuned to improve its performance. We noticed sudden peak in contrastive loss and drop in image-text matching loss happening at the same time during Task1 training for ALBEF. We hypothesize that these losses are competing with each other. Indeed, by dropping the contrastive loss, we get improvements for Task2, although the gain is marginal. In Table 2, row 3 and row 7 we fine-tune the models on MTurk train dataset using Task 3. We see CLIP's relative accuracy performance compared to the baseline improve by 8.63%, which is slightly higher than when fine-tuned on Task 1. However, for ALBEF model fine-tuned with Task 3 objective, we see a more significant 24.03% relative accuracy improvement. We attribute this gain to the usage of supervised signals for Task 3.

# 5 CONCLUSION

We introduce a novel visual shopping use case that is deployed as an Alexa Skill. We define a niche "Visual Item Selection" task within the wider context of an interaction between a voice assistant and a user. We show that embeddings such as CLIP [30] and ALBEF [24] trained with a contrastive loss give zero-shot accuracy performance that is above a 70% threshold for this "Visual Item Selection" task. We also show that further fine-tuning these embeddings on a multi-modal dataset collected via Amazon Mechanical Turk [3] improves their performance.

# REFERENCES

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al.

2022. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198* (2022).
[2] Amazon. 2022. Alexa Presentation Language. https://developer.amazon.com/en-US/docs/alexa/alexa-presentation-language/add-visuals-and-audio-to-your-skill.html
[3] Amazon. 2022. Amazon Mechanical Turk. https://www.mturk.com/
[4] Amazon. 2022. Amazon Skills Kit. https://developer.amazon.com/en-US/alexa/alexa-skills-kit
[5] Amazon. 2022. OpenAI. https://openai.com/
[6] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. 2022. Conditioned and composed image retrieval combining and partially fine-tuning CLIP-based features. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 4955–4964. https://doi.org/10.1109/CVPRW56347.2022.00543
[7] Soravit Changpinyo, Jordi Pont-Tuset, Vittorio Ferrari, and Radu Soricut. 2021. Telling the What while Pointing to the Where: Multimodal Queries for Image Retrieval. https://doi.org/10.48550/ARXIV.2102.04980
[8] Tongfei Chen, Chetan Naik, Hua He, Pushpendre Rastogi, and Lambert Mathias. 2019. Improving Long Distance Slot Carryover in Spoken Dialogue Systems. https://doi.org/10.48550/ARXIV.1906.01149
[9] Yanbei Chen, Shaogang Gong, and Loris Bazzani. 2020. Image search with text feedback by visiolinguistic attention learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3001–3011.
[10] S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. 539–546 vol. 1. https://doi.org/10.1109/CVPR.2005.202
[11] Nilotpal Das, Aniket Joshi, Promod Yenigalla, and Gourav Agrwal. 2022. MAPS: Multimodal Attention for Product Similarity. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 3338–3346.
[12] Sonam Goenka, Zhaoheng Zheng, Ayush Jaiswal, Rakesh Chada, Yue Wu, Varsha Hedau, and Pradeep Natarajan. 2022. FashionVLP: Vision Language Transformer for Fashion Retrieval with Feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14105–14115.
[13] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Xin-Shun Xu, and Mohan Kankanhalli. 2018. Multi-modal preference modeling for product search. In *Proceedings of the 26th ACM international conference on Multimedia*. 1865–1873.
[14] Mariya Hendriksen, Maurits Bleeker, Svitlana Vakulenko, Nanne van Noord, Ernst Kuiper, and Maarten de Rijke. 2022. Extending CLIP for Category-to-image Retrieval in E-commerce. In *European Conference on Information Retrieval*. Springer, 289–303.
[15] Mariya Hendriksen, Maurits Bleeker, Svitlana Vakulenko, Nanne van Noord, Ernst Kuiper, and Maarten de Rijke. 2021. Extending CLIP for Category-to-image Retrieval in E-commerce. https://doi.org/10.48550/ARXIV.2112.11294
[16] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. *OpenCLIP*. https://doi.org/10.5281/zenodo.5143773 If you use this software, please cite it as below..
[17] Aishwarya Kamath, Mannat Singh, Yann LeCun, Ishan Misra, Gabriel Synnaeve, and Nicolas Carion. 2021. MDETR - Modulated Detection for End-to-End Multi-Modal Understanding. *CoRR* abs/2104.12763 (2021). arXiv:2104.12763 https://arxiv.org/abs/2104.12763
[18] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in vision: A survey. *ACM computing surveys (CSUR)* 54, 10s (2022), 1–41.
[19] Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. 2021. SIMMC 2.0: A Task-oriented Dialog Dataset for Immersive Multimodal Conversations. *CoRR* abs/2104.08667 (2021). arXiv:2104.08667 https://arxiv.org/abs/2104.08667
[20] Anjishnu Kumar, Arpit Gupta, Julian Chan, Sam Tucker, Bjorn Hoffmeister, Markus Dreyer, Stanislav Peshterliev, Ankur Gandhe, Denis Filiminov, Ariya Rastrow, Christian Monson, and Agnika Kumar. 2017. Just ASK: Building an Architecture for Extensible Self-Service Spoken Language Understanding. https://doi.org/10.48550/ARXIV.1711.00549
[21] Katrien Laenen, Susana Zoghbi, and Marie-Francine Moens. 2018. Web search of fashion items with multimodal querying. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 342–350.
[22] Hung Le, Nancy F Chen, and Steven CH Hoi. 2022. Multimodal dialogue state tracking. *arXiv preprint arXiv:2206.07898* (2022).
[23] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086* (2022).
[24] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven C. H. Hoi. 2021. Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. *CoRR* abs/2107.07651 (2021). arXiv:2107.07651 https://arxiv.org/abs/2107.07651
[25] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. 2022. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10965–10975.
[26] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. 2016. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1096–1104.
[27] Ilya Loshchilov and Frank Hutter. 2017. Decoupled Weight Decay Regularization. https://doi.org/10.48550/ARXIV.1711.05101
[28] Corey Lynch, Kamelia Aryafar, and Josh Attenberg. 2015. Images Don't Lie: Transferring Deep Visual Semantic Features to Large-Scale Multimodal Learning to Rank. https://doi.org/10.48550/ARXIV.1511.06746
[29] Seungwhan Moon, Satwik Kottur, Paul A. Crook, Ankita De, Shivani Poddar, Theodore Levin, David Whitney, Daniel Difranco, Ahmad Beirami, Eunjoon Cho, Rajen Subba, and Alborz Geramifard. 2020. Situated and Interactive Multimodal Conversations. *CoRR* abs/2006.01460 (2020). arXiv:2006.01460 https://arxiv.org/abs/2006.01460
[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. *CoRR* abs/2103.00020 (2021). arXiv:2103.00020 https://arxiv.org/abs/2103.00020
[31] A. Rubio, LongLong Yu, E. Simo-Serra, and F. Moreno-Noguer. 2017. Multimodal joint embedding for fashion product retrieval. In *2017 IEEE International Conference on Image Processing (ICIP)*. 400–404. https://doi.org/10.1109/ICIP.2017.8296311
[32] Amazon Web Services. 2022. Amazon SageMaker. https://aws.amazon.com/sagemaker/
[33] Amazon Web Services. 2022. AWS Lambda. https://aws.amazon.com/lambda/
[34] Aman Shrivastava, Ramprasaath R. Selvaraju, Nikhil Naik, and Vicente Ordonez. 2021. CLIP-Lite: Information Efficient Visual Representation Learning from Textual Annotations. https://doi.org/10.48550/ARXIV.2112.07133
[35] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2019. Contrastive Multiview Coding. *CoRR* abs/1906.05849 (2019). arXiv:1906.05849 http://arxiv.org/abs/1906.05849
[36] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748 (2018). arXiv:1807.03748 http://arxiv.org/abs/1807.03748
[37] Tan Wang, Xing Xu, Yang Yang, Alan Hanjalic, Heng Tao Shen, and Jingkuan Song. 2019. Matching Images and Text with Multi-modal Tensor Fusion and Re-ranking. https://doi.org/10.48550/ARXIV.1908.04011
[38] Prashan Wanigasekara, Nalin Gupta, Fan Yang, Emre Barut, Zeynab Raeesy, Kechen Qin, Stephen Rawls, Xinyue Liu, Chengwei Su, and Spurthi Sandiri. 2022. Multimodal Context Carryover. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Association for Computational Linguistics, Abu Dhabi, UAE, 417–428. https://aclanthology.org/2022.emnlp-industry.42
[39] Prashan Wanigasekara, Kechen Qin, Emre Barut, Fan Yang, Weitong Ruan, and Chengwei Su. 2022. Semantic VL-BERT: Visual Grounding via Attribute Learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*. 1–8. https://doi.org/10.1109/IJCNN55064.2022.9892420
[40] Lilian Weng. 2021. Contrastive Representation Learning. *lilianweng.github.io* (May 2021). https://lilianweng.github.io/posts/2021-05-31-contrastive/
[41] Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogerio Feris. 2021. Fashion iq: A new dataset towards retrieving images by natural language feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11307–11317.
[42] Wenyi Wu, Karim Bouyarmane, and Ismail Tutar. 2022. Catalog Phrase Grounding (CPG): Grounding of product textual attributes in product images for e-commerce vision-language applications. (2022).
[43] Peng Xu, Xiatian Zhu, and David A Clifton. 2022. Multimodal learning with transformers: a survey. *arXiv preprint arXiv:2206.06488* (2022).
[44] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. CoCa: Contrastive Captioners are Image-Text Foundation Models. https://doi.org/10.48550/ARXIV.2205.01917
[45] Licheng Yu, Jun Chen, Animesh Sinha, Mengjiao Wang, Yu Chen, Tamara L Berg, and Ning Zhang. 2022. Commercemm: Large-scale commerce multimodal representation learning with omni retrieval. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4433–4442.

# A APPENDIX

## A.1 Limitations

Our architecture is language-specific to English; all of the reference queries and product titles are in English. The model will need to be

retrained, and there may need to be architecture changes in order for the model to be transferred to another language. This is because the current embedding modules may not be sufficient to embded languages which aren't in a roman script or handle code switching. Since we employed MTurkers to collect data, the referring queries may not accurately represent our intended demographic of users which is currently North America. Additionally, we observed that the dataset was biased since annotators frequently chose common words, e.g. "shoe" rather than "sneakers", to describe items in images. The curation and validation of visual expressions is a time-consuming and expensive process, which is why our dataset is constrained in size even though gathering more data can alleviate some of the limitations. In the future we can also develop data synthesis mechanisms to close some of these gaps.

## A.2 Ethics Statement

We must take into account fairness even though our solution has no unethical uses or dangerous wider effects. In our data collection, users may see images of people in addition to the products. We paid strong attention to how delicate questions, such as those which allude to protected characteristics or statements that use hateful or derogatory language should be handled. During data collections we had strong guidelines for MTurkers to not use specific protected keywords such as *woman, man, young, old* and we also ran a sensitive query filter after the data was collected to ensure our model doesn't see such queries.

## A.3 Related work

In recent years there has been an explosion of multimodal transformer based models. [18, 43] provide surveys and track the progression of models over time. These models can be used for text-to-image retrieval [6, 7, 15, 21, 31, 34, 37], multimodal ranking [13, 28], multimodal dialogue state tracking [22], visual grounding [12], visual question answering, multimodal natural language inference and a plethora of other tasks. More recent work show that multi-modal solutions can deliver competitive performance on different tasks, and multi-modal, multi-task training has been one of the common themes in current state-of-the-art techniques [1, 9, 11, 14, 17, 42, 45]. Even though these systems are the state-of-the-art, they tend to be large, complex, memory intensive, and they cannot be easily added to an existing system that is already in use. In other words, to start using these large systems, there would need to be major system revamps.

In the literature, there are various categorizations of multi-modal models. One is based on architecture design; single encoder, dual encoder [39], and encoder-decoder [44]. Another is by when information fusion occurs; early fusion, late fusion, and hybrid. Among these models CLIP [30], BLIP [23], GLIP [25], ALBEF [24] falls into the category of dual encoder, late fusion models that are trained using a contrastive objective [10, 35, 36, 40] at some stage in their learning process. They have been demonstrated to work at industry scale, able to be trained on massive datasets and have very low inference latency. Our work uses CLIP [30] and ALBEF [24] embeddings on what we define as a niche "Visual Item Selection" task in the context of a voice assistant. A comparable task to "Visual Item Selection" is described in the "Multimodal Learning to Rank" paper [28] but it does not take the full co-observed context into account and only focuses on pairwise ranking. To the best of the authors' knowledge, there is no off-the-shelf available solution that solves this niche "Visual Item Selection" task, yet it is an important task that has practical applicability for modern-day voice assistants.

In terms of datasets there are a few fashion image retrieval datasets like Fashion IQ [41], DeepFashion [26], yet they do not have item selection of the type of selecting 1 item from $n$ number of shown items which is the task of interest for this paper. The only other comparable datasets are SIMMC 1.0/2.0 [19, 29] that have interactions between a voice assistant and a user in Virtual Reality (VR). The users shop for fashion and furniture items in VR, and the interaction is given as a multi-turn dialogue between the user and the system. The conversation and image interactions of the VR datasets did not closely resemble our real word use case; thus we did not pursue the use of SIMMC 1.0/2.0. We conduct an Amazon Mechanical Turk survey of our own and collect data relevant to our use case.

In concurrent work, the authors of this paper published a paper on a multi-modal dialogue state tracking model [38], which is an extension to [8]. The main difference of the concurrent work is that the multi-modal context carryover model makes a binary prediction using an encoder-decoder type transformer model on which multi-modal context to carryover from one turn to the next. The model and methodology described in this paper is less complex than the concurrent work, and the shopping experience described in this paper is made available via an Alexa Skill in the Alexa Skills store.

## A.4 Alexa Skills Interaction Model

The interaction model defined in Alexa Skill Kit allows Alexa devices to interact with customers. When building skills, developers define how users can interact with the skill via the interaction model.

Besides using the interaction model, we create our visual item selection skill to support multiple images on screen by integrating with Alexa Presentation Language (APL) [2]. APL is a responsive layout language that let the Skill render visual content on Alexa-enabled multi-modal devices. To display visual content, we design the APL template in JSON format to provide the structure and layout on the screen. We collect product image URLs as our data source. When using the item selection skill, users will see three to six randomly sampled images on the screen and can select one of them through voice command.

## A.5 Multi-modal Contrastive Learning

The training requires having $N$ matched $(image, text)$ pairs where the text can be free form. The bidirectional contrastive losses for the $i^{th}$ image-text pair is given in equation 1 and equation 2. The image and text are projected onto a shared embedding space $\mathbf{I} \in \mathbb{R}^d$, $\mathbf{T} \in \mathbb{R}^d$ respectively. $\langle \mathbf{I}_i, \mathbf{T}_i \rangle$ represents the cosine similarity and $\tau \in \mathbb{R}^+$ is a temperature parameter. The losses are then added as seen in equation 3.

$$\mathcal{L}_i^{(image \rightarrow text)} = -\log \frac{\exp\left(\langle \mathbf{I}_i, \mathbf{T}_i \rangle / \tau\right)}{\sum_{k=1}^{N} \exp\left(\langle \mathbf{I}_i, \mathbf{T}_k \rangle / \tau\right)}, \tag{1}$$

$$\mathcal{L}_i^{(text \rightarrow image)} = -\log \frac{\exp\left(\langle \mathbf{T}_i, \mathbf{I}_i \rangle / \tau\right)}{\sum_{k=1}^{N} \exp\left(\langle \mathbf{T}_i, \mathbf{I}_k \rangle / \tau\right)}. \tag{2}$$

The final loss is a weighted combination of the two losses averaged over the training dataset of size $N$. Here $\lambda \in [0, 1]$ is a scalar weight.

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\lambda\mathcal{L}_i^{(image\rightarrow text)} + (1-\lambda)\mathcal{L}_i^{(text\rightarrow image)}\right). \quad (3)$$

## A.6 Additional Details on the Experimental Setup.

Our loss functions are derived from ALBEF. The ALBEF ("Align before Fuse") paper shows that an image-to-text contrastive alignment task ("align") performed before a multi-modal visual grounding task that has cross modal attention ("fuse") leads to better "grounded vision and language learning representations". Authors introduce a contrastive learning component (corresponding loss is "Image-Text Contrastive (ITC) Loss" defined as $\mathcal{L}_{ITC}$ ) which serves to create better aligned image-text representations that could do well on several downstream tasks (e.g., image-text retrieval, Visual Question Answering, Natural Language for Visual Reasoning). ALBEF model is pretrained with 2 additional objectives: Masked Language Modeling (MLM) loss ($\mathcal{L}_{MLM}$) and Image-Text Matching (ITM) loss ($\mathcal{L}_{ITM}$). In MLM, the image and textual embeddings are fused with a cross-modality attention and predict the masked tokens based on the multi-modal encoder output. ITM predicts whether text and image pair match each other or not. For this purpose [CLS] token of the multi-modal encoder output is used to represent the joint image-text pair. This representation is passed through a fully connected layer before calculating the probability of match as a 2-class prediction problem. Since pretraining data used by ALBEF contains only images and corresponding textual labels as positive pairs, the ALBEF authors use hard negative mining to get negative examples. For this purpose they reuse the intermediate similarities calculated for ITC loss.

To improve learning from noisy web data, the ALBEF authors introduce a "momentum distillation step" which treats the momentum models as teacher model to distill from. They suggest that since web data are inherently noisy, the momentum distillation step which involves a self-training method that learns from pseudo-targets produced by a momentum model helps to generate better negative samples and create a more stable representation.

For fine-tuning ALBEF for our paper, the loss associated with Task 1 in Section 4.3 (disregarding the momentum distillation terms) can be simply written as

$$\mathcal{L} = \mathcal{L}_{ITC} + \mathcal{L}_{ITM} + \mathcal{L}_{MLM} \quad (4)$$

whereas for Task 2 for ALBEF we drop $\mathcal{L}_{ITC}$ and $\mathcal{L}_{MLM}$ terms, but keep $\mathcal{L}_{ITM}$ and the negative image mining using similarities calculated with the help of momentum models.

For Task 3, we again drop the $\mathcal{L}_{ITC}$ and $\mathcal{L}_{MLM}$, but also the momentum models. In Task 2, although dropping the same terms, we kept using hard negative mining as in original ALBEF setup. However, for Task 3 (since all images associated with the co-observed context are given to the model) we refrain from using hard negative mining as we have an access to negative images directly. Therefore, we adapt ITM to directly use negative images coming from the data. We pass all image-text pairs through the multi-modal encoder and concatenate the output corresponding to the images given with same utterance into a single vector. Also the binary ITM head is replaced with multi-class classification where these concatenated vectors are fed in. For Task 3, we train ALBEF exclusively with cross-entropy loss

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_c^i \log\left(p\left(c|\left\langle \mathbf{I_i^1}, .., \mathbf{I_i^C}\right\rangle, \mathbf{T_i}\right)\right) \quad (5)$$

where $\left\langle \mathbf{I_i^1}, .., \mathbf{I_i^C}\right\rangle$ are the $C$ images (a fixed integer before the training, usually from $[3; 6]$) associated with the $\mathbf{T_i}$ text utterance and $y_c^i \in \{0, 1\}$ is the label associated with each image indicating whether it is a positive or a negative image for the text. All this loss functions are summarized in Table 1.

For ALBEF fine-tuning, we refrain from hyperparameter fine-tuning and reuse the same set of hyperparameters across our fine-tuning experiments. We use AdamW [27] optimizer with a weight decay of 0.02, cosine learning rate schedule with initial learning rate of 1e-5 and decrease it until 1e-6 with decay rate of 1 epoch. We do run each of three tasks for ALBEF over 10 epochs. We use effective batch size of 256 and resize the images to have 384x384 size. All fine-tuning tasks for ALBEF initialize the model from the weights of ALBEF model published with [24] that was trained on 14M images.

For CLIP fine-tuning for our paper, we use a pre-trained model from OpenAI [5] with ViT-32-B vision encoder. As our baseline, we use the model without fine-tuning for zero-shot inferences. Next, for Task 1, we fine-tune the model on the MTurk dataset using the losses given in Table 1, Task 1 (CLIP). We run the fine-tuning for 10 epochs, use a batch-size of 90, learning rate of 1e-3, weight decay of 0.1, and warm-up iterations of 10,000. Finally, as Task 3 we run a supervised fine-tuning on the MTurk dataset for CLIP. We modify the CLIP loss equation by adding a cross-entropy categorical loss as shown in Table 1, Task 3 (CLIP), essentially treating the problem as a classification problem. For all our experiments, we used an open-source implementation of CLIP model [16]. For both CLIP and ALBEF, fine-tuning is run on a p3.8xlarge EC2 instance running 4 Nvidia Tesla V100 GPUs.

## A.7 Additional details on Fine-tuning Task Evaluation

We select the public models published with CLIP [30] and ALBEF [24] as baseline for our experiments and do a zero-shot assessment of baseline models on "Visual Item Selection" system. For CLIP, this means that we take the dot product of all images and text embeddings produced by the corresponding encoders and pick the image with the largest score as the correct one. For ALBEF, we pair utterance with both positive and negative images and feed them to the model. Note that for zero-shot evaluation, as well as for Task 1 and 2, binary prediction is used in ALBEF and hence we do decision making by picking the image with the largest probability from independent decisions. However, for Task 3, we directly compute the probability for all classes with *softmax* and label the right image accordingly.

Received 7 November 2022