

# i-Code Studio: A Configurable and Composable Framework for Integrative AI

Yuwei Fang\*, Mahmoud Khademi\*, Chenguang Zhu, Ziyi Yang, Reid Pryzant, Yichong Xu, Yao Qian, Takuya Yoshioka, Lu Yuan, Michael Zeng and Xuedong Huang

Microsoft Cognitive Services Research Group

{yuwfan, mkhademi, chezhu}@microsoft.com

## Abstract

Artificial General Intelligence (AGI) requires comprehensive understanding and generation capabilities for a variety of tasks spanning different modalities and functionalities. Integrative AI is one important direction to approach AGI, through combining multiple models to tackle complex multimodal tasks. However, there is a lack of a flexible and composable platform to facilitate efficient and effective model composition and coordination. In this paper, we propose the i-Code Studio, a configurable and composable framework for Integrative AI. The i-Code Studio orchestrates multiple pre-trained models in a finetuning-free fashion to conduct complex multimodal tasks. Instead of simple model composition, the i-Code Studio provides an integrative, flexible, and composable setting for developers to quickly and easily compose cutting-edge services and technologies tailored to their specific requirements. The i-Code Studio achieves impressive results on a variety of zero-shot multimodal tasks, such as video-to-text retrieval, speech-to-speech translation, and visual question answering. We also demonstrate how to quickly build a multimodal agent based on the i-Code Studio that can communicate and personalize for users. The project page with demonstrations and code is at <https://i-code-studio.github.io/>

## 1 Introduction

Large language models (LLMs) such as BERT (Devlin et al., 2018) and GPT-3 (Brown et al., 2020), visual-language models (VLMs) like CLIP (Radford et al., 2021a) and DALL-E (Ramesh et al., 2021), and audio language models (ALMs) such as W2V-BERT (Chung et al., 2021) have enabled a variety of capabilities, from zero-shot image classification to reading comprehension, automatic speech recognition, and photorealistic image generation. The performance and capability of these pre-trained

models are, however, influenced by the data they are exposed to, which varies across different domains; LLMs are trained on diverse sources of data, such as webpages, novels, and Wikipedia corpora, while VLMs are trained on pairs of images or videos and their captions, and ALMs are trained on audio data such as speech. These distinct training

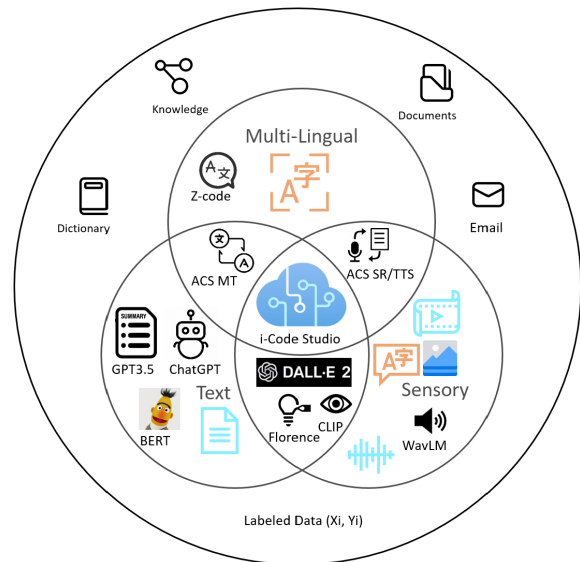


Figure 1: The i-code Studio is a configurable and composable architecture for integrative AI allowing developers to quickly and easily orchestrate various cutting-edge pre-trained models in a finetuning-free fashion.

domains render the pre-trained models different and sometimes complementary capabilities. For instance, LLMs are suitable for tasks such as reading comprehension but unable to interpret audio and visual information; VLMs can produce photorealistic images but cannot tackle complex language understanding. On the other hand, humans can often easily handle distinct tasks like the above with multimodal input and output. Therefore, in order to build Artificial General Intelligence (AGI), we need to break the barriers between modalities and specific tasks.

Instead of building a single model to handle all

Co-first authors.

possible tasks, which is infeasible under current technology, a lot of research has recently emerged to focus on the composition of large pre-trained models to achieve integrative AI, either via finetuning them jointly on new tasks (Yang et al., 2022; Hu and Singh, 2021; Wang et al., 2021b; Alayrac et al., 2022; Tewel et al., 2022), or via a shared modality, such as language, to capture new multimodal capabilities without the need for finetuning (Tewel et al., 2022; Zeng et al., 2022; Wang et al., 2022; Li et al., 2022). Issues with these approaches are that 1) there often lacks data and computation resources for joint finetuning, and 2) one cannot easily configure and compose different large pre-trained models in an agile framework to adapt to different needs. Therefore, in this paper, we propose the i-Code Studio, a configurable and composable framework for integrative AI (Figure 1). The i-Code Studio allows developers to quickly and easily orchestrate various cutting-edge pre-trained models in a finetuning-free fashion.

These pre-trained models are from different modalities, and the strength of each individual model is integrated to conduct complex multimodal tasks. For each task, a directed acyclic graph (DAG) is configured so that the related models cooperate to produce the desired output. The input data flows through each node in the DAG, enabling complex multimodal tasks to be completed. This makes i-Code Studio an integrative, flexible, and composable framework. For instance, for visual question answering task, a DAG is configured using the input image, the input question, the Florence (Yuan et al., 2021) vision foundation model, a language prompt, the ChatGPT, and an output, each represented by a node. The visual information from the input image is fed into Florence. The Florence node processes the image and outputs a set of detected object categories/tags and a caption. These outputs and the input question are then fed into a node that generates a VLM-informed language prompt. Finally, this cross-modal prompt is used by ChatGPT to generate an answer to the input question which is sent to the output node.

In this paper, we showcase the effectiveness of the i-Code Studio using models from Azure Cognitive Services (ACS) and OpenAI services. The resulting integrative model achieves the state-of-the-art (SOTA) or comparable to the SOTA performance on zero-shot tasks such as speech-to-speech translation, video-to-text retrieval, and visual ques-

tion answering. We also show how to quickly build a multimodal agent to interact with a user. In summary, our main contributions are the following:

- (1) We propose i-Code Studio, a new integrative, configurable, and composable framework which can be used to compose various pre-trained models.
- (2) We show how i-Code Studio can achieve impressive results on a variety of zero-shot multimodal tasks, e.g. video-to-text retrieval, speech-to-speech translation, and visual question answering.
- (3) We utilize i-Code Studio to build a multimodal agent that can communicate and personalize for users by leveraging ACS and OpenAI services.

## 2 Related Work

Recently, the composition of large pre-trained models has been extensively studied. The most common way to compose these models is to fine-tune them jointly on new tasks. Hu and Singh (2021) proposed UniT, a Unified Transformer model that is capable of learning several tasks across multiple domains, including object detection and multimodal reasoning. This model is based on a transformer encoder-decoder architecture, where each input modality is encoded with an encoder, and shared decoders are used to make predictions for each task. Wang et al. (2021b) proposed a Vision-Language Pretraining framework, called SimVLM that is trained end-to-end with a single language modeling objective. The SimVLM reduces the complexity of training by utilizing weak supervision on a large scale. Alayrac et al. (2022) proposed Flamingo, a collection of VLMs that can connect pre-trained vision-only and language-only models, process sequences of interleaved visual and textual data, and accept images or videos as inputs. However, these methods can be computationally expensive. The i-Code Studio differs from these approaches since it does not require finetuning, which enables the fast composition of pre-trained models for a variety of tasks and reduces the time and expense associated with finetuning.

Unlike these work, models can be composed via a shared modality, such as language. Tewel et al. (2022) combined a visual-semantic model with a large language model, enabling the models to take advantage of the knowledge present in both web-scale models for image caption generation task. More related to our work, Zeng et al. (2022) proposed Socratic Models, a modular framework that enables multiple pre-trained models to exchange

information with each other, capture new multimodal capabilities without the need for finetuning, and be composed without any prior training using multimodal-informed prompting. Our i-Code Studio is a more integrative, flexible, and composable framework compared to these work, allowing users to compose cutting-edge models and technologies customized for their particular needs easily.

Distinct from the work mentioned, Li et al. (2022) proposed a closed-loop approach to combining pre-trained models in such a way that they act as generators and scorers. The generators create proposals, while the scorers provide feedback to improve the generated results. This type of iterative consensus optimization allows models to correct mistakes made by other models, leading to significant improvements in downstream tasks. (Huang et al., 2022) studied the application of LLMs in embodied environments for robotic control. They combined LLMs with different sources of text feedback and found that natural language acts as a universal means of communicating with the model. The resulting system, called Inner Monologue, integrates various components such as perception models, robotic skills, and human feedback to effectively execute user commands.

### 3 The i-Code Studio Framework

In this section, we introduce i-Code Studio, a configurable and composable framework for integrative AI. Given a complex multimodal task, the i-Code Studio provides a generic framework for developers to quickly and easily integrate and compose several large pre-trained models and services across different modalities without any training or finetuning to accomplish the task. Figure 2 shows examples of building AI solutions for various multimodal tasks using the i-Code Studio framework. For each task, the framework can be represented via a DAG, where the nodes with no incoming edge are the raw input data such as image, text, video and speech, the nodes with no outgoing edges are the outputs of the given task, and the rest of the nodes are foundation models/services or hold intermediate model outputs from other models/services. The input to a node comes from the raw input, and/or the output from previous nodes. The input data flows through each node in the DAG, enabling complex multimodal tasks to be completed. An outgoing edge from a model/service node represent an API provided by the model/service. For each

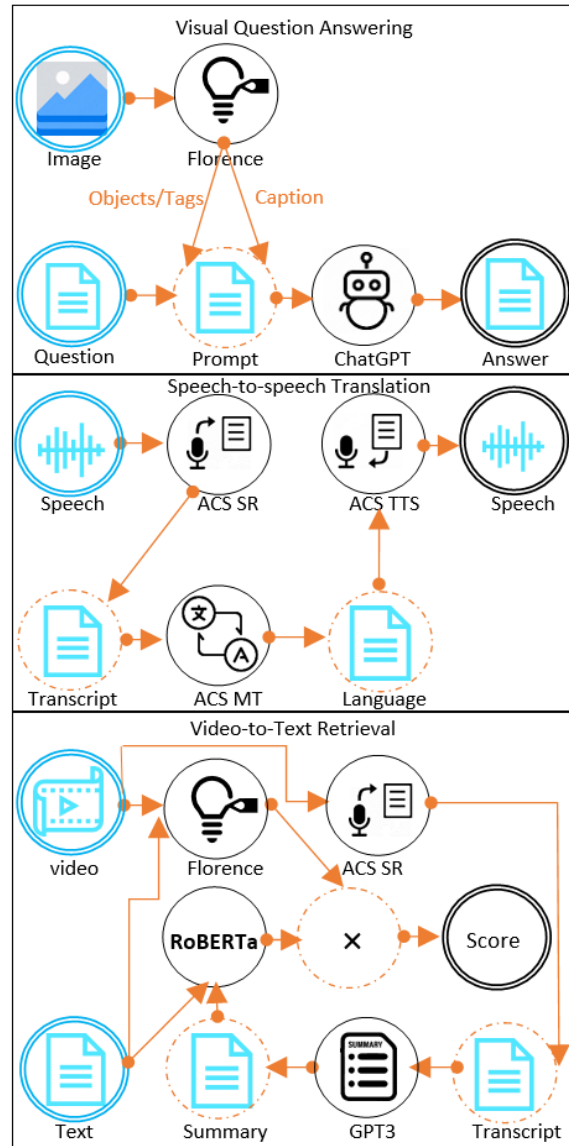


Figure 2: The i-Code Studio can be used to build AI solutions for various multimodal tasks. For each task, a DAG is configured so that the related models cooperate to produce the desired output. The input data flows through each node in the DAG, enabling complex multimodal tasks to be completed. The input nodes are represented by double blue circles, the model/service nodes, e.g. ChatGPT and Florence, by black circles, the output nodes by double black circles, and the rest by dash-dotted red circles. See the text for details about each multimodal task.

task, the inputs enter the DAG from the input nodes, and are processed by one or more models or model services. In the process, edges convert the format of a module’s output, filters data, or apply an API such as summarization, translation, object detection, image captioning, transcribing, text-to-speech synthesis, etc.

For each task, a DAG is configured so that the

related models cooperate to produce the desired output. The different components of i-Code Studio cooperate seamlessly to form a single, integrated AI solution, and can be adjusted to fit the specific needs of the user. For instance, for visual questions answering (VQA) task the input is an image and a question related to the image (see Figure 2). We can first apply image captioning and object detection services to the input image. The output text, which contains the visual information, is merged with the input question as the prompt to ChatGPT, which answers the question. For speech-to-speech translation, the DAG is configured with Speech Recognition (SR)  $\rightarrow$  Machine Translation (MT)  $\rightarrow$  Text-To-Speech (TTS). This DAG transcribes the source speech, translates the transcription into the target language, and generates the target speech.

To build i-Code Studio, we utilize Azure Machine Learning Studio, a cloud-based, collaborative, drag-and-drop development environment for building, testing, and deploying machine learning models. We encapsulate available models and services from Azure Cognitive Services (ACS) as independent APIs and deploy them as an integrated web service for real-time invoking. In this way, it allows developers to flexibly combine them to build their own applications. More details about the available foundation models and services are presented in Appendices A and B.

## 4 Evaluations

In this section, we present our experiments in three tasks covering language, speech and vision modality: 1) video-to-text retrieval; 2) visual question answering and 3) speech-to-speech translation.

### 4.1 Video-to-Text Retrieval

Video-to-Text retrieval task is to select the most relevant text from a pool of candidates given the video, which typically involves all modalities across language, vision and speech. Thus, it can be an ideal task to test the capabilities of i-Code Studio. Following Zeng et al. (2022), the pipeline is organized into the following steps: (i) calculate the similarity score  $s_1$  between the average vision features of video and text features of captions via ACS Vision service (Yuan et al., 2021); (ii) calling ACS Speech service to transcribe the video to text; (iii) summarize the transcript with Azure OpenAI services using GPT-3 (Brown et al., 2020); (iv) compute a text-based similarity score  $s_2$  between the gen-

Method	R@1	R@5	R@10
JMEC (Mithun et al., 2018)	12.5	32.1	42.4
Finetuned Collab. Experts (Liu et al., 2019)	15.6	40.9	55.2
CLIP2Video (Fang et al., 2021)	<b>54.6</b>	<b>82.1</b>	<b>90.8</b>
CLIP (Portillo-Quintero et al., 2021)	40.3	69.7	79.2
Zero-shot SMs (Zeng et al., 2022)	44.7	71.2	80.0
i-Code Studio	<b>49.8</b>	<b>74.8</b>	<b>82.2</b>

Table 1: Video-to-text retrieval results on MSR-VTT (Xu et al., 2016) dataset.

erated summary and the captions with pre-trained language model; (v) compute the final relevance score  $s = s_1 \times s_2$ , combining vision-text based score and speech-text based score; (vi) select the text with the highest relevance score as answer.

Table 1 shows our results on MSR-VTT (Xu et al., 2016), which is the most popular large-scale dataset for video-to-text retrieval and consists of 10,000 video clips from 20 categories, and each video clip is annotated with 20 English sentences by Amazon Mechanical Turks. We use the standard recall metrics for evaluation and compare our approach with both finetuned and zero-shot methods. We can see that in zero-shot setting, i-Code Studio outperforms previous state-of-the-art (SOTA) SMs by 5.1 points in R@1, thus achieving the new SOTA in this setting. Compared with finetuned approach, i-Code Studio significantly narrowed the gap between the zero-shot and fine-tuned approach, showing the promising of the zero-shot approach.

### 4.2 Visual Question Answering

The i-Code Studio can be used to answer visual questions (see Figure 3). Specifically, Azure Cognitive Services’ Florence (Yuan et al., 2021) is used to zero-shot detect a set of object categories in the input image, generate a set of tags associated to it, and create a caption that describes the image. These descriptions and the input question are then used to form a VLM-informed language prompt, which is fed into ChatGPT to predict an answer. We evaluated i-Code Studio’s performance on the FVQA dataset (Wang et al., 2017) for the visual question answering task. FVQA is a VQA dataset that mostly contains questions requiring external knowledge to answer, and provides supporting fact triplets alongside the image-question-answer triplets. Following (Wang et al., 2017), we used 1,090 test images, amounting to 2,899 questions. Our results are presented in Table 2. The i-Code Studio significantly outperforms Fact-based VQA without the support facts from the dataset,



likely due to the power of Florence’s vision foundation model and ChatGPT’s capability to answer questions requiring external knowledge.

Method	Accuracy
Human	77.99
Fact-based VQA (Wang et al., 2017)	56.91
Fact-based VQA (Ensemble) (Wang et al., 2017)	58.76
i-Code Studio	<b>60.59</b>

Table 2: VQA results on FVQA dataset.

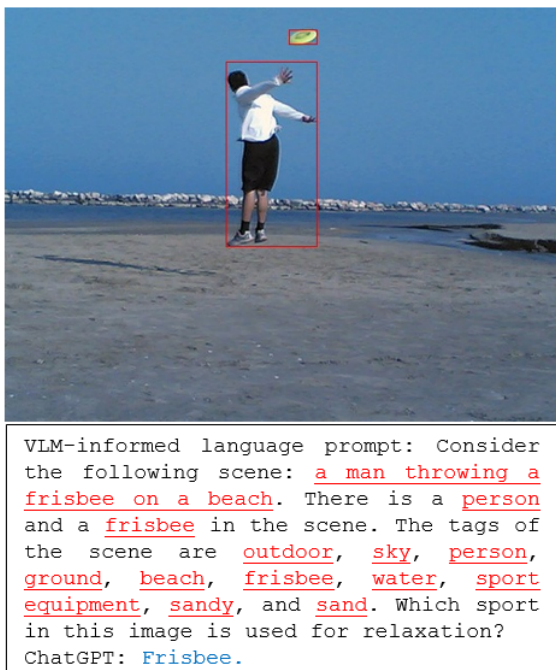


Figure 3: VQA with i-Code Studio: a VLM-informed language prompt is created using Florence outputs and input question. The red underlined text are the caption, object categories, and tags detected by Florence. The prompt is then fed into ChatGPT to predict an answer.

### 4.3 Speech-to-speech Translation

Speech-to-speech translation task consists of transcribing spoken language into text, translating the text into another language, and then generating speech in the target language. We use this task to evaluate the multilingual and speech capabilities of i-Code Studio. Specifically, we first leverage ACS Speech Recognition service to transcribe the incoming speech, then use ACS Language Machine Translation service to translate in the target languages, and finally call ACS Text-To-Speech to synthesize the speech in the target languages.

We evaluate i-Code Studio on CVSS (Jia et al., 2022) dataset, a massively multilingual-to-English

Model	All	Hi-Res	Lo-Res
Li et al. (2021) (Scratch-BL)	-	14.8	-
Wang et al. (2021a) (A2A-L)	7.5	24.0	3.7
Wang et al. (2021a) (A2E-M, arXiv)	-	24.5	-
Jia et al. (2022)	11.0	29.4	6.7
Jia et al. (2022) (ASR pre-training)	13.3	31.4	9.0
i-Code Studio	35.8	39.7	34.8

Table 3: Speech-to-text evaluation results on CVSS dataset. We call ACS Speech Recognition, ACS Machine Translation, and ACS Text-to-Speech services in a cascade approach. Hi-Res and Lo-Res stand for high-resource and low-resource languages respectively.

speech-to-speech translation corpus. It covers sentence-level parallel speech-to-speech translation pairs from 21 languages into English and is derived from the Common Voice speech corpus (Ardila et al., 2020) and the CoVoST 2 (Wang et al., 2020) speech-to-text translation corpus. The translation speech in CVSS is synthesized with two state-of-the-art TTS models trained on the LibriTTS corpus. As the speech generation quality is measured by human in mean opinion score (MOS) on naturalness and speaker similarity metrics, here we only report translated text result in BLEU metric using SacreBLEU with its default configuration. Following Jia et al. (2022), we group the evaluation results on high-resource source languages (French, German, Catalan and Spanish) and low-resource ones (all the rest). From Table 3, we can see the i-Code Studio outperforms previous SOTAs significantly by 22.5 points on average. The improvement of high-resource languages still has about 8.3 points, demonstrating the strong capabilities of the i-Code Studio framework.

## 5 Applications: Multimodal Agents

As humans, we have a complex sensory system that allows us to experience the world around us. We use our eyes to see, ears to hear, mouths to talk, and brains to process and interpret the information we receive. Inspired by this, we utilize i-Code Studio to build a multimodal agent that can communicate and personalize for users. Specifically, the eyes of the agent use Azure Vision services to interpret visual images signals and send signals to the brain; the ears and mouth use Azure Speech services to collect sound waves and produce sounds; the brain leverage Azure OpenAI services to integrate all the sensory signals received from the eyes, ears and uses them to make decisions. This interconnected

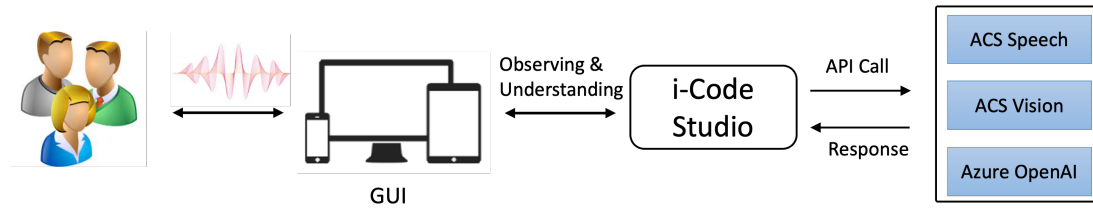


Figure 4: An overview of the multimodal agent which is built using the i-Code Studio.

The following is a conversation with an AI assistant. The assistant can interact with the human by seeing.

AI: Hello!

Human: Hey, nice to meet you! (Seeing: A man holding a backpack. Amazon, Backpack)

AI: Hi! Are you working in Amazon?

Human: Yes, how do you know?

AI: I guess it from your backpack.

Human: Can you guess something from me? (Seeing: A man playing piano. Piano)

AI: You must be a music artist.

Human: Yes. How did you find it?

AI: Because you are playing a piano.

Human: Thank you for your assistance. Have a nice day!

AI: You're welcome! Have a nice day too!

AI: Hello, I am a multimodal agent created by Azure Cognitive Services. How can I help you today?

Human: Hi! Nice to meet you.

(a) (Seeing: A man wearing glasses. Microsoft, Glasses, T-shirt)

AI: Hi! Are you working in Microsoft?

Human: Yes, how do you know?

(Seeing: A man wearing glasses. Microsoft, Glasses, T-shirt)

AI: I can see from your t-shirt.

Human: When was Microsoft founded?

(Seeing: A man wearing glasses. Microsoft, Glasses, T-shirt)

AI: Microsoft was founded on April 4, 1975.

Human: Then, how old would Microsoft be in 2023?

(Seeing: A man wearing glasses. Microsoft, Glasses, T-shirt)

AI: Microsoft will be 48 years old in 2023.

Human: Can you recognize anything from me?

(b) (Seeing: A man holding a violin. Violin, Glasses)

AI: You must like music.

Human: How do you know?

(Seeing: A man holding a violin. Violin, Glasses)

AI: Because you are holding a violin.

Human: Thank you! Have a nice day!

(Seeing: A man wearing glasses. Microsoft, Glasses, T-shirt)

AI: You're welcome! Have a nice day too!

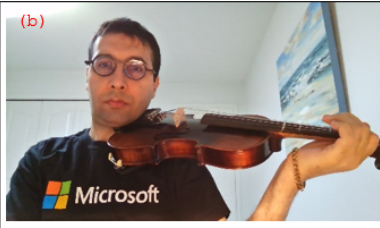



Figure 5: The i-Code Studio can be used to build a multimodal virtual assistant. During the conversation the user input and history context are prepended with the captions/tags from Florence vision (shown in red) and fed as input into GPT-3. The bottom boxes show the conversation as well as two snapshots of the input video from the camera.

system of sensory organs and the brain are what enables our multimodal agents to understand and interact with the world around us. Our multimodal agent is a virtual assistant with “eyes” (Florence), “ears” (ACS ASR), “brain” (e.g. ChatGPT and GPT-3) and mouth (ACS TTS). The i-Code Studio integrates speech and vision signals from users by composing and configuring services from ACS and OpenAI. Figure 5 shows a demo example. Using VLM-informed language prompting, i-Code Studio can enable multimodal dialogue between the user and agent. GUIs call i-Code Studio once to simplify the developing cost while giving consistent user experience.

## 6 Conclusion

The i-Code Studio, is a new configurable and composable framework for Integrative AI. It orchestrates multiple pre-trained models to conduct complex multimodal tasks, without the need for finetuning. We showed the i-Code Studio can achieve impressive results on three multimodal tasks. We also demonstrated how to build a multimodal virtual assistant agent with the i-Code Studio. With further research and development, the i-Code Studio can be extended to be more flexible and powerful to create even more complex applications.

## 7 Screencast Video

In this section, the public link to one of our example demos for the multimodal agent is provided<sup>1</sup>.

## 8 Limitations

The i-Code Studio currently relies on a limited number of pre-trained models and services. While this is sufficient for many multimodal tasks, the framework needs additional services to support more complex multimodal tasks. Moreover, to demonstrate the capabilities of the i-Code Studio, we need to apply the framework to more complex multimodal tasks such as meeting summarization and image generation from textual descriptions.

## References

- 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*.
- Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. **Common voice: A massively-multilingual speech corpus**. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.
- Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. 2022. **Wavlm: Large-scale self-supervised pre-training for full stack speech processing**. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE.
- 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*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **BERT: Pre-training of deep bidirectional transformers for language understanding**. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. 2021. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. **Deberta: Decoding-enhanced bert with disentangled attention**. In *International Conference on Learning Representations*.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.
- Ronghang Hu and Amanpreet Singh. 2021. Unit: Multimodal multitask learning with a unified transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1439–1449.
- Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. 2022. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.
- Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR.
- Ye Jia, Michelle Tadmor Ramanovich, Quan Wang, and Heiga Zen. 2022. CVSS corpus and massively multilingual speech-to-speech translation. In *Proceedings*

<sup>1</sup>[https://drive.google.com/file/d/101EZQ9LbQpR\\_kc8zsmenRXjCQejS0-G3/view?usp=share\\_link](https://drive.google.com/file/d/101EZQ9LbQpR_kc8zsmenRXjCQejS0-G3/view?usp=share_link)



- of Language Resources and Evaluation Conference (LREC)*, pages 6691–6703.
- Shuang Li, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, and Igor Mordatch. 2022. Composing ensembles of pre-trained models via iterative consensus. *arXiv preprint arXiv:2210.11522*.
- Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. [Multilingual speech translation from efficient finetuning of pretrained models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 827–838, Online. Association for Computational Linguistics.
- Jian Liang, Chenfei Wu, Xiaowei Hu, Zhe Gan, Jianfeng Wang, Lijuan Wang, Zicheng Liu, Yuejian Fang, and Nan Duan. 2022. [NUWA-infinity: Autoregressive over autoregressive generation for infinite visual synthesis](#). In *Advances in Neural Information Processing Systems*.
- Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman. 2019. Use what you have: Video retrieval using representations from collaborative experts. In *arXiv preprint arxiv:1907.13487*.
- Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metzger, and Amit K. Roy-Chowdhury. 2018. [Learning joint embedding with multimodal cues for cross-modal video-text retrieval](#). In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, ICMR '18*, page 19–27, New York, NY, USA. Association for Computing Machinery.
- Jesús Andrés Portillo-Quintero, José Carlos Ortiz-Bayliss, and Hugo Terashima-Marín. 2021. [A straightforward framework for video retrieval using clip](#). In *Pattern Recognition: 13th Mexican Conference, MCPR 2021, Mexico City, Mexico, June 23–26, 2021, Proceedings*, page 3–12, Berlin, Heidelberg. Springer-Verlag.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021a. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021b. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR.
- Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*.
- Yoad Towel, Yoav Shalev, Idan Schwartz, and Lior Wolf. 2022. Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928.
- Changhan Wang, Anne Wu, Jiatao Gu, and Juan Pino. 2021a. Covost 2 and massively multilingual speech translation. In *Interspeech*, pages 2247–2251.
- Changhan Wang, Anne Wu, and Juan Pino. 2020. Covost 2 and massively multilingual speech-to-text translation. *arXiv preprint arXiv:2007.10310*.
- Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. 2023. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*.
- Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. 2017. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2413–2427.
- Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, et al. 2022. Language models with image descriptors are strong few-shot video-language learners. *arXiv preprint arXiv:2205.10747*.
- Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. 2021b. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*.
- Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296.
- Ziyi Yang, Yuwei Fang, Chenguang Zhu, Reid Pryzant, Dongdong Chen, Yu Shi, Yichong Xu, Yao Qian, Mei Gao, Yi-Ling Chen, et al. 2022. i-code: An integrative and composable multimodal learning framework. *arXiv preprint arXiv:2205.01818*.
- Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. 2021. Florence:



A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*.

Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Ayelet Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, et al. 2022. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

## A Foundation Models

Foundation models, first introduced by [Bommasani et al. \(2021\)](#), refer to any model that is pre-trained on broad data at scale and can be adapted to a wide range of downstream tasks. As a general paradigm of AI, foundation models have shown impressive performances and generalization capabilities in various modalities ([Brown et al., 2020](#); [Radford et al., 2021b](#); [Yuan et al., 2021](#)).

**Large Language Models** Large language models (LMs), trained on massive text collections such as BERT ([Devlin et al., 2019](#)), GPT-2 ([Radford et al., 2019](#)), DeBERTa ([He et al., 2021](#)), achieve state-of-the-art performances on many natural language processing benchmarks. More recent works, like GPT-3 ([Brown et al., 2020](#)), OPT ([Zhang et al., 2022](#)), PaLM ([Chowdhery et al., 2022](#)), Chinchilla ([Hoffmann et al., 2022](#)), have shown surprising emergent capabilities to generate text and can be “prompted” to perform a range of language tasks given zero or few examples of the task as input. In the i-Code Studio framework, we include three language-based foundation models to support diverse tasks and applications: Z-Code <sup>2</sup> for multilingual tasks like machine translation, GPT-3 ([Brown et al., 2020](#)) and ChatGPT <sup>3</sup> for general NLP tasks like text summarization and question answering.

**Vision Language Models** Vision language models (Vision LMs), trained on web-scale image-text and video data, such as CLIP ([Radford et al., 2021b](#)), ALIGN ([Jia et al., 2021](#)), DALL-E ([Ramesh et al., 2021](#)), Imagen ([Saharia et al., 2022](#)) and Nuwa-infinity ([Liang et al., 2022](#)), demonstrate superior performance on various computer vision tasks, such as classification, retrieval, object detection, VQA, image caption, video retrieval and action recognition. In Azure Cognitive Services, the Project Florence <sup>4</sup> is initiated to advance state-of-the-art computer vision technologies and develop the next-generation framework for visual recognition. Specifically, Florence ([Yuan et al., 2021](#)) is trained on noisy Web-scale data end-to-end with a unifying objective, allowing the model to achieve state-of-the-art performances across a

<sup>2</sup><https://www.microsoft.com/en-us/research/project/project-zcode/>

<sup>3</sup><https://chat.openai.com/>

<sup>4</sup><https://www.microsoft.com/en-us/research/project/projectflorence/>

wide range of benchmarks. In i-Code Studio, Florence is utilized as the vision foundation model.

**Audio Language Models** Audio language models leverage discretized audio tokens/codes to train a model by using a language modeling task, such as w2v-BERT ([Chung et al., 2021](#)), WavLM ([Chen et al., 2022](#)), and Vall-E ([Wang et al., 2023](#)), and bring significant improvements for various speech processing tasks like speech-to-text, text-to-speech, speaker recognition/diarization, speech separation, etc. In Azure Cognitive Speech Services, speech models were trained by using more than a few hundred of thousand hours of speech audio in a manner of supervised learning.

## B Machine Learning Services

A machine learning service is usually built on top of the foundation models, provide a comprehensive suite of cloud-based artificial intelligence (AI) and machine learning (ML) tools and services. These tools provide developers with easy-to-use, pre-built algorithms and APIs that can be integrated into a wide range of applications. The i-Code Studio adopt Azure Cognitive Services<sup>5</sup>, which provides a variety of models and services for different modalities. Developers can easily leverage Azure Cognitive services to add intelligence features to their applications, such as sentiment analysis, object detection, speech recognition and text-to-speech, without having to build the AI models from scratch

We include the following services for each modality in one framework so that our architecture can flexibly enable complicated applications that are difficult to create with an end-to-end approach and meanwhile provide users with a consistent experience. The i-Code Studio adopts the design of prompt learning [cite] to quickly adapt the architecture to different tasks through informed multimodal prompting with just a few labeled examples.

**Language** Azure Cognitive Services for Language (ACS Language) is a cloud-based service that provides Natural Language Processing (NLP) features for understanding and generation by using REST APIs and client libraries. Using Z-Code as the backbone, the language services provide the following functionalities: natural language understanding, question answering, text summarization

<sup>5</sup><https://azure.microsoft.com/en-us/products/cognitive-services/#overview>

and machine translation. Besides, we also integrate Azure OpenAI Services which use ChatGPT, GPT-3, Codex and Embeddings from OpenAI as the backbone to enable new reasoning and comprehension capabilities for building cutting-edge applications. Specifically, in our architecture, we include three language APIs: *(i)* machine translation: translating text from one language to another. This can be used to realize multilingual communication between human and machines. *(ii)* ChatGPT: an interactive dialogue language model; *(iii)* GPT-3: capable of a wide range of NLP tasks such as text generation, translation, summarization and question answering.<sup>6</sup>

**Speech** Azure Cognitive Speech Service (ACS Speech) provides speech capabilities with an Azure Speech resource. It can accurately transcribe multilingual speech-to-text, produce text-to-speech with real human-like voices, translate spoken audio, and correctly identify the speakers in conversations. We integrate two speech APIs in our architecture: *(i)* Speech-to-Text, to transcribe your speech to text in real-time or to transcribe recorded audio files to text; *(ii)* Text-to-Speech, to convert input text into synthetic speech in real-time or to generate audio files from text with either prebuilt or customized natural voice.

**Vision** Azure Cognitive Services for Vision (ACS Vision) are a set of services offered by Microsoft Azure that allow developers to add computer vision capabilities to their applications. It provides a range of services for tasks such as object detection and recognition, image analysis, optical character recognition (OCR), and facial recognition. We integrate two vision APIs in our architecture: *(i)* object detection: identify objects in an image and locate the bounding box within the frame. *(ii)* image captioning: generate a description of an entire image in human-readable language, using complete sentences.

---

<sup>6</sup>For GPT-3, We use text-davinci-003 model for downstream tasks and applications.