Automatic Spoken Language Acquisition Based on Observation and Dialogue

Ryota Komatsu *, Shengzhou Gao, Wenxin Hou, Mingxin Zhang, Tomohiro Tanaka, Keisuke Toyoda, Yusuke Kimura, Kent Hino, Yu Iwamoto, Kosuke Mori, Takuma Okamoto *, Member, IEEE, and Takahiro Shinozaki *, Member, IEEE

Abstract—Human babies are born without knowledge of any specific language. They acquire language directly from observation and dialogue without being limited by the availability of labeled data. We propose spoken language acquisition agents that simulate the process. Such an ability requires multiple types of learning, including (1) word discovery, (2) symbol grounding, (3) message generation, and (4) pronunciation generation. Several studies have targeted one or combined learning types to elucidate human intelligence and aimed to equip spoken dialogue systems with human-like flexible language learning ability. However, their language ability was partially lacking some of the components. Our agents are the first to integrate them all. Our key concept is to design an architecture to integrate unsupervised, self-supervised, and reinforcement learning to utilize clues naturally existing in raw sensory signals and drive the learning based on the agent’s intrinsic motivation. Experimental results show agents successfully acquire spoken language from scratch by interacting with an environment to act by speaking. Our proposed focusing mechanism significantly improves learning efficiency. We also demonstrate that our agents can learn neural vocoder and the concept of logical negation as a part of language acquisition.

Index Terms—Autonomous agent, reinforcement learning, self-supervised learning, spoken language acquisition, unsupervised learning.

I. INTRODUCTION

Learning and using spoken language is an essential component of human intelligence. However, the mechanism that enables humans to acquire spoken language from scratch remains mysterious. One prominent and widely accepted explanation, proposed by Skinner in 1957 [1], is that children develop a language based on behaviorist reinforcement principles by associating words with meanings. Chomsky refuted his theory [2], [3] saying it was not true that children learn a language only through detailed feedback from adults. He considered that children acquire much of their verbal behavior by casual observation and imitation of adults and other children. While there are many reasonable discussions, the true answer is so far an open question due to the lack of a quantitative and verifiable mathematical model.

What we observe in baby development is that they start with the generation of non-speech sounds [4], [5]. Gradually, they produce vowel-like sounds and then begin canonical babbling that combines a consonant and a vowel. About a year after birth, they pronounce the first word. Vocabulary growth is usually slow at first and increases from around one and a half years old [6]. They learn to learn words and gradually start making sentences combining multiple words.

From a modern machine learning perspective, humans are learning agents [7] with intrinsic motivation [8]. We have internal states that are not directly observable by others unless we express them, such as preference, appetite, thought, and knowledge. Spoken conversation is a means of mutual expression and observation of the internal states, and we often need it to satisfy our motivation. When language acquisition is considered as a reinforcement learning problem, the dimension of the action space of speech waveforms (i.e., utterances) is very high, which is challenging to address. It is virtually impossible to achieve convergence if we directly treat the speech waveform signal as the action space, requiring some parameterization or constraint. On the other hand, unsupervised learning from raw observation signals suffers from low accuracy.

When we consider how we efficiently acquire a spoken language, we can decompose the learning into multiple components, including:

1) Word discovery from the raw speech signal
2) Grounding of words and meanings
3) Language encoded message generation as an action for the current internal and external states driven by intrinsic motivation
4) Pronunciation generation to express the message.

We conjecture that the interaction of these components is essential. Some animals have the partial ability of these. For example, some dogs can learn and recognize human words and follow spoken instructions [9], which requires the first and the second ability. Some birds can imitate human utterances [10], which corresponds to the fourth ability, while they can not...
associate the pronunciation with the meaning. Only humans have all the abilities and develop outstanding social networks based on spoken conversation.

This paper aims to realize the first autonomous spoken language acquisition agent that supports all the learning aspects combining unsupervised, self-supervised, and reinforcement learning. It learns spoken language through observation and dialogue based on intrinsic motivation without assuming any specific language knowledge. While it is an initial stage, it involves a paradigm shift of system development. Instead of preparing a labeled speech corpus and training the system to infer correct output given the input, we first design the agent’s intrinsic motivation and a general learning mechanism and then let the agent freely interact with the environment or the dialogue partner. Agents with such a mechanism will adapt and behave flexibly in changing or unexpected environments, which is vital for future human-symbiotic robots.

This paper is an extension of our conference publications [11]–[14]. In [11], we proposed our first version of an open vocabulary spoken language acquisition agent. The agent utilized unsupervised word learning to discretize utterance space for reinforcement learning. The agent could successfully learn command words. However, the learning was not efficient, especially at the beginning of reinforcement learning, since it tried all possible word candidates uniformly. In [12], we proposed a focusing mechanism that adopted unsupervised sound-image grounding to improve the learning efficiency. Using the focusing mechanism, the agent could efficiently investigate word candidates paying attention to those words related to the current situation. The agent had a limitation in that it only observed images in the dialogue phase. To remove the limitation, we added a sound front-end to the agent in [13] so that the agent could listen to spoken questions. In the experiment, we also demonstrated that the agent could learn the concept of logical negation as a part of the process to satisfy its internal motivation without explicit programming of logical understanding. The discretized utterance space used in these agents was a list of waveform segments which we refer to as a sound dictionary. The agent’s pronunciation was a replay of the waveform segments, and the flexibility of the pronunciation was limited. To realize more flexible pronunciations, we replaced the sound dictionary with a neural vocoder in [14]. In this paper, we re-design our SPOken Language ACQuisition agent (Spolacq) and its implementation for improved generality and have run new experiments. The Spolacq software toolkit and our recorded data are freely available at our repository.

We organize the rest of the paper as follows. We first survey existing language acquisition studies in Section II and review some component learning modules that we use in our agent in Section III. We define evaluation tasks in Section IV. Then we explain our agents in Section V. For the evaluation, we describe our data set in Sections VI, and the experimental setup in Section VII. Section VIII shows the results. Finally, we summarize the paper and discuss the future directions in Section IX.

II. RELATED WORKS

As a pioneering constructive approach with spoken language acquisition from an engineering standpoint, Gorin et al. developed an automatic call-routing system that detected new words in speech utterance and added them to the recognition vocabulary [15], [16]. It applied an out-of-vocabulary detection and worked with speech utterance and call routing pairs without using text labels. While the recognition vocabulary was open, the available call routing options were pre-defined as an action space, where the number of routing options or the actions was three. Iwahashi implemented an advanced physical robot system that used a hidden Markov model (HMM) and stochastic context-free grammar (SCFG) [17]. The system learned vocabulary from isolated word pronunciations and discovered an association between words and images from multimodal observation based on mutual information. The robot acted by moving an arm based on a heuristic program, combining the imitation-based learning results. The actions for the system were selecting a target object and generating the trajectory for the arm movement. Roy et al. proposed a system to learn vocabulary from a continuously spoken utterance [18]. It could also learn associations between words and image objects. However, it needed a pre-trained phonetic speech recognizer. The system aimed to acquire image-grounded language knowledge and did not include action generation. Taniguchi et al. proposed a system based on a hierarchical Bayes model, which realized an integrated representation of language knowledge [19], [20]. It needed a pre-trained phoneme recognizer as the starting point of vocabulary learning. Their task in [19] was digit categorization based on multimodal observations. They used sound pronunciation of 0 to 9 and corresponding digit images from MNIST in their experiments.

From research on learning the pronunciations of new words, Taguchi et al. proposed a system to recognize and utter new place names by assuming a trained phoneme model [21]. It learned the place names from observed utterances and location coordinates within a fixed story, where the coordinate was estimated based on a laser range-finder. There were 12 place names in their evaluation story. Zuo et al. proposed a method to update pronunciation interactively [22]. They oriented rudimentary investigation, and the method needed a pre-trained phoneme recognizer and a pre-defined grammar that modeled the dialogue story. These studies adapted the pronunciation at a transcribed phoneme level and did not adapt the speech synthesizer.

These systems, except for [15], [16] and [22], learn based on co-occurrence modelings of speech sounds, images, and actions, whereas Gorin and Zuo’s systems learn based on the error feedback. For the language acquisition agents, actions are to interact with the environment. Pronouncing an utterance corresponds to moving a stone or pressing a controller button for game-playing agents [23], [24]. However, the co-occurrence modeling-based systems lack a mechanism to learn the effect of executing an action as the means to satisfy its goal through trial and error. Therefore, the agent’s actions are limited to reproducing those directly taught by teachers. As a reinforcement learning-based language acquisition system, Cruz et al. proposed a robot arm
system that learned to follow spoken instructions [25] based on SARSA [26]. Four types of arm movements (i.e., get, drop, go, and clean) were pre-programmed, and the action was to select one of the movement types. While the system could learn the instructions, it needed a trained speech recognition system to transcribe speech utterances. The action for the robot is not to pronounce an utterance but to move its arm.

Compared to the existing language acquisition studies, the novelty of our agent is that it is the first open vocabulary speaking agent that learns and understands the meaning of its utterance pronunciation. It learns the vocabulary as an action space through observation of continuous utterances, associates the learned word with visual meaning by sound-image grounding, and understands the meaning of the utterance pronunciation through trial-and-error of dialogues. It can learn spoken language without relying on trained sub-modules.

III. COMPONENT LEARNING MODULES
A. Unsupervised Word Learning

Unsupervised word learning discovers word segments from raw speech signals by detecting repeating units. Embedded Segmental K-Means (ES-KMeans) [27] is one of the methods that has shown effectiveness in previous works [28]. As shown in Algorithm 1, given a feature vector sequence of a continuous utterance $X = < x_1, x_2, \ldots, x_T >$, ES-KMeans algorithm aims to break it down into a sequence of meaningful segments or words $W = < w_1, w_2, \ldots, w_L >$ via an iterative optimization of segmentation $Q$ and clustering $Z$, where $T$ is the number of frames and $L$ is the number of words in the utterance.

An embedding function $f_e$ is used to map a variable-length segment to a fixed dimensional vector, which is implemented by down-sampling. The overall optimization objective is:

$$H(Q, Z) = \sum_{c=1}^{K} \sum_{w \in W_c} \text{len}(w) || f_e(w) - \mu_c^Z ||^2,$$

where $K$ is the number of clusters, $W_c$ is a set of segments by $Q$ assigned to cluster $c$ by $Z$, $\text{len}(w)$ is segment length of $w$, and $\mu_c^Z$ is $c$-th cluster mean of $Z$.

Given a fixed clustering $Z$, the objective (1) becomes:

$$H(Q) = \sum_{w \in W} d(w),$$

where $d(w) = \text{len}(w) || f_e(w) - \mu_c^Z ||^2$ and $\mu_c^Z$ is the cluster mean of $Z$ closest to $f_e(w)$. Let $\gamma[t] = H(Q_t^c)$ be the optimal segmentation score for a partial utterance up to $t$-th frame $X_t = < x_1, x_2, \ldots, x_t >$. Using dynamic programming, the optimal segmentation score for the whole utterance $\gamma[T] = H(Q_T^c)$ is efficiently obtained by recursively applying (3) from $t = 1$ to $T$.

$$\gamma[t] = \min_{j=1}^{t} \{ d(w_{t-j+1:t}) + \gamma[t-j] \},$$

where $w_{t-j+1:t}$ is a segment starting at frame $t-j+1$ and ending at $t$, and $\gamma[0] = 0$. By backtracking the recursion process, the optimal $L$ and segmentation $Q^* = Q_T^c$ are obtained.

Algorithm 1: ES-KMeans Algorithm.

1: Input: feature vectors $X = < x_1, x_2, \ldots, x_T >$
2: Output: segmentation $Q$, cluster assignments $Z$
3: Randomly initialize $Q$
4: Randomly initialize $Z$
5: while not converge do
6: Fix $Z$, optimize $\min_Q H(Q, Z)$
7: Fix $Q$, optimize $\min_Z H(Q, Z)$
8: end while
9: return $Q$ and $Z$

The clustering $Z$ for the obtained segments in the embedding space is performed by the standard $K$-means method. The process of the segmentation and the clustering is repeated until it converges.

B. Sound-Image Grounding

Harwath et al. [29] introduced an unsupervised sound-image correspondence learning approach. For sound-image grounding, they first use sound and image encoders to embed the audio descriptions and images to the same feature space with dimension $f$. Then they obtain the similarity score by calculating $L_2$-norm or cosine similarity between the audio and the image features. They expect the similarity scores between corresponding (positive) description-image pairs to be higher than the unrelated (negative) ones. They employ triplet-loss learning to train the audio and image encoders. Following them, we train the model by minimizing the following loss function:

$$L = t(f_I(x_I), f_S(x_S^+), f_S(x_S^-)),$$

$$t(v_I, v_S^+, v_S^-) = \frac{1}{2} ||v_I - v_S^+||^2 + \frac{1}{2} \max(\Delta - ||v_I - v_S^-||, 0)^2,$$

where $f_I$ and $f_S$ are image and sound encoders, $x_I$ is an anchor image, $x_S^+$ and $x_S^-$ are positive and negative audio descriptions, $\Delta$ is a positive margin, $v_I = f_I(x_I)$, $v_S^+ = f_S(x_S^+)$, and $v_S^- = f_S(x_S^-)$.

C. Pronunciation Learning

To implement the pronunciation capacity with the agent, we need a speech synthesizer. Record and playback is the most straightforward approach, where we make a sound dictionary and select and replay the item in the dictionary. A limitation is that modifying the pronunciation is not possible other than changing the selection of the item.

Another approach is to use a neural speech synthesizer, which provides flexible modification. While there are various neural vocoders, training stability and synthesis speed are important to use as a speech organ of the automated agent. Based on preliminary experiments comparing Parallel WaveGAN [30], WaveGrad [31], and DiffWave [32] neural vocoders, we adopt WaveGrad [31] for the proposed system as a real-time neural vocoder.
WaveGrad is one of the diffusion probabilistic vocoders based on denoising score matching [33] and diffusion probabilistic model [34]. It models the diffusion process from a speech sound to random noise and generates speech sounds as its inverse process [31], [32], [35]. Compared with conventional non-autoregressive neural vocoders, the diffusion probabilistic vocoders can be trained with a simple loss function in the time domain. It does not use the generative adversarial network (GAN) [36] training as in [30], [37]–[40] which requires careful architecture design and nontrivial parameter tuning to converge. It trains a neural network \( \epsilon_{\theta} \) that predicts Gaussian white noise \( \epsilon \) from the mixture of speech waveform \( x_0 \) and noise \( \epsilon \).

Fig. 1 shows an implementation example of \( \epsilon_{\theta} \), which we use in our experiment. It consists of up and down sampling blocks as used in GAN-TTS [41] and a Feature-wise Linear Modulation (FiLM) [42] module. To control the sequential diffusion process, it uses a gradually increasing noise schedule \( \beta_1, \beta_2, \cdots, \beta_N \), where \( N \) is the number of diffusion steps [34]. The network \( \epsilon_{\theta} \) has inputs \( h \) to control the speech waveform content and \( c \) to indicate the noise level at each step. The time domain loss function for the training is defined as shown in (6).

\[
\mathbb{E}_{\epsilon,c} \left[ \left\| \epsilon - \epsilon_{\theta} \left( \sqrt{\alpha_n} x_0 + \sqrt{1 - \alpha_n} \epsilon, h, c \right) \right\|_{1} \right],
\]

(6)

where \( \alpha_n \) is a random sample drawn uniformly between \( \sqrt{\alpha_{n-1}} \) and \( \sqrt{\alpha_n} \), \( c = \sqrt{\alpha_n} \), \( \alpha_n = \prod_{s=1}^{n} \alpha_n \), and \( \alpha_n = 1 - \beta_n \).

In the inference, input Gaussian white noise \( x_N \sim \mathcal{N}(0, I) \) is iteratively converted into a speech waveform by the denoising process based on Langevin dynamics [43] with \( n = N, N - 1, \ldots, 1 \) as follows:

\[
x_{n-1} = \frac{1}{\sqrt{\alpha_n}} \left( x_n - \frac{1 - \alpha_n}{\sqrt{1 - \alpha_n}} \epsilon_{\theta} (x_n, h, c) \right) + \sigma_n z, \quad n > 1,
\]

(7)

where \( \sigma_n = \sqrt{\beta_n (1 - \alpha_{n-1})} / (1 - \alpha_n) \), \( z \sim \mathcal{N}(0, I) \) for \( n > 1 \), and \( z = 0 \) for \( n = 1 \).

Typically, \( h \) is an acoustic feature sequence [31] or phoneme sequence [44] that represents a sentence for text-to-speech. It can also be a one-hot vector as in DiffWave [32] to specify a word for unconditional training and synthesis. We adopt the latter strategy for our speaking agent. Our agent internally generates a paired data set \{\( <h, x_0> \)\} for self-supervised learning.

### D. Reinforcement Learning

With reinforcement learning, an agent learns to maximize cumulative reward by optimizing its action taking. An agent’s policy \( \pi \) is a distribution over actions given a state. An action-value function \( Q(s, o) \) is the expected cumulative reward starting from state \( s \), taking action \( a \), and following policy \( \pi \). Q-learning iteratively estimates \( Q(s, a) \) for the optimal policy. Given estimated \( Q(s, a) \), we can derive an optimal action at that state.

Deep Q-learning (or Deep Q-Network; DQN) is an extension of Q-learning that Mnih et al. [24] introduced. It has proven to be effective in many challenging tasks such as computer resource management [45], robotics [46] and chemistry [47]. It estimates the action-value function using two deep neural networks with weight parameters \( \theta \) and \( \theta' \). One is the policy network \( Q(s, a; \theta) \) used to decide an action, and the other is the target network \( \hat{Q}(s', a'; \theta') \) that estimates the target action-value during training. When we perform the stochastic gradient descent, the samples are assumed to be independent. However, successive transitions are correlated, resulting in unstable training. To address this problem, DQN accumulates transitions in a replay buffer \( D \) and performs the gradient descent on the samples from \( D \) as shown in Algorithm 2.

### IV. TASK DESIGN

As environments for spoken language acquisition, we design two tasks. Both tasks consist of the observation phase and dialogue phase.

#### A. 3D Homing Task

In this task, the agent is placed in a 3D space, as shown in Fig. 2, and it is on a cart that recognizes voice commands. The agent is motivated to be as closer to the origin as possible. The agent has to learn what voice commands are available and when to pronounce which command to move efficiently. To model the
Algorithm 2: Deep Q-Learning With a Replay Buffer [24].

1: Initialize replay buffer $D$ to a certain size
2: Initialize action-value function $Q$ with random weights $\theta$
3: Initialize action-value function $\hat{Q}$ with random weights $\theta^-$
4: Initialize the number of collected transitions $n = 0$
5: Set the period of gradient descent $P_g$
6: for episode $= 1, 2, \ldots, S$ do
7: Initialize a state $s_0$
8: for $t = 1, 2, \ldots, T$ do
9: Select
$$a_t = \begin{cases} \text{random action,} & \text{with probability } \varepsilon \\ \arg \max_a Q(s_t; a; \theta), & \text{otherwise} \end{cases}$$
10: Execute an action $a_t$ and observe reward $r_t$ and the next state $s_{t+1}$
11: Store transition $(s_t, a_t, r_t, s_{t+1})$ in $D$
12: Increment the number of collected transitions $n$ by 1
13: if $n \equiv 0 \mod P_g$ then
14: Sample minibatch $(s_\tau, a_\tau, r_\tau, s_{\tau+1})$ from $D$
15: Set
$$y_\tau = \begin{cases} r_\tau, & \text{if episode is done at } \tau + 1 \\ r_\tau + \gamma \max_{a'_{\tau+1}} \hat{Q}(s_{\tau+1}; a;' \theta^-), & \text{otherwise} \end{cases}$$
16: Perform gradient descent with
$$(y_\tau - Q(s_\tau; a_\tau; \theta))^2$$
17: end if
18: Every $C$ steps, $\hat{Q}$ loads the weights of $Q$
19: end for
20: end for

In the observation phase, the agent first “listens” to the sounds coming from the environment. The sound is a single stream that contains audio descriptions of foods with random intervals of 1 to 3 seconds. For each food, there are four types of descriptions: “Food,” “A (An) food,” “A (An) color food,” and “It’s a (an) food”. Then, the agent observes food images with corresponding audio descriptions.

In the dialogue phase, the environment shows two foods to the agent with a question “Which do you want?”. The agent wants the one whose color is closer to its favorite color. If the agent pronounces one of the shown food names and the environment correctly recognizes it, the environment gives the food to the agent as feedback. If the agent answers something else, the environment gives nothing to the agent.

V. PROPOSED AGENT

Fig. 4 illustrates the common structure of our proposed agents. They consist of sensory front-ends for internal and external state observations, an intention and message generator, a speech synthesizer, an internal state, and a reward evaluator. The agents first observe the internal and external states through the sensory front-ends. Then they generate a message based on their satisfaction level between two consecutive steps:

$$SL(t) = - (x_t^2 + y_t^2 + z_t^2),$$

$$r(t) = SL(t) - SL(t - 1),$$

where $x_t$, $y_t$, and $z_t$ are the coordinates of the agent at step $t$.

In the observation phase, the agent listens to a long speech stream that contains the command and unrelated noise words. In the dialogue phase, the agent observes the current position and pronounces an utterance. If the cart recognizes the utterance as a directional command, it pushes the agent along in the corresponding direction by one unit. It is the agent’s side responsibility that the cart correctly recognizes the agent’s pronunciation.

B. Food Task

The goal of the agent is to get food that matches the agent’s preference, as shown in Fig. 3. The agent has a favorite color as its internal state. The agent becomes happy only when it gets the food that it better prefers between the two choices. When it is happy, the reward is 1.0, and otherwise, it is 0.0. In every dialogue episode, the agent has a random favorite color.

In the observation phase, the agent first “listens” to the sounds coming from the environment. The sound is a single stream that contains audio descriptions of foods with random intervals of 1 to 3 seconds. For each food, there are four types of descriptions: “Food,” “A (An) food,” “A (An) color food,” and “It’s a (an) food”. Then, the agent observes food images with corresponding audio descriptions.

In the dialogue phase, the environment shows two foods to the agent with a question “Which do you want?”. The agent wants the one whose color is closer to its favorite color. If the agent pronounces one of the shown food names and the environment correctly recognizes it, the environment gives the food to the agent as feedback. If the agent answers something else, the environment gives nothing to the agent.
intention. The sensory front-ends and the message generator form a policy function of the agent. The speech synthesizer module makes a waveform utterance from the internally represented message. The outside environment or the dialogue partner recognizes the agent’s utterance. Based on the recognized result, the environment provides feedback to the agent, such as moving its position and passing food. The reward evaluator calculates the reward, and the agent tries to maximize it. The internal state is simply a variable; the reward evaluator is a fixed function.

The sensory front-ends, intention and message generator, and the speech synthesizer are learnable modules. They are learned from scratch during the language acquisition process without relying on manually prepared labeled data.

In observation phase, the agent silently observes the outside environment. Depending on the task settings, the agent accumulates raw speech samples or paired speech and image samples. The agent can use the data for unsupervised or self-supervised learning. In dialogue phase, the agent tries to work on the outside environment by speaking. The agent can use the experience for reinforcement learning. Additionally, the agent may use the recorded experiences for unsupervised and self-supervised learning.

A. Spolacq 1: Vocabulary Learning

If we simply treat the waveform utterance as a high-dimensional continuous action space, there is effectively no chance that the agent generates a meaningful utterance. The probability of a randomly generated signal becoming a meaningful utterance is almost zero at the beginning of the learning. Without generating an utterance that results in a positive reward, reinforcement learning does not proceed.

The first version of our spoken language acquisition agent (spolacq 1) uses the sound dictionary as the utterance action space, which we have firstly proposed in [11]. Fig. 5 shows the structure. The agent learns the dictionary using ES-KMeans in the observation learning. When making the dictionary, recall is vital so that the dictionary contains the correct segmentation of all needed words. The accuracy of ES-KMeans is not high, and the produced sound dictionary contains many broken segments. However, as far as the dictionary contains correct segments, the agent can learn the selection of word segments by using reinforcement learning in the dialogue learning. Even though the dictionary is very inaccurate, the effect of using discrete action space is enormous to improve the efficiency of reinforcement learning.

We use a multi-layer fully connected neural network (FCN) as the policy network. The agent observes its 3D position through an observation encoder in the 3D task and two images through two image encoders.

B. Spolacq 2: Focusing Mechanism

In spolacq 1, the action space is the size of the sound dictionary, which becomes larger than the actual vocabulary due to the low accuracy of unsupervised word learning. At the beginning of the dialogue learning, the agent uniformly tries the item in the dictionary since it has no prior information. Therefore, the search becomes inefficient with the increase of the dictionary size. When humans learn a language, visual information plays an important role in improving learning efficiency by association. By introducing such a mechanism, we could improve the learning efficiency of the agent.

We propose spolacq 2 that introduces a focusing mechanism that guides the agent’s attention to those concepts in the agent’s eyesight, as shown in Fig. 6. With this design, we assume the agent has both hearing and sight at least in the observation phase. Note that we have refined the design and implementation of our focusing mechanism from [12] for simplicity and extensibility. Here we explain the refined version.

---

2Following the convention of Stable-Baselines3 reinforcement learning toolkit, we call both narrow-sense policy and value broad-sense policy.

3For the conciseness of the explanation, we do not describe the first version here. While the basic idea and the effect are the same, the first version has a more complicated structure. For the details of the first version, please refer to [12].
In the observation learning, the agent performs the triplet-loss-based sound-image grounding using paired sound-image samples. It makes the image encoder $f_I$ and the sound encoder $f_S$. The agent initializes the image observation front-end of the policy network by $f_I$. It applies $f_S$ to the items in the sound dictionary and makes an array of their embedded representations $k = [k_1, k_2, \ldots, k_n]$. Optionally, it also uses $f_S$ as sound observation front-end. For the dialogue learning, it uses image feature $f_I(x_1)$ as a query and the embedded dictionary entries $k_a$ as keys and calculates a weight vector $w^{x_1}$ as in the end-to-end memory networks [48] as shown in (11) and (12).

$$k_a = f_S(x_S^{(a)}),$$

$$w^{x_1} = \frac{1}{\max(\bar{w}^{x_1})} \bar{w}^{x_1},$$

where $x_S^{(a)}$ is an audio segment in the sound dictionary. The weight vector provides a prior distribution for the dictionary item selection. Using the prior distribution, the agent has a higher chance of selecting relevant actions.

Because the correct action may not always be related to the image input and the sound-image association obtained by the unsupervised learning may be inaccurate, the policy function needs a mechanism to overwrite the vision-based focusing. For this, we calculate a weighted average of the output from the fully connected feed-forward neural network $g_c$ used in spolacq 1 and the image query-based weight vector. We obtain their weights by another neural network $\hat{g}_c$ with a softmax output layer. If the agent has multiple image inputs, as in our food task experiment, we first calculate separated image query-based weight vectors for each of them and then obtain their weighted average. The following equations summarize the calculation of the policy network.

$$s = (s_m, x_S, x_{11}, x_{12}),$$

$$\hat{s} = g_p(s_m) \oplus f_S(x_S) + f_I(x_{11}) \oplus f_I(x_{12}),$$

$$[\alpha_1, \alpha_2, \alpha_3] = \text{softmax}(\hat{g}_c(\hat{s})),$$

$$Q(s, a; \theta) = (\alpha_1 g_c(s) + \alpha_2 w^{x_{11}} + \alpha_3 w^{x_{12}})[a],$$

where $\oplus$ is a concatenation operator, $x_S$ is a spoken question from the dialogue partner, $g_p$, $\hat{g}_c$, and $g_c$ are randomly initialized FCNs, and $[\cdot][a]$ indicates the a-th element of a vector. This way, spolacq 2 can transfer more knowledge from observation learning to dialogue learning than spolacq 1.

C. Spolacq 3: Neural Vocoder-Based Pronunciation

One limitation of spolacq 1 and 2 is that the agent pronounces only by selecting and replaying an element in the sound dictionary; thus, it lacks the mechanism to modify and improve the pronunciations. To enable the ability to adapt the pronunciation, we propose spolacq 3 that replaces the sound dictionary with a trainable speech synthesizer [14]. The synthesizer should be able to produce variations of waveforms for the same utterance so that the agent can explore different pronunciations. Additionally, it should have few tuning factors and be computationally efficient to fit with the self-supervised learning framework. For these reasons, we choose the WaveGrad method. As shown in Fig. 7, the new agent has the same structure as the original one except for the WaveGrad-based speech organ. The action vector obtained from the policy network works as the condition $h$ for the WaveGrad.

We integrate the training of WaveGrad into the self-supervised learning framework as shown in Algorithm 3. In the algorithm, the agent first uses the sound dictionary approach with $N_A$ dialogue episodes. During the dialogue-based learning process, it records its experience. Using the record, the agent makes a data set $D_0$ of pairs of an index of the sound dictionary element and the corresponding waveform segment obtained from a successful dialogue episode. Because the original sound dictionary contains many junk segments, the agent compresses it by removing those elements that have never resulted in successful episodes. Accordingly, the agent prunes the policy function’s output units corresponding to the removed entries. The index IDs in $D_0$ are also updated for consistency. Then the agent uses the modified $D_0$ to train the WaveGrad synthesizer that takes the element index represented by a one-hot vector as the condition $h$, and generates the corresponding waveform as the output.

After obtaining the trained WaveGrad, the agent plugs in it on top of the action-value function’s output and starts using it. In the succeeding dialogues, the agent makes a new data set $D_1$ of pairs of an action vector and a generated sound waveform for every
Algorithm 3: WaveGrad Speech Organ-Based Spoken Language Acquisition.

1: Train the sound segment dictionary-based language acquisition agent with $N_A$ dialogue episodes.
2: Make a set of paired data $D_0$ of an index of the sound dictionary element and the corresponding waveform segment that is used in a successful dialogue episode.
3: Train a WaveGrad speech synthesizer using $D_0$. Replace the sound dictionary of the agent with the trained WaveGrad synthesizer.
4: for $l = 1$ to $L_T$ do
5: Repeat the dialogue episodes for $N_B$ times and train the value function of the agent.
6: Make a set of paired data $D_l$ of the action vector and the generated sound waveform recorded from the successful episodes.
7: Update WaveGrad speech synthesizer using $D_l$.
8: end for

$N_B$ episodes choosing successful experiences and updating the WaveGrad module using it.

VI. DATA SET

A. Data Set for 3D Homing Task

As the long sound stream in the observation phase, we use the Google Speech Commands Dataset [49]: an English voice command dataset with 65,000 one-second-long utterances of 35 short words by thousands of different people. We use a subset of the data set consisting of six words indicating direction and one unrelated noise word: up, down, left, right, forward, backward, and marvin. For each type of word, we pick 200 samples. In total, there are 1,400 samples, which we then concatenate into a single wave file.

B. Data Set for Food Task

Our food data set [12] has 20 types of common fruits and vegetables: apple, banana, carrot, cherry, cucumber, egg, eggplant, green pepper, hyacinth bean, kiwi fruit, lemon, onion, orange, potato, sliced bread, small cabbage, strawberry, sweet potato, tomato, and white radish. Each food type has 120 images, 90 of which are used for training and the rest are used as the test set. We used the Google Text-to-Speech library\(^4\) to generate four types of audio descriptions for each food using the template described in Section IV-B changing the food names. In each experiment, we use a subset of the food types.

VII. EXPERIMENTAL SETUP

A. Configuration of 3D Homing Task

The agent’s observation is a 3D coordinate represented by a triplet of integer values. We omit the internal state encoder from spolacq 1 in Fig. 5 since the agent’s internal state is the destination location, and it is fixed to the origin (0,0,0). The dialogue learning is based on DQN with a replay buffer of size 10,000. The neural network has a simple feed-forward structure with three ReLU hidden layers. The hidden layer size is 32. As the hyper-parameters in Section IV-A, we set the position range $k$ at 22 and the maximum number of steps taken for each round at 5,000. If the agent does not come back to the origin after 5,000 steps, the round fails, and the agent position is reset. To simulate the speech recognition cart that recognizes the speech commands, we use the Google Speech-to-Text API,\(^5\) which is a trained general-purpose ASR system.

To see the influence of the accuracy of the unsupervised word learning to make the sound dictionary, we compare random segmentation and ES-KMeans. For the random segmentation, we cut the sound stream randomly with an average duration of approximately one word (e.g., 500–1,200 ms). As it is a random cut, broken word segments are expected to be more frequent.

B. Configuration of Food Task

For ES-KMeans described in Section III-A, we set $K = 2$ in (1) and make the sound dictionary by collecting all the obtained segments. For the triplet-loss-based sound-image grounding in the observation learning, we use $f = 50$ as the feature size. We used a sound sample paired to an anchor image as a positive audio description, and the one randomly selected from minibatch as the negative audio description. The image and sound front-end networks $f_I$ and $f_S$ have the ResNet-50 [50] structure. For $f_S$, we modify the first 2D convolution layer, where it has a

\(^4\)Available: https://pypi.org/project/google-cloud-texttospeech/

\(^5\)Available: https://cloud.google.com/speech-to-text/docs/reference/rest/
configuration of input channel of 1, output channel of 64, kernel size of 7, stride of 2, padding of 3, and no bias. While we use the ResNet-50 structure, we randomly initialize the parameters.

We use DQN for dialogue learning. As shown in Figs. 6 and 7, we connect two image encoders to the policy network as the observation encoders and initialize them with \( f_I \). We additionally add a sound encoder to the policy network and initialize it with \( f_S \) when we input a sound question to the agent.

For the WaveGrad training with spolacq 3, we set \( L_T = 10 \), \( N_A = 120,000 \), and \( N_B = 24,000 \) in Algorithm 3. For the noise schedule used in (7) in Section III-C, we used the arithmetic progression with \( N = 50 \), \( \beta_1 = 0.0001 \) and \( \beta_N = 0.05 \). The sampling rate of the generated waveforms was 8,000 Hz.

To implement the speech recognition in the dialogue environment, we use Fairseq\(^6\) with wav2vec 2.0 \([51]\) based on LibriSpeech 960 h that is publicly available at Hugging face\(^7\).

With our refined implementation, we use OpenAI Gym \([52]\) to implement reinforcement learning (RL) environment and Stable-Baselines3 \([53]\) for RL algorithms.

VIII. RESULTS

A. Results of 3D Homing Task

We first investigate spolacq 1 using the 3D homing task. Table I compares the segmentation accuracy by the random cut and ES-KMeans by means of pseudo recognition rate. This is calculated by the number of recognized meaningful word segments over the total number of actual words in the sound stream (i.e., 1,200). We blindly obey the results from ASR but ignore the true quality of the identified words based on the fact that the only information source for the agent is the environment (ASR in this case). We observed that ES-KMeans-based segmentation is 18% more accurate than the random-cut method.

Fig. 8 shows the results of the dialogue phase. The vertical axis represents the steps taken by the agent to return to the origin in each episode. The results are obtained by repeating 100 experiments with random initialization. From the figure, we can safely conclude that the acquisition of spoken language is successful as the number of steps taken eventually converges. The random-cut-based sound dictionary does work as well because as long as there are meaningful candidates for valid words, the agent is able to learn to speak correctly by reinforcement learning. However, we do observe a difference in their speed of learning. The agent using the ES-KMeans-based sound dictionary does excel the random-cut-based agent with a 35.45% reduction in the average number of steps taken for the first episode. The ES-KMeans method also shows greater stability in performance, considering the standard deviation shown in the results.

As a supplemental evaluation, we performed a subjective test asking six participants. The participants tracked the agent’s behavior in the last 50 steps of the first and fifth episodes and evaluated the validity of the agent’s utterances. Each participant evaluated two runs of the learning experiments with random initializations. Table II shows the mean opinion score (MOS). As can be seen, the MOS score improved in the fifth episode compared to the first episode. The participants observed that the agent often uttered irrelevant directions for its destination and sometimes pronounced broken words in the first episode. In the fifth episode, the agent mostly spoke the right direction, while sometimes it uttered non-appropriate directions.

B. Results of Food Task

Fig. 9 compares spolacq 1 and 2 with the food task. The figure plots the test set reward for the number of episodes in the dialogue phase. We used 16 food types in this experiment:
The single-stream sound $L$ in Algorithm 2 $\in \mathcal{M}_{\text{closest sound segments}}$ for each of the $L_{1}$ classes by the $D_M$ elements $\lambda = 0.97$. The number of different indexes in $D_0$, i.e., the number of active sound segments, was 213. As seen in Fig. 10, the reward curve by the sound dictionary-based spolacq 2 agent was almost flat after we added the sinusoid noise. This was because it had no means to adjust the pronunciation other than changing the segment selection. On the other hand, the curve by WaveGrad-based spolacq 3 gradually improved because it could adapt its pronunciation based on its experience. When we listened to the WaveGrad generated utterances after the sinusoid noise adaptation, they were natural.

Lastly, we investigated whether the agent learns and understand logical negation in an utterance question [13]. We added a sound-encoder to the policy network and asked the agent one of two questions randomly: “Which do you want?” or “Which do not you want?” When the question is the latter, the agent obtains the desired food when it answers the opposite one. For this experiment, we used ten food types: apple, banana, carrot, cucumber, eggplant, onion, orange, potato, strawberry, and tomato. The combination of choosing two types of foods from 10 is $C(10, 2) = 45$. To evaluate the agent’s response for new food combinations that do not appear in training, we used 35 combinations for the training and the remaining 10 for the testing. Additionally, we kept 10% of the training samples of each food as the development set. The single-stream sound in the observation phase contained 7,200 audio descriptions with 20 dB of Gaussian noise. We used the Google Text-to-Speech to generate the sound utterances of the two questions. We used the first version of our focusing mechanism [12] with $M = 24$, $L = 500$, and $\lambda = 0.9$.

Fig. 11 shows the development set reward obtained during the dialogue phase. We also evaluated the agent with a fixed question utterance of one of the question types for the comparative purposes. We can observe from the figure that the agent learns to pronounce the appropriate food names to obtain the desired ones regardless of the question types.

Table III shows the test set rewards after the training. The reward for the fixed negative question “Which do not you want?” was higher than that for the positive question “Which do you want?”. The difference arose because the food color spolacq 2, the fully connected feed-forward neural network $g_c$ introduced plays the same role.

---

8 Since the speech recognizer used to implement the environment had difficulty recognizing some of the food names even if their pronunciations were clear, we used a subset of the food types. The speech model was wav2vec2-large-robust-ft-libri-960 h.

9 It makes the sound dictionary in two steps. The first step clusters the image features extracted by the image encoder $f_I$ to $M$ classes by the K-means method. The second step forms the sound dictionary having $ML$ elements by selecting $L$ closest sound segments for each of the $M$ cluster centroids. To overwrite inaccurate initial sound-image grounding, it has an action filter mechanism with a parameter $\lambda \in [0, 1]$. The parameter controls the decay speed of the weights for unsuccessful action choices. In our refined implementation of

---

apple, banana, carrot, cherry, cucumber, green pepper, hyacinth bean, lemon, onion, orange, potato, sliced bread, small cabbage, strawberry, sweet potato, and tomato. The single-stream sound in the observation phase contained 720 audio descriptions of foods. We added 30 dB white noise to the agent’s utterances to simulate an environmental noise. For both spolacq 1 and 2, we used two image encoders initialized with $f_I$ as the observation encoders. The shapes of FCNs $g_p$, $g_c$, and $g_e$ are (3, 50), (150, 75, 1488), and (150, 75, 3), respectively. With the DQN-based learning, we accumulated the experiences in the replay buffer with a size of 1,000 for 50 episodes before we started training the models and kept the other hyper-parameters as the default of Stable-Baselines3 [53]. Compared to the configuration in [12], we decreased the period of gradient descent $P_g$ in Algorithm 2 from every 16 steps to 4 steps, which improved the learning speed of both spolacq 1 and 2. We can observe from the figure that the proposed focusing mechanism boosts both learning speed and the final reward.

To see how the agents react to the change of the environmental noise, Fig. 10 compares spolacq 2 and 3. We added extra 0 dB, 300 Hz sinusoid noise to the agent’s utterances after 120,000 episodes in addition to the 30 dB white noise. Since WaveGrad needs more computation than the segment selection approach, we ran this experiment using eight types of foods: cherry, green pepper, lemon, orange, potato, strawberry, sweet potato, and tomato. We ran the experiment using the first version of our focusing mechanism [12] that had additional parameters of dictionary construction $M = 120$, $L = 100$ and action filter $\lambda = 0.97$. The number of different indexes in $D_0$, i.e., the number of active sound segments, was 213. As seen in Fig. 10,
distribution was not uniform\(^{10}\), and different food names had different recognition accuracies. The reward for the two random questions was about an average of the two fixed questions, which indicates that an agent understood the logical negation and could appropriately answer the random questions for the unseen food combinations.

To confirm the learned agent’s generality, we additionally ran the refined spolacq 2 with the utterance questions and performed a subjective test asking six participants. In this experiment, we used the 16 food types. We used authentic human voices recorded from 31 speakers for the two types of question utterances used in the dialogue training phase instead of the synthesized utterances. Each speaker recorded five utterances for each of the question types. After the learning, the six participants randomly asked the two questions to the agent and evaluated the MOS score for the agent’s answers. There was no overlap between the 31 speakers who provided the training data and the six participants for the subjective evaluation. Table IV compares the reward and the MOS score after 50,000 and 1,000,000 training episodes. We ran two test trials changing a random seed. From the table, we can confirm that the agent can also work in this condition for the unseen food combinations.

### IX. Summary and Conclusion

We have proposed spoken language acquisition agents that learn vocabulary from scratch and understand the meaning of their pronounced utterances. The agent supports 1) word discovery from the raw speech signal, 2) grounding of meanings with words, 3) language encoded message generation as an action for the current internal and external states driven by intrinsic motivation, and 4) pronunciation generation to express the message. Our agent can efficiently learn spoken language from limited dialogue interactions because of its architecture. The key components are: 1) The discrete action space using the sound dictionary, 2) The unsupervised learning-based pre-training of the sound and image encoders, 3) The focusing mechanism utilizing the result of the unsupervised sound-image grounding. With this design, the policy network only needs to select elements in the sound dictionary. The pre-trained sound and image encoders work from the beginning of reinforcement learning. The focusing mechanism can compute keys and the weight vectors without involving learning components (i.e., the keys are obtained using the pre-trained sound encoder, and the weight vectors are obtained by similarity evaluation). All of these contribute to reducing the task complexity for reinforcement learning. The agent can also learn a neural vocoder for flexible pronunciation adaptation and the concept of logical negation. Future work includes evaluating our agents using more extensive data sets with human voices. By integrating more unsupervised and meta-learning techniques, the agent will significantly extend the learning ability.

### REFERENCES

1. B. F. Skinner, *Verbal Behavior*. New York, NY, USA: Appleton-Century-Crofts, 1957.
2. N. Chomsky, “A review of B. F. Skinner’s verbal behavior,” *Language*, vol. 35, no. 1, pp. 26–58, 1959.
3. C. B. Sturdy and E. Nicoladis, “How much of language acquisition does operant conditioning explain?,” *Front. Psychol.*, vol. 8, Oct. 2017, Art. no. 1918.
4. P. K. Kuhl, “Early language acquisition: Cracking the speech code,” *Nature Rev. Neurosci.*, vol. 5, no. 11, pp. 831–843, Nov. 2004.
5. E. Dupoux, “Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner,” *Cognition*, vol. 173, pp. 43–59, Apr. 2018.
6. B. A. Goldfield and J. S. Reznick, “Early lexical acquisition: Rate, content, and the vocabulary spurt,” *J. Child Lang.*, vol. 17, no. 1, pp. 171–183, 1990.
7. S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ, USA: Prentice Hall, 1995.
8. G. Baldassarre, T. Stafford, M. Mirolli, P. Redgrave, R. M. Ryan, and A. Barto, “Intrinsic motivations and open-ended development in animals, humans, and robots: An overview,” *Front. Psychol.*, vol. 5, no. 766, Sep. 2014, Art. no. 985.
9. J. W. Pilley and A. K. Reid, “Border collii comprehends object names as verbal referents,” *Behav. Process.*, vol. 86, no. 2, pp. 184–195, Feb. 2011.
10. C. ten Carel and P. J. Fullagar, “Vocal imitations and production learning by australian musk ducks (biziura lobata),” *Philos. Trans. Roy. Soc. B Biol. Sci.*, vol. 376, Sep. 2021, Art. no. 20200243.
11. S. Gao, W. Hou, T. Tanaka, and T. Shinozaki, “Spoken language acquisition based on reinforcement learning and word unit segmentation,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 6149–6153.
12. M. Zhang, T. Tanaka, W. Hou, S. Gao, and T. Shinozaki, “Sound-image grounding based focusing mechanism for efficient automatic spoken language acquisition,” in *Proc. Interspeech*, 2020, pp. 4183–4187.
13. K. Toyoda, Y. Kimura, M. Zhang, K. Hino, K. Mori, and T. Shinozaki, “Self-supervised spoken question understanding and speaking with automatic vocabulary learning,” in *Proc. 24th Conf. Oriental COCOSDA Int. Committee Co-Ordination Standardisation Speech Databases Assessment Techn.*, 2021, pp. 37–42.
14. T. Tanaka, R. Komatsu, T. Okamoto, and T. Shinozaki, “Pronunciation adaptive self speaking agent using wavegrad,” in *Proc. 2nd Workshop Self-Supervised Learn. Audio Speech Process.*, 2022.
15. A. L. Gorin, S. E. Levinson, and A. Sankar, “An experiment in spoken language acquisition,” *IEEE Trans. Speech Audio Process.*, vol. 2, no. 1, pp. 224–240, Jan. 1994.
16. A. L. Gorin, S. E. Levinson, A. N. Gertner, and E. Goldman, “Adaptive acquisition of language,” *Comput. Speech Lang.*, vol. 5, no. 2, pp. 101–132, Apr. 1991.
17. N. Iwashashi, “Language acquisition through a human-robot interface,” in *Proc. Int. Conf. Spoken Lang. Process.*, 2000, pp. 442–447.
18. D. K. Roy and A. P. Pentland, “Learning words from sights and sounds: A compositional model,” *Cogn. Sci.*, vol. 26, no. 1, pp. 113–146, Jan. 2002.
19. T. Taniguchi et al., “NEURO-SEIKET: Development of integrative cognitive system through the composition of deep probabilistic generative models,” *New Gener. Comput.*, vol. 38, pp. 23–48, Jan. 2020.
20. T. Taniguchi et al., “A whole brain probabilistic generative model: Toward realizing cognitive architectures for developmental robots,” *Neural Netw.*, vol. 150, pp. 293–312, Jun. 2022.

\(^{10}\)The mean of test images’ RGB \(x \in [0, 1]^{3}\) was (0.89, 0.76, 0.56), which is not close to (0.5, 0.5, 0.5).
[21] R. Taguchi et al., “Learning place-names from spoken utterances and localization results by mobile robot,” in Proc. Interspeech, 2011, pp. 1325–1328.

[22] X. Zao, T. Sumii, N. Iwashashi, M. Nakano, K. Funakoshi, and N. Oka, “Correcting phoneme recognition errors in learning word pronunciation through speech interaction,” Speech Commun., vol. 55, no. 1, pp. 190–203, Jan. 2013.

[23] D. Silver et al., “Mastering the game of go with deep neural networks and tree search,” Nature, vol. 529, pp. 484–489, Jan. 2016.

[24] V. Mnih et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, pp. 529–533, Feb. 2015.

[25] F. Cruz, J. Twiefel, S. Magg, C. Weber, and S. Wermter, “Interactive reinforcement learning through speech guidance in a domestic scenario,” in Proc. Int. Joint Conf. Neural Netw., 2015, pp. 1–8.

[26] G. A. Rummery and M. Niranjan, “On-line Q-learning using connectionist systems,” Cambridge University Engineering Department, Cambridge, U.K., Tech. Rep. CUED/F-INFENG/TR 166, Nov. 1994.

[27] H. Kamper, K. Livescu, and S. Goldwater, “An embedded segmental K-means model for unsupervised segmentation and clustering of speech,” in Proc. IEEE Autom. Speech Recognit. Understanding Workshop, 2017, pp. 719–726.

[28] Y.-A. Chung, W.-H. Weng, S. Tong, and J. Glass, “Unsupervised cross-modal alignment of speech and text embedding spaces,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2018, pp. 7765–7775.

[29] D. Harwath, A. Torralba, and J. Glass, “Unsupervised learning of spoken language with visual context,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2016, pp. 1858–1866.

[30] R. Yamamoto, E. Song, and J.-M. Kim, “Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2020, pp. 6199–6203.

[31] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, “WaveGrad: Estimating gradients for waveform generation,” in Proc. Int. Conf. Learn. Representations, 2021, pp. 1–15.

[32] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, “DiffWave: A versatile diffusion model for audio synthesis,” in Proc. Int. Conf. Learn. Representations, 2021, pp. 1–17.

[33] P. Vincent, “A connection between score matching and denoising autoencoders,” Neural Comput., vol. 23, no. 7, pp. 1661–1674, Jul. 2011.

[34] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 6840–6851.

[35] T. Okamoto, T. Toda, Y. Shiga, and H. Kawai, “Noise level limited sub-modeling for diffusion probabilistic vocoders,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2021, pp. 6029–6033.

[36] I. Goodfellow et al., “Generative adversarial nets,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2014, pp. 2672–2680.

[37] K. Kumar et al., “MelGAN: Generative adversarial networks for conditional waveform synthesis,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2019, pp. 14910–14921.

[38] J. Kong, J. Kim, and J. Bae, “HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 17022–17033.

[39] G. Yang, S. Yang, K. Liu, P. Fang, W. Chen, and L. Xie, “Multi-band MelGAN: Faster waveform generation for high-quality text-to-speech,” in Proc. IEEE Spoken Lang. Technol. Workshop, 2021, pp. 492–498.

[40] T. Okamoto, T. Toda, and H. Kawai, “Multi-stream HiFi-GAN with data-driven waveform decomposition,” in Proc. IEEE Autom. Speech Recognit. Understanding Workshop, 2021, pp. 610–617.

[41] M. Bünikowski et al., “High fidelity speech synthesis with adversarial networks,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 1–17.

[42] E. Perez, P. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, “FiLM: Visual reasoning with a general conditioning layer,” in Proc. AAAI Conf. Artif. Intell., 2018, pp. 3942–3951.

[43] Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2019, pp. 11918–11930.

[44] N. Chen et al., “WaveGrad 2: Iterative refinement for text-to-speech synthesis,” in Proc. Interspeech, 2021, pp. 3765–3769.

[45] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, “Resource management with deep reinforcement learning,” in Proc. 15th ACM Workshop Hot Top. Netw., 2016, pp. 50–56.

[46] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep viscous motor policies,” J. Math. Learn. Res., vol. 17, no. 39, pp. 1–40, Apr. 2016.

[47] Z. Zhou, X. Li, and R. N. Zare, “Optimizing chemical reactions with deep reinforcement learning,” ACS Central Sci., vol. 3, no. 12, pp. 1337–1344, Dec. 2017.

[48] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, “End-to-end memory networks,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2015, pp. 2440–2448.

[49] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” 2018, arXiv:1804.03209.

[50] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[51] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 12449–12460.

[52] G. Brockman et al., “OpenAI Gym,” 2016, arXiv:1606.01540.

[53] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, “Stable-Baselines3: Reliable reinforcement learning implementations,” J. Mach. Learn. Res., vol. 22, no. 268, pp. 1–8, Nov. 2021.

Ryota Komatsu received the B.Eng. degree in 2021 from the Tokyo Institute of Technology, Tokyo, Japan, where he is currently a Student. His research focuses on spoken language acquisition.

Shengzhou Gao received the B.Eng. degree from the National University of Singapore, Singapore, in 2019, and the M.Eng. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 2021. He is currently a Software Engineer with NetEase China. His research interests mainly include spoken language processing, speech recognition, and reinforcement learning.

Wenxin Hou received the B.Eng. degree from Nanjing University, Nanjing, China, in 2019, and the M.Eng. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 2021. He is currently a Software Engineer with Microsoft. His research interests mainly include spoken language processing, semi-supervised learning, transfer learning and its applications in natural language, and speech processing.

Mingxin Zhang received the B.Eng. degree in 2019 from the Tokyo Institute of Technology, Tokyo, Japan, where he is currently a Student. His research interests mainly include reinforcement learning and dialogue system.
Tomohiro Tanaka received the B.Eng. and M.Eng. degrees from the Tokyo Institute of Technology, Tokyo, Japan, in 2019 and 2021, respectively. He is currently a Software Engineer with Koei Tecmo Games. His research interests mainly include spoken language processing, reinforcement learning, and semi-supervised learning.

Keisuke Toyoda received the B.Eng. degree from the Tokyo Metropolitan University, Hachioji, Japan in 2020. He is currently a Student with the Tokyo Institute of Technology. His research focuses on speech processing.

Yusuke Kimura received the Bachelor of Engineering degree in March 2020 from the Tokyo Institute of Technology, Tokyo, Japan, where he is currently working toward the master’s degree.

Kent Hino received the Bachelor of Engineering degree in March 2020 from the Tokyo Institute of Technology, Tokyo, Japan, where he is currently working toward the master’s degree with the Department of Information and Communications Engineering, School of Engineering. His research focuses on optimization for automatic speech recognition.

Yu Iwamoto received the Bachelor of Engineering degree in 2021 from the Tokyo Institute of Technology, Tokyo, Japan, where he is currently a Student. His research focuses on unsupervised learning of spoken languages.

Kosuke Mori received the Bachelor of Engineering degree from the Tokyo University of Agriculture and Technology, Fuchu, Japan, in 2020. He is currently a Student with the Tokyo Institute of Technology, Tokyo, Japan. His research interests mainly include automatic speech recognition and its applications for spoken language assessment.

Takuma Okamoto (Member, IEEE) received the B.E., M.S., and Ph.D. degrees from Tohoku University, Sendai, Japan, in 2004, 2006, and 2009, respectively. In 2009, he was a Postdoctoral Research Fellow with Tohoku University. From 2012 to 2020, he was a Researcher with the National Institute of Information and Communications Technology, Tokyo, Japan, where he is currently a Senior Researcher. His main research interests include sound field synthesis based on acoustic signal processing and speech synthesis based on deep learning.

Takahiro Shinozaki (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in computer science from the Tokyo Institute of Technology, Tokyo, Japan, in 1999, 2001, and 2004, respectively. He is currently an Associate Professor with the Tokyo Institute of Technology, Tokyo, Japan. From 2004 to 2006, he was a Research Scholar with the Department of Electrical Engineering, University of Washington, Seattle, WA, USA. His research interests include semi-supervised and unsupervised learning of spoken languages, automatic spoken language acquisition, and their applications.