Spoken-text processing for spontaneous speech generation

Daiki Yoshioka

Contents

Abstract

Text-to-speech (TTS) is the task of generating speech waveforms from text. TTS has advanced dramatically with the advent of deep learning. For reading-style speech, it is now possible to synthesize speech that rivals human quality and naturalness. Recent research has moved beyond reading-style speech toward generating more human-like spontaneous speech, and a growing body of work is accumulating in this direction.

Nevertheless, most prior studies implicitly assume that the source text for spontaneous speech already contains spontaneous phenomena. In real-world scenarios (except when using transcripts) text-only data are typically written in a formal, written style; even when spoken-style text is available, spontaneous phenomena such as disfluencies are rarely present. Manually adding such elements is costly and impractical. Moreover, text-side processing and speech-side processing have generally been treated as separate pipelines and investigated in isolation.

To bridge this gap, we propose an integrated framework that unifies text processing and speech synthesis for spontaneous speech generation. Concretely, we cascade text style transfer (TST), which converts text into a different style while preserving meaning, with TTS to construct a pipeline that injects disfluencies into written input text and then synthesizes spontaneous speech from the resulting spoken-style text. Two challenges must be addressed to realize this approach.

First, TST performance in the labeled nonparallel setting remains inadequate. Rule-

based methods or settings with parallel data can yield reasonably good style transfer, but constructing parallel corpora is time- and cost-intensive. By contrast, it is comparatively easy to collect nonparallel data with style labels but without aligned pairs, making the labeled nonparallel setting practical. Yet under this setting, achieving strong content preservation and precise style control remains difficult. To address this, we propose CycleCVAE+CWS, a TST method based on a conditional variational autoencoder (CVAE) that combines a content word storage (CWS) mechanism with cyclic learning. By explicitly separating the portions that need not change during style transfer—i.e., content words—from the latent representation, we improve content preservation; further, by cyclically converting transferred text back to the original style, we perform pseudo data augmentation and enhance style transfer performance. This yields improvements in both content preservation and style controllability while maintaining the labeled nonparallel assumption. Our experimental evaluations demonstrate that CycleCVAE+CWS improves both content preservation and style control compared to conventional methods in both subjective and objective evaluations. Furthermore, even when compared to the latest methods using LLMs, our proposed method maintains equivalent content preservation while offering superior computational latency.

Second, there is no unified approach to handling disfluencies. Because text processing and speech synthesis have been studied separately, existing methods annotate disfluencies in an ad hoc, method-specific manner across the text and speech modules. When integrating TST and TTS into a single system, however, a consistent annotation scheme for representing disfluencies is desirable. We therefore propose a simple symbolization scheme for disfluencies that is shared by both TST and TTS. This scheme improves the representation of disfluencies on both sides while remaining compatible with advances in foundation models. Our experimental evaluations demonstrate that

disfluency annotations can enhance the style controllability and content preservation of TST, while also improving style reproducibility and naturalness in spontaneous speech generation.

In summary, our contributions are threefold: (i) we strengthen the TST base model under the labeled nonparallel setting; (ii) we leverage this model to construct a cascaded TST–TTS framework; and (iii) we further enhance spontaneous expressivity by introducing a disfluency annotation shared across both modules. Together, these contributions provide a new foundation for generating spontaneous speech from written text, with potential applications across a variety of human—machine interaction scenarios.

1 Introduction

1.1 General background

Speech communication constitutes one of the most fundamental and indispensable aspects of human social interaction. Spoken language enables the transmission of emotions and intentions, the sharing of knowledge, and the coordination of collaborative activities, thereby forming the basis of human society. The academic field of speech synthesis has developed with the aim of understanding and artificially reproducing this uniquely human ability. Speech synthesis is an interdisciplinary research area that spans linguistics, phonetics, cognitive science, and information engineering. It has evolved from early rule-based methods to statistical approaches, and more recently to deep learning–based models. Today, speech synthesis serves not only as a subject of scientific inquiry but also as a fundamental technology supporting a wide range of applications, including education, healthcare, dialog systems, and entertainment.

Among the major tasks of speech synthesis, one of the most representative is text-to-speech synthesis (TTS), which converts text into speech waveforms. Recent TTS technologies based on deep learning have achieved high naturalness, particularly in the context of reading-style speech, and have enabled the widespread adoption of voice interfaces such as Amazon Alexa and Apple Siri. Nevertheless, conventional TTS systems have been designed with the ideal of producing "fluent and error-free speech," and thus remain insufficient in their ability to reproduce natural human speech, namely

spontaneous speech.

The characteristics of spontaneous speech compared with those of reading-style speech include the following: (i) the linguistic form and content of speech are not predetermined, and no practice time is provided during recording; (ii) it includes non-verbal elements called spontaneous phenomena, e.g., laughter, coughing, interjections, pauses, and disfluencies caused by hesitating, speech errors, or word slurring; (iii) the presence of listeners may affect the speaker in some way [1]. These characteristics make spontaneous speech more challenging to collect data than reading-style speech, and they also make spontaneous speech modeling more challenging.

Among these characteristics, disfluency refers to elements that disrupt fluency in speech. Research on disfluency has progressed across multiple disciplines such as psychology, linguistics, and engineering, demonstrating that disfluencies not only reflect the speaker's uncertainty and cognitive processing but also affect the listener's comprehension, memory, and attention. Therefore, reproducing disfluency in speech synthesis is not merely a matter of improving naturalness, but is crucial for achieving speech that more closely resembles human spontaneity. In this thesis, we treat disfluency as a major class of spontaneous phenomena and focus on its synthesis.

1.2 Thesis Purpose, Scope and Approaches

Speech utterances are generally categorized into two primary types: "monolog," where the speaker delivers information in a one-way manner, and "dialog," which involves reciprocal interaction between the speaker and the listener. Considerable research on spontaneous speech synthesis within dialog has been undertaken. In contrast, research on speech synthesis for monologs is currently centered around synthesizing reading-style speech, and the technology for replicating spontaneous speech produc-

tion remains in its early stages of development. Moreover, spontaneous phenomena are beneficial not only in dialogs but also in monologs. In monologs, such as a lecture speech, incorporating spontaneous phenomena can facilitate memory retention and create a favorable impression of the speaker, enhancing the audience's attentiveness [2–5].

Motivated by these findings, this thesis focuses on monolog such as lectures and explanations, with the goal of realizing **spontaneous speech synthesis that incorporates disfluencies**. Furthermore, this thesis focuses primarily on filled pauses (FPs), a common type of disfluency. FPs refer to vocalizations such as "uh" or "um" produced to fill pauses in spontaneous speech, and they are among the most frequently studied disfluency in spontaneous speech synthesis. The characteristics inherent in spontaneous speech, including FPs, are discussed in detail in Chapter 2.

Below, we sequentially describe the problems encountered when implementing spontaneous speech synthesis for monologs and their respective solutions.

1.2.1 Problem 1: Challenge of generating spontaneous speech from written text

As will be discussed in Chapter 3, in most research on spontaneous speech synthesis, it is assumed that the text contains parts representing spontaneous phenomena. In practice, however, the available input often takes the form of written text, such as news articles, lecture manuscripts, or system-generated text from chatbots, which typically lacks spontaneous elements. Although manual conversion of written text into spoken-style text is possible prior to speech synthesis, the requirement for costly annotations makes the process impractical at scale and severely limits its practical applicability.

This mismatch between written inputs and the spoken-style text required by TTS systems raises an important research question: *How can spontaneous and natural speech*

be generated directly from written text?

Solution: TST–TTS Cascade

To address this gap, we propose a framework for generating natural and human-like lecture and explanatory speech directly from fluent written text. The approach combines text style transfer (TST) with TTS. TST transforms the stylistic attributes of text (e.g., sentiment, formality) while preserving its semantic content. By converting written text into spoken-style text and inserting appropriate disfluencies, TST provides realistic input for spontaneous TTS systems. This enables the synthesis of lecture or explanatory speech from existing written materials, thereby reducing the need to create new manuscripts. Nevertheless, the TST–TTS cascade approach presents two further challenges that must be addressed.

1.2.2 Problem 2: Performance of TST under nonparallel conditions

Another important challenge concerns the performance of TST. Using rule-based methods and deep learning methods with parallel corpora, one can easily provide a desired style to a text while keeping the semantics of the text. However, as will be discussed in Chapter 4, TST under nonparallel conditions—where no explicit source-target text pairs are available—remains a difficult problem. While recent advances in large language models (LLMs) have enabled higher-quality text generation than before, their application to TST faces issues in maintaining semantic consistency and computational efficiency.

Solution: CycleCVAE with Content Word Storage

To overcome these limitations, we adopt a conditional variational autoencoder (CVAE)–based approach combined with a content word storage (CWS) mechanism. This method explicitly separates content from style, thus significantly enhancing content preservation during style transfer. Furthermore, we introduce cyclic learning as a means of improving the effectiveness of style control without requiring parallel training data. This cyclic strategy, which involves generating pseudo-parallel data for cycle-reconstruction learning, has shown considerable improvement. Details of this method are provided in Chapter 5.

1.2.3 Problem 3: Inconsistent treatment of disfluency

A further challenge lies in the inconsistent handling of disfluencies across modules. In prior work, disfluencies have often been annotated or modeled in an ad hoc manner, with different approaches applied in different modules. This inconsistency arises partly because TST and TTS have traditionally been treated as separate research domains. However, when integrating TST and TTS into a unified system, it is desirable to establish a consistent framework for representing disfluency.

Solution: Disfluency-annotated TST + TTS

To address this issue, we introduce symbolic annotation strategies for representing disfluencies, extending the FP annotation strategy originally proposed in the context of TTS [6]. By incorporating these annotations into both TST and TTS, the proposed framework achieves consistent treatment of disfluencies across modules while enhancing expressive capacity. Moreover, the use of simple symbolic annotation ensures flexibility,

enabling the system to adapt to future advances in underlying models. This approach is described in detail in Chapter 6.

1.3 Contributions

The main contributions of this thesis are summarized as follows:

1. This thesis proposes a **cascade framework that combines TST with TTS**, enabling the direct synthesis of spontaneous speech from written text. This framework addresses the mismatch between written inputs and spoken-style requirements in TTS, thereby establishing a foundation for monologic spontaneous speech synthesis.
2. This thesis proposes a **new TST model for supervised nonparallel settings** that improves in both content preservation and style controllability, making it more suitable as input for TTS. This model enhances the naturalness of the generated speech and contributes to bridging the gap between written and spoken modalities.
3. This thesis introduces a **symbolic annotation strategy for representing disfluencies** in a consistent manner across both TST and TTS modules. This unification not only improves the expressive capacity of the system but also ensures adaptability to future advances in underlying models.
4. By integrating the above elements, this thesis establishes a **new basis for generating spontaneous speech from written text**. The proposed framework contributes both theoretically and practically to advancing speech synthesis research, with potential applications in education, dialog systems, and human–machine interaction.

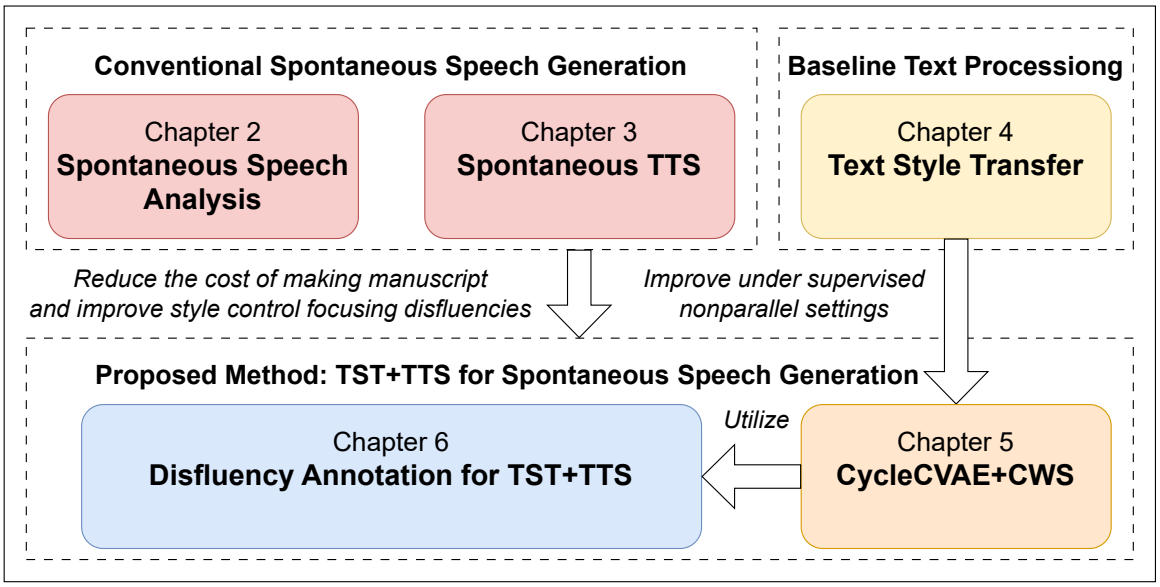

Figure 1.1: *Thesis overview.*

1.4 Thesis Overview

This thesis addresses the three problems outlined in Section 1.2. Figure 1.1 shows the overview of the thesis. The structure of the thesis is as follows. In Chapters 2–4, we provide the necessary background knowledge and a survey of related work. In Chapter 2, we introduce the foundations and existing studies on spontaneous speech analysis. we summarize the characteristics of spontaneous phenomena from multiple perspectives and discuss their roles from psychological and sociolinguistic viewpoints. Particular emphasis is placed on disfluency, clarifying its importance and considering its implications for speech synthesis. In Chapter 3, we review research trends in conventional TTS methods and spontaneous TTS methods. This chapter highlights the constraints that disfluencies impose on speech synthesis and discusses the limitations of current approaches.

In Chapter 4, we provide a detailed overview of TST and offer a comparative review

of existing TST methods. We introduce the fundamental concepts and classifications of TST. We then examine methods under parallel, supervised nonparallel, and unsupervised nonparallel settings and discuss latent representation learning approaches, explicit editing approaches, and the recent rise of LLM-based approaches. We also review studies on handling spontaneous styles such as disfluencies and dialects in the field of natural language processing (NLP), thereby positioning the contributions of this thesis.

In Chapter 5, we present the proposed *CycleCVAE+CWS* method for TST, which addresses Problem 2 described in Section 1.2.2. In Chapter 6, we introduce the integration of TST and TTS with *Disfluency Annotation*, proposed as a solution to Problem 1 (Section 1.2.1) and Problem 3 (Section 1.2.3). Finally, in Chapter 7, we provide a summary of the thesis and discuss future directions of this research.

2 Spontaneous Speech Analysis

Speech is the most natural medium of human communication and is commonly divided into reading-style speech and spontaneous speech. The former renders written scripts faithfully as audio, tends to be well organized, and contains few disfluencies. The latter comprises extemporaneous utterances produced in everyday conversations, lectures, and similar settings, and is characterized by disfluencies, incomplete sentence structures, and variability in intonation and pausing.

Traditional speech synthesis research has primarily focused on reading-style speech because it is easier to collect and its content is well defined. However, in real-world conversational and lecture scenarios, reading-style speech is often perceived as "unnatural" or "mechanical" owing to the absence of distinctive properties and functions inherent to spontaneous speech.

This chapter is organized as follows: Section 2.1 enumerates the characteristics of spontaneous speech, and Section 2.2 organizes the roles of these characteristics—especially disfluencies—from psychological and sociolinguistic perspectives. Section 2.3 surveys existing spontaneous speech corpora and analysis methods, and Section 2.4 discusses how these insights into spontaneous speech inform speech synthesis.

2.1 Characteristics of Spontaneous Speech

Compared with reading-style speech, spontaneous speech exhibits a wide range of properties, often referred to in recent work on spontaneous speech synthesis as spontaneous behavior [7, 8] or spontaneous (speech) phenomena [9, 10] (we refer to it as spontaneous phenomena). In this chapter, we organize these spontaneous phenomena into linguistic characteristics, prosodic characteristics, and nonverbal elements. In particular, we provide a more detailed account of disfluencies, which are the main focus of this thesis.

2.1.1 Linguistic Characteristics

- **Disfluency**. Although many definitions have been proposed across communities, no single standard has been universally adopted. In this thesis, we adopt the following definition: "disfluency can be defined as a phenomenon that interrupts the flow of speech and does not add any propositional content" [11]. There are also many ways to categorize disfluency; the main types are as follows.
  - **Filled pauses (FPs or fillers)**. A phenomenon in which a speaker "fills" the gap between words with a sound. Beyond signaling delays in preparation processes (speech planning) or hesitation by the speaker, FPs also function to indicate to the listener that the turn is continuing (turn-holding) [2–4, 12, 13].
  - **Repairs**. A phenomenon in which a speaker modifies a word mid-utterance or replaces it with another. Repairs reflect trial-and-error in spontaneous production and serve functions, such as error correction, specification, and outcomes of self-monitoring [12, 14–16].

  - **Repetitions**. A phenomenon in which part or all of a word is repeated immediately. Unlike repairs, repetitions are often produced not with a corrective intention but to bridge planning delays, add emphasis, or hold the turn, and thus do not change the final propositional content [11, 12, 17].

  Depending on the definition, silent pauses may also be counted as disfluencies [18]. In addition, purely vowel-like FPs are sometimes distinguished from lexicalized FPs and treated as *discourse markers* or *lexical fillers* [19]. Repairs and repetitions at the beginning of an utterance are often called *false starts* [11, 12]. In our proposed models, we deal with *FPs*—widely studied in previous research—and *stutter words*, i.e., fragments of words that result from repetitions and repairs.

- **Dialectal and speaker-specific expressions**. Linguistic characteristics that depend on the speaker's regional and social background. In spontaneous speech, these appear naturally and provide important cues to individuality and identity [20, 21]. Furthermore, research has been conducted on the aforementioned disfluency, particularly FPs, revealing that its usage varies depending on factors such as region, age, and gender [22, 23].

- **Incomplete sentence structures**. Spontaneous speech differs from reading-style speech in that it is not produced under strict speech planning. Furthermore, when attempting to speak long texts, humans are constrained by the biological phenomenon of breathing. Therefore, spontaneous speech is not always produced as complete sentences; ellipsis and interruptions are frequently observed [24, 25].

2.1.2 Prosodic Characteristics

- **Insertion of silent pauses**. Silent pauses in spontaneous speech tend to cluster around prosodic/clausal junctures and are systematically related to utterance planning and hesitation phenomena [26,27]. Their duration and placement reflect the speaker's planning, hesitation, and related cognitive processes [18, 24].

- **Prolongations**. A phenomenon in which the duration of a phoneme within or at the end of a word becomes longer than usual. Prolongations can occur alone or in combination with disfluencies. Prolongations are often regarded as a type of disfluency and, like FPs, indicate planning delays, hesitation, and turn-holding [13, 28].

- **Variability in intonation and accent**. Compared with read speech, spontaneous speech exhibits systematic differences in prosodic realization (including F_0-based patterns) that can be observed and, to some extent, discriminated using prosodic features [29]. Pitch-related variation also reflects the speaker's state (e.g., emotion), contributing to expressiveness [30]. Dialectal and variety-specific systems may further yield intonational patterns that diverge from the standard language [31, 32], and prosodic adequacy has been linked to listeners' perceived naturalness and even "aliveness" in perceptual settings [33].

- **Variation in speaking rate (tempo)**. Speaking rate varies both within and across speakers and differs by speaking style, indicating substantial tempo variability under natural conditions [34]. Speaking rate also changes with content and speaker state. For example, speakers may slow down for especially important information [35] or speed up when excited [30]. Like intonational variability, such tempo variation contributes to perceived expressiveness [36].

2.1.3 Non-linguistic Vocalizations

- **Emotional expressions such as laughter, crying, and sighs**. Although these lack propositional meaning, they convey the atmosphere of the conversation and the speaker's affect. For instance, laughter can signal affiliation, relieve tension, manage stance, and soften disagreement, thereby serving multiple functions in topic progression and stance management [37, 38].

- **Physiological vocal phenomena such as coughing, hiccups, and breathing sounds**. Although these phenomena likewise lack propositional meaning, they contribute to the perceived spontaneity of speech and can occasionally convey subtle nuances [39–41].

- **Listener responses and backchannels**. Especially in dialogic settings, listener responses constitute an integral component of spontaneous speech. Backchannels are a type of listener response, typically realized as brief utterances (e.g., "uh-huh") produced in reaction to a speaker's ongoing talk. Although backchanneling is not intended to initiate turn-taking, it is widely regarded as facilitating interactional quality [42, 43].

2.2 Roles of Spontaneous Phenomena

In this section, we discuss the communicative roles of spontaneous phenomena, with a focus on disfluencies. In the earliest studies, disfluency was examined mainly from a medical perspective (e.g., stuttering or aphasia) and in relation to language development in young children. Conversely, disfluency in healthy adults' spontaneous speech was treated as a "*redundant and useless element*" and was excluded from the

scope of linguistic research [13]. From the mid-1980s through the 1990s, psychology and psycholinguistics began to analyze the mechanisms and cognitive processes of language production by examining spontaneous speech as it naturally occurs. Around the same time, sociolinguistics also began to investigate the functions and roles of disfluencies in human–human interaction, with an emphasis on discourse. Other spontaneous phenomena have likewise been analyzed from psychological and sociolinguistic perspectives. Below, we summarize the main findings reported in prior work.

2.2.1 Role as a Predictive Cue

Listeners sometimes use disfluencies as signals that "new or complex information is about to appear." For example, Arnold *et al.* demonstrated that article fluency (e.g., "thee uh" vs. "the") affects how a listener interprets the following noun by monitoring listeners' eye movements toward displayed objects [3]. With fluent articles, listeners were biased toward previously mentioned objects; with disfluent articles, they were biased toward unmentioned objects. These results suggest that listeners use disfluency as a predictive cue to the newness or givenness of upcoming information.

Following this, Watanabe *et al.* investigated whether FPs affect listeners' predictions about the complexity of the following phrase in Japanese [4]. Participants listened to sentences describing simple and complex shapes on a computer screen and pressed a button as soon as they identified the corresponding shape. An FP, a silent pause of equal duration, or no pause was inserted immediately before the description. For native Japanese and non-native Chinese listeners, reaction times to complex shapes were shorter when FPs preceded the phrase than when no FP was present, indicating that FPs serve as a cue to complexity. Response times of the lowest-proficiency non-native listeners were unaffected by the presence of FPs, suggesting that the effect

depends on language proficiency.

Beyond disfluencies, other spontaneous phenomena can also function as predictive cues. For instance, speaking rate and intonation provide signals for turn-holding and turn-yielding [44], and the presence and depth of inhalation sounds can indicate whether the next utterance will begin and how long it may be [39, 40].

2.2.2 Facilitating Comprehension and Memory

Disfluencies have been reported to orient listeners' attention and facilitate processing, memory, and recall of subsequent information. Fox and Jean [2] tested whether English "uh" and "um" facilitate online comprehension using a word-monitoring task with natural spontaneous speech: target words immediately following a spontaneously produced *uh* or *um* were compared across unedited vs. digitally excised (FP-removed) versions of the same audio; the design was run twice (English and Dutch). Across both languages, hearing *uh* reliably sped recognition of the immediately following word relative to the edited version, whereas hearing *um* produced no reliable benefit or cost.

Fraundorf and Watson [45] showed, using a listen-and-recall task with recorded storytelling, that FPs facilitate recall not only at the utterance level but also at the discourse level. Participants who heard versions containing FPs were compared with those who heard versions containing silence or coughing of equal duration; the FP group recalled more. By contrast, a German web-based (non-laboratory) study using a similar paradigm reported opposite results [46], suggesting that language differences and experimental design (e.g., reduced control over distractors online) may affect outcomes.

2.2.3 Sociolinguistic Roles

From a sociolinguistic perspective, conversation-analytic work has clarified how disfluencies and other spontaneous phenomena function within discourse and interaction. Their relationships with social and cultural factors have also been examined.

- **Building interpersonal relations**. The use of FPs and silent pauses can shape impressions such as "polite" or "approachable." For example, moderate use of FPs can ease tension and foster rapport. In Japanese, "えーと (eeto)" typically signals cognitive planning, whereas "あのー (anoo)" can serve a socially oriented FP indicating consideration for the addressee (mitigating the utterance or providing a polite preface) [47]. The role of a disfluency also varies with position and activity type: for instance, *uh* or *um* occurring in the response slot after requests or proposals is understood to foreshadow a dispreferred response, whereas *uh* or *um* placed immediately after greetings at the start of a telephone call functions as a preface to the business at hand [48].

- **Constructing speaker identity**. Dialectal features and distinctive phraseology reflect the speaker's regional and social background and, in spontaneous speech, naturally highlight individuality and identity. Patterns of disfluency use can likewise correlate with social factors: for example, the ratio of *um* to *uh* tends to be higher among younger than older speakers and higher among women than men, a tendency reported for both American and British English [22, 23].

2.3 Corpora and Analysis Methods for Spontaneous Speech

To understand the characteristics of spontaneous speech and clarify their roles, large-scale, systematically collected corpora are indispensable. Numerous spontaneous-speech corpora have been constructed both in Japan and abroad and have been used in linguistic, psychological, and speech-technology research. This section surveys representative corpora and analysis methods.

2.3.1 Spontaneous Speech Corpora

- **Corpus of Spontaneous Japanese (CSJ)** [49]. The largest spontaneous speech corpus for Japanese, consisting primarily of public speeches such as academic presentations, lectures, and simulated public talks. It includes detailed annotations for disfluencies such as FPs and repairs, and has become foundational for spontaneous speech research in Japanese.

- **Switchboard Corpus** [50]. A large corpus of American English telephone conversations between previously unacquainted speakers on preselected topics. A substantial subset was annotated in the Penn Treebank for syntactic structure and disfluencies (e.g., fragments, interruptions, FPs, discourse markers), and it underlies many studies of dialog and disfluency.

- **Trinity Speech-Gesture Dataset (TSG/TSGD)** [51]. The corpus comprises 25 impromptu monologs by a male voice actor speaking Hiberno-English in a natural, colloquial style; recordings include high-quality audio (and, in some parts, motion-capture data) with an average monolog length of about 10.6 minutes. The

speaker talks without verbal interruptions, addressing a silent listener behind the cameras. This dataset has been widely used in spontaneous TTS research.

- **A Richly Annotated Mandarin Conversational Speech Dataset (MagicData-RAMC)** [52]. An open-source Mandarin conversational-speech corpus totaling roughly 180 hours, contributed by 663 speakers across 351 multi-turn conversations, with transcripts and rich metadata (speaker, recording environment/device, topics). Designed for ASR and conversational analysis.
- **Buckeye Corpus of Conversational Speech** [53]. High-quality recordings of free conversation with 40 speakers from central Ohio, with orthographic transcriptions and time-aligned phonetic labels—useful for detailed analysis of pronunciation variation and disfluencies in conversational English.
- **AMI Meeting Corpus (AMI)** [54]. A multimodal meeting dataset (about 100 hours) including naturally occurring and scenario-driven meetings, with audio, video, and annotations (e.g., topics, dialog acts). It supports research on multi-party spontaneous interaction.

2.3.2 Analysis Methods

Research on spontaneous speech has advanced along two complementary lines: large-corpus quantitative analysis and experimental validation (linguistic, psychological, and sociolinguistic). These findings provide an important basis for Chapter 3 ("Synthesis of Spontaneous Speech"), indicating which characteristics should be reproduced and how.

- **Manual annotation by expert annotators**. Specialists label disfluencies—FPs, repairs, repetitions, prolongations, and related phenomena—often following

established schemes (e.g., the Penn Treebank's disfluency annotation based on Shriberg's framework [55]). Such annotation enables the extraction of structural characteristics, frequencies, and positional patterns.

- **Computational and linguistic processing**. Automatic pipelines combine ASR outputs, forced alignment, and acoustic-prosodic feature extraction. Forced alignment tools such as the Montreal Forced Aligner (MFA) provide word/phone-level timestamps on spontaneous speech, facilitating precise localization of disfluencies. Acoustic features (e.g., F_0, intensity, spectral measures) are routinely extracted with tools like Praat [56] and openSMILE [57,58] for quantitative prosodic analysis and the automatic detection of FPs, silent pauses, and breaths.

- **Psychological and sociolinguistic experiments**. Behavioral tasks (e.g., gating, eye-tracking, word monitoring, recall) and conversation-analytic studies assess functional roles (prediction, attention, memory, turn management, stance) using stimuli that include or manipulate spontaneous phenomena. Meeting/discussion corpora (e.g., AMI) are often used to derive and test models of disfluency in multi-party interaction.

2.4 Implications of Spontaneous Speech Research for Speech Synthesis

Prior research has shown that spontaneous speech exhibits diverse characteristics and that these are not mere "imperfections," but elements that contribute to comprehension, memory, and listener impressions. These insights directly inform research on speech synthesis.

First, spontaneous phenomena are not "errors" to be eliminated but functional elements to be modeled. Conventional speech synthesis has targeted fluent, clear reading-style speech; however, by itself it can sound insufficiently "human" in spontaneous settings such as lectures and conversations. Therefore, disfluencies should not be treated merely as noise to be removed but as targets to be actively controlled and generated.

Second, spontaneous speech embodies stylistic diversity. The use of disfluencies and dialectal or speaker-specific expressions strongly shapes the individuality of a voice and the impressions it creates. In speech synthesis, reproducing regional and personal styles—beyond standardized, uniform reading-style speech—can improve naturalness and persuasiveness.

Third, the properties of spontaneous speech manifest at both the linguistic and the acoustic levels. For example, the placement and type of FPs can be controlled during text processing—e.g., text style transfer (TST)—whereas their actual acoustic duration and the variability of intonation must be realized during TTS. Hence, handling spontaneous speech in synthesis requires an integrated design that spans text processing and waveform generation.

These implications connect directly to the central objective of this thesis, "Spoken-text processing for spontaneous speech generation." Building on the features and functions identified in spontaneous speech research and reproducing them through both TST and TTS is key to achieving spontaneous and natural speech generation.

2.5 Summary

In this chapter, we organized the characteristics and roles of spontaneous speech from multiple perspectives. Section 2.1 showed that spontaneous speech exhibits a variety of properties not typically found in reading-style speech, such as FPs, repairs,

silent pauses, and variability in intonation. Section 2.2 then reviewed psychological and sociolinguistic findings indicating that these elements are not merely mistakes but serve functions that support listener comprehension and memory and express speaker individuality and social relations. Section 2.3 surveyed large-scale corpora and analysis methods that underpin spontaneous speech research, underscoring their value. Finally, Section 2.4 distilled the implications for speech synthesis: (i) treat spontaneous elements as targets to be actively reproduced; (ii) account for speaker- and context-dependent diversity; and (iii) design text processing and speech generation in an integrated fashion.

In summary, although spontaneous speech includes redundancy and apparent incompleteness, these properties play indispensable roles in information transmission and perceived human-likeness. Accordingly, insights from the analysis of spontaneous speech are essential for future speech synthesis research and constitute the foundation for the proposed **Spoken-text processing for spontaneous speech generation**. The next chapter builds on these analytical findings to survey prior work on synthesizing spontaneous speech and to clarify its limitations and open challenges.

3 Spontaneous TTS

As reviewed in Chapter 2, spontaneous speech exhibits a wide range of phenomena—e.g., FPs, repairs, and prosodic variability—that affect listeners' comprehension, memory, and impressions of the speaker. This chapter surveys prior attempts to reproduce these characteristics in speech synthesis.

Recent advances in deep learning have driven substantial progress in text-to-speech synthesis (TTS). In particular, neural approaches have yielded markedly more natural speech than traditional parametric methods [59]. However, most advances have focused on reading-style speech, and current systems still struggle to generate speech that faithfully reflects spontaneous phenomena.

This chapter is organized as follows: Section 3.1 surveys the development of conventional TTS techniques and clarifies their limitations in generating spontaneous speech. Section 3.2 then summarizes research that aims to reproduce spontaneous phenomena, as well as studies investigating their effects. Section 3.3 delineates the remaining gaps, thereby motivating the research questions addressed in the subsequent chapters.

3.1 Conventional TTS and the Challenge of Spontaneous Speech

3.1.1 Progress in Conventional TTS

Research on TTS has progressed through three major waves beyond the early rule-based and unit-selection [60] era.

1. **Parametric (HMM-based) speech synthesis.** Hidden Markov model (HMM)–based statistical parametric synthesis enabled flexible control of speaker identity, speaking style, and emotions via model adaptation and explicit acoustic-feature modeling. However, its reliance on vocoding and maximum-likelihood parameter generation led to over-smoothed spectra and a characteristic "robotic or breathy" timbre that limited perceived naturalness [61].
2. **DNN-based TTS (neural acoustic models and neural vocoders).** Deep neural networks replaced decision tree–based acoustic models, improving contextual modeling and reducing over-smoothing [62]. In parallel, neural vocoders such as WaveNet [63] generated high-fidelity waveforms from acoustic features, substantially narrowing the quality gap with natural speech.
3. **End-to-end TTS.** Sequence-to-sequence (seq2seq) models [64] learn text-to-mel mappings directly and, when paired with neural vocoders, achieve near-human naturalness on reading-style speech. Representative systems include Tacotron [65] and Tacotron2 [66]. Tacotron2 reported mean opinion scores (MOS) of 4.53 vs. 4.58 for studio recordings on the same U.S. English test set. Transformer TTS [67] improved efficiency via self-attention, and non-autoregressive models such as FastSpeech and FastSpeech2 [68, 69] advanced speed, robustness, and controllability.

More recently, VITS [70] unified direct text-to-waveform modeling with variational inference and adversarial learning, including a stochastic duration predictor for greater prosodic diversity. In addition, efficient GAN [71] vocoders—Parallel WaveGAN [72] and HiFi-GAN [73]—enable real-time or faster-than-real-time synthesis with competitive perceptual quality.

Thanks to these developments, single-speaker reading-style speech in controlled conditions is now often rated at or near "human-level" naturalness when trained on high-quality data and combined with strong neural vocoders. However, this progress does not automatically transfer to spontaneous speech, whose disfluencies, irregular timing, and prosodic variability remain under-modeled; we discuss these limitations in Section 3.1.2.

3.1.2 Limitations of Conventional TTS for Spontaneous Speech

Spontaneous speech exhibits various phenomena described in Section 2.1 that conventional TTS pipelines were not designed to model. We summarize key challenges and the corresponding bottlenecks in existing TTS designs.

1. **Front-end assumptions vs. disfluencies.** Typical TTS front-ends—including text normalization, prosodic phrasing, and grapheme-to-phoneme (g2p) conversion [74]—presuppose well-formed written text and often ignore or filter out tokens coresponding to spontaneous phenomena such as *uh* and *um*. Furthermore, because silent pauses and intonation are commonly determined using punctuation-dependent heuristics, cues of spontaneous phenomena, such as the insertion point and duration of FPs, are easily lost at the input stage. In neighboring pipelines (automatic speech recognition: ASR / natural language processing: NLP), disflu-

ency removal is explicitly applied before downstream processing, reinforcing a bias toward fluent text.

2. **Training data mismatch.** Widely used TTS corpora consist of clean, reading-style speech (e.g., LJSpeech [75], LibriTTS [76], JSUT [77]), whereas spontaneous corpora such as Switchboard, Buckeye, and CSJ contain overlaps, partial words, and telephone-bandwidth recordings, complicating TTS training and inducing domain mismatch in prosody and timing.
3. **Alignment and modeling constraints.** Seq2seq-based TTS systems assume monotonic alignments between phoneme (or grapheme) sequences and acoustic frames, enforced via location-sensitive, forward, or monotonic attention together with accurate duration prediction. Disfluencies present in the speech signal but omitted from the transcript violate these assumptions and cause alignment errors such as skips and repeats. Fong *et al.* [78] analyzed seq2seq-based TTS under transcription errors and found that, although inserted extraneous words are often skipped, performance degrades substantially under word substitutions and deletions. Moreover, when it is unknown which phonemes in spontaneous speech will be affected by disfluency—leading to repetitions, prolongations, or interruptions—the mapping from text to speech becomes ambiguous [79]. Non-attentive, duration-based architectures improve robustness, but they still assume nearly monotonic, minimally disfluent input [69, 70].
4. **Prosody representations with limited control.** Utterance-level style embeddings (e.g., global style tokens: GST [80]) capture coarse speaking style but offer limited, factorized control over pause insertion and hesitation timing. Recent controllable or LLM-prompted approaches broaden control yet still lack fine-grained spontaneous behaviors without explicit supervision [81].

5. **Evaluation gaps.** MOS [82] is widely used as a human evaluation metric for synthetic speech. In a typical MOS test, listeners are presented with individual synthesized utterances and asked to rate a specific aspect of the speech (e.g., perceived naturalness) on a Likert scale, typically five points; MOS is then computed by averaging the individual ratings for each system. However, there is little consistency across studies regarding which aspect is rated and how the prompt/scale is formulated [83]; further, as modern TTS quality has improved, MOS may lack sensitivity to small quality differences [84]. Moreover, in spontaneous TTS, functional requirements such as turn-taking, appropriate placement and duration of FPs, and consistency with dialog context are crucial, yet MOS alone cannot capture these properties [85–87].

Addressing spontaneous TTS requires (i) disfluency-preserving front-ends, (ii) relaxed or explicit alignment mechanisms for disfluencies, (iii) fine-grained control over the timing and duration of spontaneous phenomena, (iv) training on annotated spontaneous corpora, and (v) interaction-aware evaluation. Notably, when spontaneous phenomena are annotated and modeled, perceived naturalness in interactive settings can improve, as described in Section 3.2.

3.2 Research Trends in Spontaneous TTS

Spontaneous speech synthesis refers to TTS systems that mimic unplanned, conversational- or lecture-style speech. Unlike reading-style speech, spontaneous speech contains disfluencies, prosodic irregularities, and other spontaneous phenomena. Since around 2010, numerous studies in journals and international conferences have tackled the challenge of synthesizing natural, spontaneous speech. Broadly, prior work falls into three cate-

gories:

1. **Techniques for modeling spontaneous phenomena.** Methods that reproduce spontaneous characteristics in synthetic speech, for example, insertion of FP words and breathing sounds, control of intonation variability, and modeling of other spontaneous phenomena.
2. **Perceptual impact of spontaneity.** Evaluations of how modeling these features affects listener perception, such as naturalness, intelligibility, listener effort, and impressions of friendliness or personality.
3. **Comprehensive approaches combining modeling and evaluation.** Holistic studies that both propose a spontaneous TTS method and analyze its perceptual effects, often integrating elements of (1) and (2).

Below, we organize representative studies according to the first two categories above and summarize their key findings to provide an overview of research trends in spontaneous TTS.

3.2.1 Modeling Techniques for Spontaneous Phenomena

Early work in the 2010s focused on adapting statistical TTS (then dominated by unit selection and HMM-based methods) to handle disfluencies. Adell *et al.* pioneered "FP synthesis" for a unit-selection system [88]. Using a Spanish conversational corpus, they analyzed how inserting FPs affects duration and pitch. For example, syllables preceding an FP were significantly lengthened, and longer FP insertions often consisted of a silence plus the FP token. They implemented regression models to predict these local prosodic changes and integrated them into TTS. In listening tests, the MOS of FP-containing synthetic speech was statistically indistinguishable from that of the same

utterances without FPs, showing that, when handled appropriately, inserting FPs can preserve perceived naturalness. This is a foundational result for synthesizing disfluent speech.

Around the same time, Andersson *et al.* investigated HMM-based TTS trained on natural conversational speech [89]. They created an English conversational corpus rich in FPs, trained TTS on (i) conversational speech and (ii) the same speaker's reading-style speech, and also built a "blended" model switchable by a style context label (spontaneous vs. reading-style). AB preference tests along two axes (naturalness and conversational style) showed that, for utterances containing FPs, the speech trained only on conversational data was judged more natural and more conversational than the speech trained only on reading-style data. However, they also reported a negative result: in tests with speech from the blended model, 66.5% of listeners preferred the reading-style version without FPs over the spontaneous version with FPs for naturalness, while the presence of FPs did not significantly affect ratings of conversational style. This suggests that, even if conversational training improves overall style, overuse or poorly rendered synthetic FPs can sound unnecessary or mildly unnatural; we revisit perceptual effects in the next subsection.

A key challenge in spontaneous TTS is *where* and *how* to insert disfluencies for a given text. Dall *et al.* conducted a thorough study of FP insertion [90]. Using large English corpora (e.g., Switchboard, AMI), they trained models that predict the probability of FP insertion at various word boundaries, comparing several automatic methods (n-gram and recurrent neural language models, decision trees) and collecting human annotations in which participants marked natural FP positions in fluent text. Humans showed high agreement, and their selected positions tended to coincide with FP locations in real conversational speech. Perception tests further demonstrated that

FP placement matters: synthetic speech with FPs at *Top* positions (those most frequently selected by annotators) was preferred over *Unused* positions (never selected). However, when comparing *Top* to *NotFP* (no FP), listeners preferred *NotFP*. Given that their TTS system was HMM-based and trained on reading-style speech, this indicates that proper placement is necessary but not sufficient—FPs must also be rendered acoustically plausibly.

With the rise of seq2seq-based neural TTS in the late 2010s, researchers began training on conversational corpora to test whether neural models can inherently learn spontaneous patterns. Székely *et al.* made a notable contribution on "spontaneous speech synthesis from found data" [91]. They leveraged publicly available English conversational podcasts, segmented utterances by breaths, and performed ASR, recording not only words but also tokens such as FPs and laughter. They then trained Tacotron2 with Griffin–Lim [92], applying partial transfer learning from a large reading-style model to improve phoneme coverage. They compared models trained on (a) the original conversational data, (b) FP-removed data, and (c) a smaller but high-quality lab-recorded conversational set. Listening tests showed: (i) phoneme-level inputs and transfer learning improved pronunciation accuracy; (ii) training with FPs yielded a significantly more *authentic* impression than training without FPs; and (iii) in style-suitability evaluations, the podcast-trained conversational model was judged better suited to casual dialog prompts (and even informal public addresses) than a TTS trained on audiobook narration. In short, neural TTS can capture conversational idiosyncrasies from found data; including FPs in training can enhance authenticity, while FP-free variants can still be trained as cleaner alternatives.

Beyond FPs, breathing is another spontaneous feature of interest. Székely *et al.* targeted breath synthesis for spontaneous TTS [85]. Human speakers time breaths

at linguistically and prosodically appropriate locations, and disfluencies can disrupt breathing patterns. They automatically labeled breath events in a conversational corpus and incorporated them as tokens, training Tacotron2 with Griffin–Lim to predict breath boundary points. At inference, they avoided inserting labels corresponding to irregular/disfluent breaths, thereby rendering only fluent, pleasant-sounding breaths. This improved perceived fluency while suggesting that the approach could, conversely, be configured to reproduce disfluent breathing when desired.

More recent work has begun to address multiple spontaneous phenomena jointly. Qader *et al.* proposed an algorithm that generates disfluent sentences from fluent text, probabilistically integrating insertion positions not only for FPs but also for discourse markers and repetitions/repairs [93]. More recently, Li *et al.* introduced contextual encoders and a method for predicting FPs and prolongations from text via semi-supervised pretraining of a label predictor that leverages pseudo-labeled data from a multimodal detector trained on high-quality Mandarin conversational speech [94]. A different Li *et al.* explicitly modeled diverse conversational behaviors within a unified neural framework [10]: a spontaneous-style encoder first predicts bottleneck acoustic features from text, followed by a neural vocoder. The style encoder learns latent representations of both prosody and five annotated spontaneous phenomena (silent pauses, prolongations, fast speaking rate, liaison, stress), thereby capturing disfluency and prosodic variability. Both Li *et al.* papers reported significantly higher naturalness and listener preference over baselines, indicating that explicit modeling of spontaneous phenomena can markedly improve the reproduction of conversational style in TTS.

3.2.2 Perceptual Impact of Spontaneous Phenomena

How do FPs and other spontaneous phenomena affect listeners' perception of synthetic speech? This question has been studied from the perspectives of naturalness, comprehension, listening effort, and even personality impressions.

Early work in the 2010s suggested potential gains in perceived naturalness and spontaneity. For example, Adell *et al.* not only achieved MOS on par with FP-free synthesis but also reported user preference for FP-containing outputs in some dialogs [88]. Similarly, Andersson *et al.* found that synthetic speech trained on spontaneous speech was judged to exhibit a more conversational speaking style than synthetic speech trained on reading-style speech [89]. As noted in Section 3.2.1, however, inserting FPs can also hurt perceived naturalness in some settings.

Wester *et al.* [95] examined personality attributions to synthetic speech with vs. without FPs. They built an English unit-selection system whose acoustic inventory was derived from a reading-style speech corpus and then inserted disfluencies from the same speaker's spontaneous speech, such as FPs and discourse markers, into the synthesized utterances. They found that the presence of these disfluencies made the synthetic speech sound more nervous, less open, less conscientious, and less extroverted. Gustafson *et al.* [96] corroborated these findings with neural TTS: for English, inserting FPs increased perceived nervousness, reduced openness, and slightly reduced extroversion in reading-style synthesis, whereas no significant differences were observed for spontaneous-style synthesis. Taken together, these results suggest that inserting disfluencies into reading-style synthesis can be inappropriate or undesirable, at least for certain personality impressions. Cross-linguistic differences have also been reported: for Swedish, FP presence mainly increased perceived spontaneity, with little effect on other traits in both reading-style and spontaneous speech.

Kirkland *et al.* [97] investigated how FP placement, speech rate, and F_0 (fundamental frequency) shape perceived speaker confidence. Inserting an FP reduced perceived confidence, especially when inserted mid-utterance compared with utterance-initial insertion. These results suggest that well-rendered spontaneous phenomena can reliably convey hesitation in synthetic speech.

Another critical dimension is comprehension and cognitive load. Human-speech studies [2–4] show that listeners can benefit from disfluencies; for instance, the token "uh" can signal an upcoming delay and help prepare for complex words. To test this with synthetic speech, Dall *et al.* conducted a reaction-time experiment [98]. After hearing a sentence, participants reacted to a target word; conditions compared an inserted "uh" (embedded delay) with a silent pause (or no pause). With natural speech and with vocoded natural speech (degraded but not fully synthetic), the known facilitation effect was replicated. With HMM-based synthetic speech, however, the advantage disappeared: the synthetic "uh" yielded slower reactions than silence. The authors concluded that then-current TTS did not yet capture the subtle acoustic/prosodic cues by which real FPs aid processing, and suggested that improving FP naturalness could reduce listener effort; they further proposed reaction-time improvement as a success criterion for spontaneous TTS.

A recent study by Schettino *et al.* [5] partially supports this suggestion in a different setting. They synthesized Italian tourist-guide speech with and without disfluency (FPs, lengthenings, silent pauses) using Tacotron2 (+ WaveGlow [99]) continued from English pretraining and fine-tuned on an Italian dataset with disfluency annotations. In an AB test, a significantly larger number of participants chose the disfluent synthetic speech as more natural, yet a significantly larger number chose the fluent speech as more suitable for a virtual avatar. In a second experiment, a parallel written-recall

task and an impression test showed significantly higher recall scores for listeners of the disfluent speech, and synthesizing disfluencies did not harm perceived quality or liking.

Finally, speaker individuality raises a further question: *do speakers have idiosyncratic "disfluency styles," and can TTS reproduce them?* Matsunaga *et al.* addressed personalized spontaneous speech synthesis for Japanese [100]. They built a FastSpeech2–based system capable of replicating both a target speaker's timbre and FP usage. Using a lecture corpus rich in FPs and a dedicated FP-insertion predictor, they compared two models: one inserting ground-truth FPs from the actual speaker (simulating full personalization) and one inserting FPs predicted by a generic model. Metrics included naturalness, speaker similarity, and listener effort ("Which speech sample required less effort to listen to?"). A key finding was an entanglement between FP placement and word realization: precise placement strongly influenced naturalness, whereas FP type and prosody influenced perceived speaker identity. Ground-truth FP placement preserved both naturalness and the sense of the same speaker, whereas generic predictions sometimes placed FPs slightly off or used expressions atypical for the speaker, degrading naturalness or the illusion of identity. This highlights the need to capture not only *whether* to use disfluency but also *where* and *how* to realize it when aiming for speaker-specific or spontaneous TTS.

3.2.3 Summary of Findings

In summary, since 2010, the field has advanced substantially, driven by progress in modeling techniques (from HMM-based to neural approaches) and a deeper understanding of how style mismatch affects perception.

The area has evolved from unit-selection and HMM-based TTS—once constrained by data sparsity and simplistic prosody models—to neural approaches that leverage

large corpora and explicit modeling of spontaneous phenomena. A unifying question has persisted: *do disfluencies and other spontaneous nuances improve the quality and value of synthetic speech?* The emerging consensus is that, with appropriate modeling, these features can markedly enhance the realism and contextual appropriateness of TTS in dialogic or casual settings. At the same time, careful deployment is essential. Misplaced or poorly rendered disfluencies (e.g., out-of-context FPs) can detract from naturalness and confuse listeners. Recent work addresses these challenges, making it increasingly feasible to reproduce FPs and breath sounds with human-like naturalness.

Finally, as in both human–human and human–machine communication, outcomes vary by language and by experimental conditions. Results reported for one language, style, or scenario do not necessarily transfer to others. Cross-linguistic and multi-condition studies are therefore intrinsically valuable and necessary for a comprehensive understanding of spontaneous TTS.

3.3 Current Limitations and Research Challenges in Spontaneous Speech Synthesis

Building on the foregoing discussion, we summarize the key challenges to be addressed.

1. **From separation to integration of text (linguistic) and speech (acoustic) components**. Except for early explorations [88,89], research has increasingly specialized, with linguistic (text-side) and acoustic (speech-side) components studied separately. However, keeping linguistic decisions about the location and type of spontaneous phenomena on the text side separate from prosodic decisions on the

speech side tends to undermine discourse-level coherence. Integrated optimization (e.g., joint training or cascaded architectures with feedback) is required to tune semantic preservation and spontaneity simultaneously. This is also a central theme of our *Spoken-text processing for spontaneous speech generation* agenda.

2. **Establishing control granularity: location, type, and strength**. Most prior work controls either the location or the type; research on continuous control that also includes strength (e.g., duration and prosodic patterns) remains nascent, with serious attempts only recently emerging.
3. **Choice and alignment of the generative backbone**. TTS backbones continue to advance, yet in current studies of spontaneous speech synthesis—especially those focusing on perceptual effects—Tacotron2 appears to be the most widely used, likely because it is accessible to researchers outside core speech synthesis. At the same time, as evidenced by large gaps between statistical and neural TTS results, backbone performance substantially affects experimental outcomes. TTS based on diffusion probabilistic models has shown promise for robustness to noisy transcripts and uncertain alignments, which is attractive for disfluency-rich spontaneous synthesis. Nevertheless, such backbones still face challenges, including slow generation and complex control interfaces. Further advances in alignment-free/relaxed-alignment models are anticipated.
4. **Evaluation design**. Beyond short utterances, standardized evaluations that measure effects on comprehension, memory, and impressions in context (in-dialog AB tests, task performance, user studies) are needed to enable comparisons that go beyond MOS-style naturalness ratings.

3.4 Summary

In this chapter, we surveyed research trends for synthesizing spontaneous speech. As reviewed in Section 3.1, conventional text-to-speech synthesis (TTS) has dramatically improved the naturalness of reading-style speech through deep learning, yet remains limited in reproducing spontaneous phenomena. Building on this, Section 3.2 reviewed recent progress from two perspectives: (i) reproducing spontaneous phenomena and (ii) evaluating how spontaneous phenomena affect the perception of synthetic speech. These studies contribute to partial reproduction and evaluation but still fall short of a comprehensive framework.

Section 3.3 then organized open challenges: (i) integrated optimization across linguistic and acoustic levels, (ii) establishing control granularity that unifies location, type, and strength, (iii) advancing robust generative backbones, including diffusion models, and (iv) designing new evaluations that measure contextual appropriateness and cognitive effects.

Taken together, research on spontaneous speech synthesis is still at an early stage: despite progress on generating individual phenomena, natural spontaneity at the discourse level has not yet been achieved. Moving forward, spontaneity should be introduced systematically via frameworks that integrate text processing with speech synthesis. This is the core of our proposal for "direct spontaneous speech generation from written text," and it connects to the approach developed in subsequent chapters, which uses *text style transfer (TST)* as the other foundational component.

4 Text Style Transfer

As surveyed in Chapter 3, reproducing spontaneous phenomena such as disfluencies cannot be achieved through waveform generation alone; manipulating style at the text-processing stage is indispensable. This chapter therefore focuses on *text style transfer (TST)* [101], the task of rewriting an input while preserving its propositional content and converting it to a target style.

Regarding the notion of style in TST, we adopt a data-driven operationalization rather than a traditional linguistic definition. Instead of treating style as an invariant property, we view it as a *set of attributes* that vary across corpora. The motivation is practical: deep neural network models require large corpora to learn style, yet perfectly matched, large-scale datasets for every target style rarely exist. Consequently, except for a few manually annotated resources built from linguistic criteria, recent dataset-construction efforts typically mine metadata that link corpora to particular attributes. A canonical example is the Yelp review dataset [101], widely used in TST, which is constructed by treating low-star reviews as a "negative" corpus and high-star reviews as a "positive" corpus. Note, however, that positive/negative are more content-related than stylistic in the linguistic sense.

Historically, TST research centered on sentiment transfer (positive vs. negative) and formality transfer (informal vs. formal). More recently, applications have broadened to a wider set of styles—e.g., speaker persona, dialect, prosodic correlates, and even the insertion of disfluencies—bringing TST closer to spontaneous, conversational us-

age. The research paradigm has also shifted. Earlier work emphasized achieving both content preservation and style control under nonparallel data. Since 2023, with the rise of large language models (LLMs), prompt-based and few-shot TST has gained prominence, substantially improving controllability and flexibility while leaving open challenges in content preservation and reproducibility.

Advances in TST directly inform spontaneous text-to-speech synthesis (TTS): to insert disfluencies, transcribe speaking style into text, and maintain discourse-level coherence, one must first manipulate textual representations appropriately.

This chapter is organized as follows: Section 4.1 outlines basic concepts and taxonomies of TST. Section 4.2 reviews TST methods, such as latent-representation learning, explicit-editing approaches, and LLM-based techniques. Section 4.3 then discusses spontaneous style control in natural language processing (NLP), and Section 4.4 situate our proposed methods within this landscape and connects to in Chapter 5.

4.1 Taxonomy and Core Problems of TST

Drawing inspiration from neural image style transfer [102] and neural machine translation (NMT), TST has attracted considerable attention [103]. TST aims to simultaneously achieve *content preservation* and *style control*. Consequently, the (in)separability of content and style remains a central challenge. This section outlines representative axes of classification and the core problems faced in TST.

4.1.1 Axes of Classification

1. **By data conditions**

    - **Parallel (paired) data.** When input–output pairs are available, supervised

learning can achieve high-quality transfer. In practice, however, constructing large, clean parallel corpora is costly and often infeasible.

- **Nonparallel with style labels.** A realistic setting in which only unpaired collections for different styles are available, possibly with style labels. Key techniques include content–style separation, cycle consistency, and carefully designed latent representations.
- **Nonparallel without labels.** This is an easier data-collection regime but a harder learning problem, as models must infer style without explicit labels.
- **Hybrid / semi-supervised.** Small, high-quality parallel data are used to bootstrap large "silver" corpora via pseudo-labeling, which are then leveraged for training at scale.

2. **By transformation method**

- **Rule-based / statistical.** Hand-crafted transformation rules or classical statistical machine-translation (SMT)–style approaches.
- **Implicit-disentanglement.** Map content and style to separate latent variables and replace only the style component (e.g., autoencoders with adversarial or variational objectives).
- **Explicit editing.** Remove or mask style-bearing elements in the input (e.g., sentiment words, honorifics) and regenerate conditioned on the target style. Intuitive and controllable, but often vocabulary-dependent.
- **Prompted / LLM-based.** Use LLMs with instructions or few-shot exemplars to perform transfer. Highly flexible and powerful, yet stability and reproducibility remain concerns.

4.1.2 Core Challenges

1. **Trade-off between content preservation and style control.** Stronger transfer risks altering or dropping semantic content; prioritizing content can yield weak style changes. Balancing this trade-off is a central research theme.
2. **Fluency and naturalness.** Word-substitution methods are prone to grammatical errors or awkward phrasing. Neural generation (sequence-to-sequence, Transformers, LLMs) improves fluency, but maintaining coherence at the paragraph- or discourse- level remains challenging, and LLM rewrites can hallucinate unsupported details.
3. **Evaluation uncertainty.** Automatic metrics along three axes—style fit (e.g., style-classifier accuracy), content preservation (e.g., BLEU [104], BERTScore [105]), and fluency (e.g., perplexity)—often diverge from human judgments or task-specific utility, and the metrics used in each study are not uniform. "LLM-as-a-judge" is gaining traction, but its reliability and bias characteristics are still under active investigation.

4.2 Text Style Transfer Methods with Deep Learning

Figure 4.1 shows the outline of TST methodology types. This section surveys recent deep-learning approaches to TST, organized first by data conditions and then by transfer method.

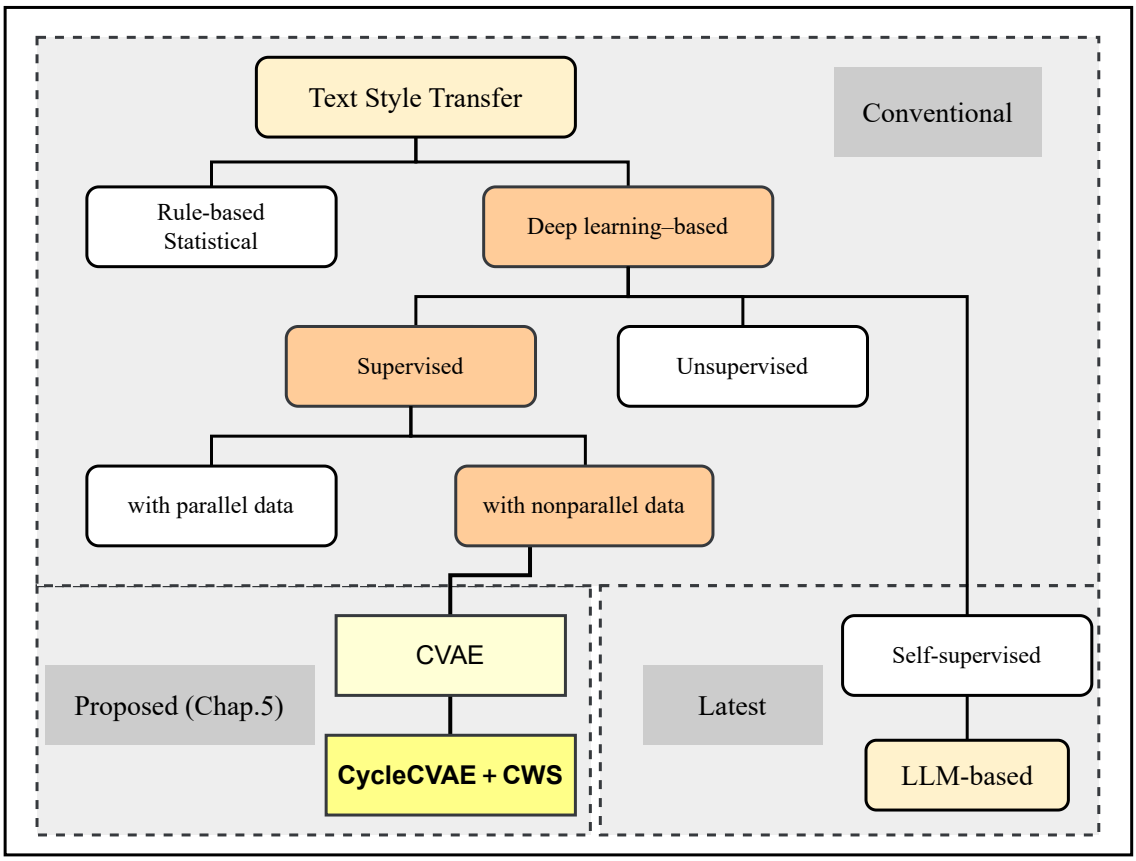

Figure 4.1: *Outline of TST methods.*

4.2.1 Methods Using Parallel Corpora

In settings with parallel corpora [106, 107], sequence-to-sequence (seq2seq) models are commonly used, as in neural machine translation (NMT), trained on paired texts in different styles. For example, Jhamtani *et al.* [106] extended a seq2seq model with a copy mechanism [108] to selectively copy tokens from the input while transforming Modern English into Shakespearean English. This mechanism exploits the substantial lexical overlap between Shakespearean and Modern English and helps with infrequent nouns that a vanilla seq2seq model may fail to generate. In addition, Carlson *et al.* [107]

proposed a TST model based on seqence-to-sequence (seq2seq) with attention [109] and evaluated it on a collected parallel corpus of biblical prose styles.

4.2.2 Methods Using Nonparallel Corpora with Style Labels

When only unpaired corpora with style labels are available, the goal is to perform style transfer without access to paired examples. Most existing TST studies fall into this category due to the scarcity of large, clean parallel corpora. Methods can be grouped by how they disentangle style from content.

Explicit-Editing Methods

While latent-representation approaches have become mainstream, *explicit-editing* offers a more intuitive and interpretable alternative. For attributes such as sentiment, style is often signaled by characteristic words or phrases (e.g., “nice,” “good”). A straightforward solution is therefore to replace such style markers to change the sentiment of a sentence.

A representative framework is Delete–Retrieve–Generate (DRG) [110]: given an input, (i) identify and *delete* style markers (e.g., by frequency or salience), (ii) *retrieve* a reference sentence from the target-style corpus that shares content words with the source, and (iii) *generate* a target-style sentence by combining the retrieved markers with the source content using rule-based templates or a seq2seq generator. Related explicit-editing methods include *prototype editing*, which selects a target-style prototype and edits it toward the source content, and *mask-and-infill*, which masks style-bearing spans and uses a language model to infill them under the target style. These methods excel at local edits, improve controllability, and tend to preserve alignment

between input and output.

Strengths of explicit-editing include relatively simple models, shorter training time, and high explainability (the edited spans are directly observable). However, they also have clear limitations: beyond sentiment, many styles are not tied to simple lexemes; as a result, explicit-editing is often limited to sentiment transfer and struggles to generalize to other styles. Because it depends on retrieval and substitution, output can become formulaic; moreover, enforcing consistent style over long documents or paragraphs remains difficult.

Implicit-Disentanglement Methods

To separate content and style, *implicit* methods first learn latent variables for each, then combine source content features with target-style features to generate the output. Building on encoder–decoder architectures, numerous techniques have been proposed to strengthen style control, including VAE+classifier objectives [111], back-translation [112], and adversarial learning [113, 114]. However, studies report a trade-off: stronger style control via discriminators/classifiers often harms content preservation. To mitigate this, auxiliary objectives for content retention—such as bag-of-words overlap loss [113], noun-overlap loss [115], and cycle-consistency losses [116]—have been explored. Laugier *et al.* [117] combined the large pretrained model T5 [118] with cycle consistency, though content preservation remained challenging.

4.2.3 Methods Using Nonparallel Corpora without Style Labels

Conventional fully unsupervised methods

A long-standing view is that, under limited compute, modeling latent style features purely from unsupervised data is highly challenging. Even so, there have been attempts. For instance, CP-VAE [119] targeted sentiment in a fully unsupervised way and reported over 90% accuracy in detecting latent style features. Using the same model, they also separated style and content on a corpus with mixed topics and obtained good results by treating topic as the style attribute. However, most fully unsupervised TST results concern content-adjacent attributes such as sentiment or topic, so generality to other styles remains unclear. Moreover, theory and empirical evidence suggest that stable content–style disentanglement without inductive biases or supervision is fundamentally impossible; when it appears to work, it may hinge on luck or configuration choices [120].

Prompt-based (LLM) methods

Despite the lack of labels, a simple yet powerful remedy—scaling data and parameters—has led to breakthroughs. Recent LLMs, such as LLaMA [121] and GPT [122–124], have dramatically changed the TST landscape. Instead of relying solely on latent disentanglement, LLM-based approaches leverage stylistic diversity learned from massive pretraining to perform more flexible, high-quality transfer.

The most direct approach is to issue instructions to an LLM (e.g., "Rewrite this sentence in a formal style"). With few-shot prompting and step-by-step reasoning, LLMs can produce outputs that follow the requested style [125].

Research is currently underway on techniques such as combining and comparing multiple texts generated by LLM to create superior transfer texts [126,127]. Unlike classical sentence-level TST, LLMs further enable efforts to maintain document- or paragraph-level consistency—for example via hierarchical control that coordinates sentence-level edits with paragraph-level coherence, or via discourse-aware frameworks—capabilities that are particularly important for context-dependent styles such as lectures or conversational narratives [128].

LLM-based TST offers the advantage of accommodating a wide range of styles, but challenges remain regarding output stability and reproducibility due to hallucinations, as well as content preservation.

As LLM-based TST advances, evaluation practices are being revisited. In addition to BLEU and style-classifier accuracy, *LLM-as-a-judge* has been proposed to assess style fit and content preservation, potentially aligning better with human judgments. However, reliability and consistency remain open issues.

Finally, our target application—spontaneous speech generation—often requires lightweight, fast text-side style transfer. CVAE-based methods can offer advantages in computational efficiency and controllability relative to LLMs. At the same time, LLM-based TST is highly flexible and expressive; we view it as complementary and discuss LLM-based extensions in Chapter 7.

4.3 Controlling Spontaneous Style in NLP

The goal of this study is to use TST to appropriately manipulate spontaneous characteristics at the text-processing stage and to bridge them to speech synthesis. From this perspective, it is important to review how NLP has handled spontaneous style. Here, we focus on two phenomena—disfluency and dialect—and survey recent trends.

4.3.1 Disfluency

Disfluency in text has been a target of control for various purposes. For example, research that detects and removes disfluencies from ASR outputs to obtain more readable transcripts predates the notion of TST. Beginning with rule-based approaches, advances in neural networks—particularly LSTMs [129] and Transformers [130]—have led to treating disfluency detection as a sequence-labeling problem [131–134].

Research on inserting disfluencies, in contrast, started mainly for data augmentation in ASR and dialog systems and is more recent than detection and removal. Early simple methods such as randomly duplicating words were explored, but the inserted patterns were monotonous and deviated from real data [135].

To address this, Yang *et al.* [136] proposed a planner–and–generator framework for disfluent sentence generation. A planner is first trained to decide where to insert disfluent segments and how long they should be; a generator then follows the plan to produce FPs and repairs consistent with the content. This two-stage generation yields diverse and natural disfluencies unattainable by random insertion and, when used for pretraining, substantially improves detection performance over prior methods.

More recently, LLM-based disfluent text generation has also been explored. For instance, Marie [137] fine-tuned a pretrained T5 [118] model with a small amount of fluent/disfluent parallel data via few-shot learning to build a disfluency paraphrase generator. Augmenting user utterances in an English dialog dataset with artificial FPs improved both dialog state tracking accuracy and response generation of systems trained on the augmented data.

However, these studies aim at generation for data augmentation—i.e., producing diverse data—rather than style conversion *per se*; consequently, they often do not impose the constraint of preserving the original text's meaning. This distinguishes

them from TST, whose goal is to convert a given text into another style while preserving its semantics.

4.3.2 Dialect

Control of styles across standard Japanese and dialects has long been framed as an "intralingual machine translation" task and studied as such. Consequently, most approaches rely on parallel data, and in TST—where nonparallel settings are prevalent—there have been few studies that treat dialect as a style. Recently, methods that leverage LLMs in combination with rule-based components have been proposed [138].

In Japan, the "Corpus of Japanese Dialects (COJADS)" [139], consisting of dialog speech in the dialects of all 47 prefectures and the corresponding standard-Japanese transcripts, was released in 2023. COJADS includes utterances from various regions (e.g., Kansai dialect), aligning standard-Japanese text with dialectal text written in kana. Using this resource, a proof-of-concept dialog system that answers in a dialect when asked in standard Japanese was also developed [140].

Dialectal style is closely tied not only to lexical and phrasal differences in text but also to phonological and prosodic properties in speech. Simple word substitution is therefore insufficient; coordinated text transfer and speech synthesis are essential. The development of corpora that align speech with text, such as COJADS, is expected to provide a crucial foundation for future work on dialectal TST and TTS.

4.4 Remaining Gaps and Positioning of This Thesis

In this chapter, we surveyed the research landscape of TST, organizing conventional frameworks and recent trends. Among them, methods in the lineage of *implicit-*

disentanglement trained on nonparallel data depart from machine-translation-style approaches in that they do not require parallel corpora. Unlike explicit-editing approaches, they are not restricted to a single style and can, in principle, scale to multiple target styles. Moreover, in cascaded pipelines—i.e., settings where low latency is required—they can be computationally more efficient than recent LLM-based approaches. At the same time, several challenges remain:

1. **Insufficient content preservation and style controllability**. Compared with text generated by state-of-the-art NLP systems based on recent LLM advances, the label-supervised nonparallel TST frameworks still lag in performance. This work primarily targets this issue by proposing a new content-preservation mechanism compatible with recent methods, together with a data-augmentation-oriented cyclic training scheme that complements it.
2. **Coherence at the long-text/discourse level**. While sentence-level style transfer is becoming feasible, maintaining style consistently across an entire discourse remains difficult. This is particularly critical for long-form speech generation such as lectures or dialogs, where inter-sentential consistency is indispensable. Recent studies are beginning to explore frameworks that jointly enhance content preservation and style control for long documents [128].
3. **Evaluation and reproducibility**. For spontaneous-style conversion, there is no widely accepted set of metrics or benchmarks, making cross-paper comparisons difficult. In particular, evaluating the naturalness of disfluency insertion and discourse appropriateness involves strong subjectivity; as with the TTS component, the design of objective metrics remains an open problem.

The proposed text-control module, *CycleCVAE+CWS*, belongs to the family of

implicit-disentanglement methods trained on nonparallel data. As outlined in Section 1.2.2, one of our goals is to address the aforementioned limitations in content preservation and style controllability within this framework.

4.5 Summary

In this chapter, we provided an overview of studies on text style transfer (TST). Section 4.1 reviewed the basic concepts and taxonomies of TST and organized its general challenges. Section 4.2 surveyed prior work along the two axes introduced in Section 4.1—data usage and transfer method—and Section 4.3 examined recent trends in text-level control of spontaneous phenomena. Section 4.4 identified remaining shortcomings and open issues in existing approaches and positioned our study as proposing CycleCVAE+CWS, a new method that strengthens content preservation and style control with the aim of improving performance.

Building on these points, Chapter 5 presents the formulation and experimental validation of the method based on CycleCVAE+CWS, and Chapter 6 demonstrates *spontaneous speech generation via the integration of TST and TTS*, thereby proposing a framework that surpasses the limitations of conventional spontaneous speech synthesis.

5 TST using CycleCVAE+CWS

This chapter addresses the second challenge introduced in Section 1.2.2: improving text style transfer (TST) under nonparallel conditions. We present a framework termed *cyclic conditional vatiational autoencoder* (CycleCVAE) + *content word storage* (CWS). Although this method has been reported previously in a journal article, here we reposition it within the overall flow of this thesis and consolidate it as a foundational component for spontaneous speech generation.

This chapter is organized as follows: Section 5.2 introduces CVAE-based TST. Section 5.3 proposes extending CVAE with CWS to improve content preservation and adding cyclic learning to improve style control. Section 5.4 describes our experiments evaluating the effectiveness of CWS and cyclic learning proposed in Section 5.3 and the usefulness of the proposed method as a preprocessor for TTS. Finally, Section 5.5 concludes our findings and discusses future works in the TST part.

5.1 Introduction

TST is the task of transforming the *style* of a text into a target style while preserving its meaning. The "style" here can range from sentiment and formality to personal attributes (gender, age, personality, etc.), political stance, aggression, and more.

Even without considering its application to spontaneous speech synthesis, it is beneficial to control the style of texts in coordination with the style of speech. Studies

in sociolinguistics and psychology show that the attributes and persona characteristics of a speaker (or writer) can be identified using text alone information [141–143]. For example, “この料理は非常に美味しいです (This dish is exceptionally delicious)” and “このごはん、とってもおいしい！ (This food is super yummy!)” convey the same meaning, “This food tastes very good,” yet the impression each sentence gives about the speaker differs markedly in terms of age and personality.

To achieve TST with minimal time and effort, a method that can be trained using a nonparallel corpus is required. Using rule-based methods [144] and deep learning methods with parallel corpora [106, 107], one can easily provide a desired style to a text while keeping the semantics of the text. However, creating a large number of conversion rules and a large parallel corpus requires much manual work, making it both time-consuming and impractical for various scenarios. In contrast, the *text with a specific style but without pair data* is relatively easy to obtain from various media. Therefore, we can create a nonparallel corpus with a style label that can be constructed using these texts at a lower cost than in the case of creating a parallel corpus.

The most popular TST methods for nonparallel corpora are autoencoder (AE)- and variational autoencoder (VAE) [145]-based methods, which can provide disentangled latent representations [111, 146]. The VAE-based methods model the content and style of a text separately and convert texts by manipulating only the learned style. These models enable style transfer without a parallel corpus by learning reconstruction rather than direct conversion.

As discussed in Section 4.1.2, a central difficulty in TST is balancing (i) *content preservation* and (ii) *style controllability*. The challenge is exacerbated in the nonparallel setting, where the absence of paired corpora necessitates careful inductive biases and training constraints. Prior work has frequently reported semantic drift (content

Table 5.1: *Style-transferred text example from fluent to disfluent.*

Original Text	文学研究としてはそれでも構わない訳ですが
Translation	As for literary research, that's fine, but...
Transferred text	文学研究としてはえー構わない訳ですがまー
Translation	As for literary research, well, that's fine, but, ah...

alteration or loss) alongside under- or over-transfer of style.

To address these issues, we propose a simple yet effective method that combines a CVAE [147] with CWS, a mechanism for preserving content words to improve content preservation during TST. We also introduce *cyclic learning* as a simple data-augmentation-like strategy to improve style control performance because CVAE and CWS alone do not directly enhance style control. By creating a pseudo-reference text and then converting it back to the original text, a model learns not only reconstruction but also conversion. Together, these components improve the trade-off between content preservation and style control and further enable the natural introduction of spontaneous phenomena.

We conduct a bi-directional style transfer experiment on Japanese texts targeting "disfluency" and "dialect" using the proposed method, intending to apply the method to spontaneous speech generation. From the experimental evaluation, we show that (i) CWS improves content preservation without compromising the style control performance and naturalness, (ii) cyclic learning using a pseudo-reference text improves style control performance without a parallel corpus, and (iii) the proposed method has a positive effect on perceptual style reproducibility of speech when combined with TTS as a downstream task. Examples of disfluency style-transferred text from the proposed method are shown in Table 5.1.

5.2 CVAE-based Text Style Transfer

CVAE [147] is a generative model based on unsupervised representation learning and supervised style transfer. CVAE is therefore used for TST utilizing a nonparallel corpus [111].

The upper part of Figure 5.1 shows a schematic of a CVAE-based TST model. CVAE models the conditional distribution $p(\boldsymbol{x}|y)$ of target texts $\boldsymbol{x}$ given style y. CVAE introduces the latent variable $\boldsymbol{z}$ to represent textual contents that are independent of the style labels y. The latent variable $\boldsymbol{z}$ is expected to capture the textual content as a compressed form of the input text $\boldsymbol{x}_{1:M}$ of M-length (hereafter denoted $\boldsymbol{x}$ if not otherwise required). The style class label y indicates the explicit style of utterances such as idiolect, dialect, or disfluency level. The TST using CVAE can be conducted as follows: first, a content representation $\boldsymbol{z}$ is extracted from the text with a source style $s \in y$; second, the output text $\hat{\boldsymbol{x}}_{1:N}$ of N-length (hereafter denoted $\hat{\boldsymbol{x}}$ if not otherwise required) with the target style is generated from the content representation $\boldsymbol{z}$ by conditioning a model with the target class label $t \in y$.

To learn the conditional distribution $p(\boldsymbol{x}|y)$, CVAE is trained by maximizing the variational lower bound L similar to a conventional VAE. Maximizing the variational lower bound is an approximation of maximizing the likelihood of the marginal probability: $p(\boldsymbol{x}|y) = \int p(\boldsymbol{x}|\boldsymbol{z}, y)p(\boldsymbol{z})d\boldsymbol{z}$

$$\begin{aligned}\log p(\boldsymbol{x}|y) &\geq L \\ &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z}, y)] - \mathrm{KL}[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})], \end{aligned} \tag{5.1}$$

where KL is the Kullback–Leibler divergence. The first term in R.H.S. optimizes the reconstruction probability of observed texts given the annotated style label y and latent feature $\boldsymbol{z}$. The second term in R.H.S. works for regularization to enforce the latent

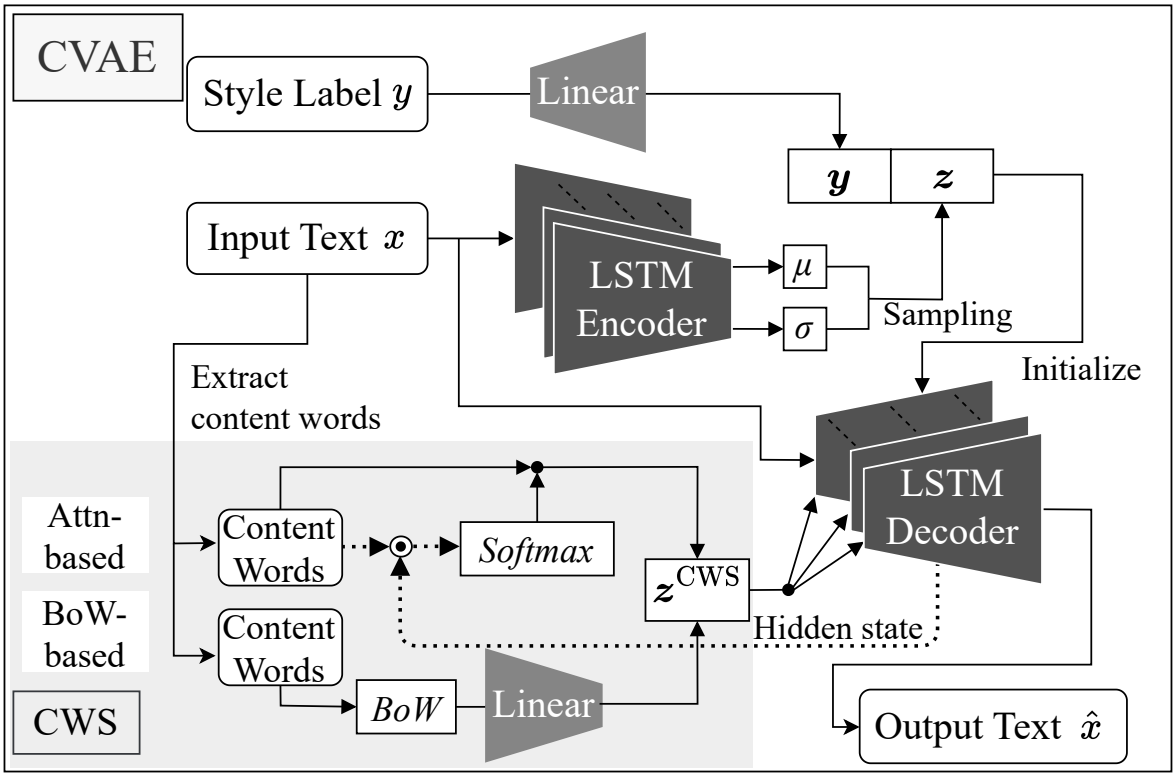

Figure 5.1: *Schematic of CVAE and CWS.*

feature $\boldsymbol{z}$ to capture only content information by introducing the prior distribution $p(\boldsymbol{z})$ of $\boldsymbol{z}$. The approximate posterior distribution q is introduced to model the distribution of the latent feature $\boldsymbol{z}$. Minimizing the KL divergence between the approximate posterior and prior limits the latent representation capacity. This behavior is expected to exclude style information from the latent feature $\boldsymbol{z}$.

The probabilistic model of CVAE can be implemented on the basis of the autoencoder structure. The approximate posterior $q(\boldsymbol{z}|\boldsymbol{x})$ can be implemented as an encoder to encode the input text $\boldsymbol{x}$ into the content representation $\boldsymbol{z}$. We design the distribution of the content representation as a Gaussian distribution, so the encoder models the mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ parameters of the Gaussian distribution. The content latent variable can be sampled by the location-scale relationship: $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \boldsymbol{\epsilon}$, where

$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. This reparametrization trick enables the joint optimization of the encoder and decoder without the isolation of backpropagation paths caused by sampling [145]. The output probability $p(\hat{\boldsymbol{x}}|\boldsymbol{z}, y)$ can be implemented as a decoder to decode the output text $\hat{\boldsymbol{x}}$ from the content representation $\boldsymbol{z}$ and style label y. We model the output probability as an autoregressive distribution, where output probability is decomposed into the product of probabilities of each output token $p(\hat{\boldsymbol{x}}|\boldsymbol{z}, y) = \prod_{n=1}^{N} p(\hat{\boldsymbol{x}}_n|\hat{\boldsymbol{x}}_{n-1}, \boldsymbol{z}, y)$.

During training, CVAE is trained to reconstruct input texts given the style label y on the basis of Eq. (5.1). This training scheme enables us to use nonparallel corpora consisting of texts $\boldsymbol{x}$ with the source style alone, without depending on their parallel texts $\hat{\boldsymbol{x}}$ with the target style. Note that it is expected that the length M of the input is equal to the length N of the output.

During inference, the latent content feature $\boldsymbol{z}$ is firstly sampled from the approximate posterior $\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x})$ by feeding a source text $\boldsymbol{x}$. A style-converted text $\hat{\boldsymbol{x}}$ is then generated from the output distribution $\hat{\boldsymbol{x}} \sim p(\hat{\boldsymbol{x}}|\boldsymbol{z}, y)$ by feeding the sampled content feature $\boldsymbol{z}$ and target style label y.

The concrete implementations to realize the components of CVAE represented in Eq. (5.1) are as follows. The encoder representing the approximate posterior $q(\boldsymbol{z}|\boldsymbol{x})$ can be implemented with bi-directional LSTM [148] :

$$[\overrightarrow{\boldsymbol{h}_m^{(E)}}, \overleftarrow{\boldsymbol{h}_m^{(E)}}] = \mathrm{BiLSTM}\left(\overrightarrow{\boldsymbol{h}_{m-1}^{(E)}}, \overleftarrow{\boldsymbol{h}_{m+1}^{(E)}}, \mathrm{Embed}(\boldsymbol{x}_m)\right), \tag{5.2}$$

where $\overrightarrow{\boldsymbol{h}_m^{(E)}}$ and $\overleftarrow{\boldsymbol{h}_m^{(E)}}$ are the hidden outputs of forward and backward LSTM at the m-th input token, respectively. Embed() is a function for obtaining embedding vectors for tokens. The parameters of the approximate posterior can be obtained from the final time step of the encoder output:

$$\boldsymbol{z} \sim q(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{z} \mid \boldsymbol{\mu}([\overrightarrow{\boldsymbol{h}_M^{(E)}}, \overleftarrow{\boldsymbol{h}_1^{(E)}}]), \boldsymbol{\sigma}([\overrightarrow{\boldsymbol{h}_M^{(E)}}, \overleftarrow{\boldsymbol{h}_1^{(E)}}])\right), \tag{5.3}$$

where $\boldsymbol{\mu}()$ is the linear transformation function for obtaining the mean parameter and $\boldsymbol{\sigma}()$ is the function as $\boldsymbol{\sigma}(\boldsymbol{a}) = \exp\left(0.5 * \text{Linear}(\boldsymbol{a})\right)$ for obtaining the standard deviation parameter that has the diagonal component only, respectively. The content latent variable $\boldsymbol{z}$ can be sampled using the obtained parameters. To model the conditional distribution of the decoder, content, and style variables are treated as the initial hidden inputs of the decoder by applying linear transformation as follows:

$$\boldsymbol{h}_0^{(D)} = \text{Linear}([\boldsymbol{z}, y]). \tag{5.4}$$

The output conditional distribution $p(\hat{\boldsymbol{x}}_n|\hat{\boldsymbol{x}}_{n-1}, \boldsymbol{z}, y)$ at time n can be implemented with the LSTM decoder:

$$p(\hat{\boldsymbol{x}}_n|\hat{\boldsymbol{x}}_{n-1}, \boldsymbol{z}, y) = \text{Softmax}(\text{LSTM}(\boldsymbol{h}_{n-1}^{(D)}, \text{Embed}(\hat{\boldsymbol{x}}_{n-1})), \tag{5.5}$$

where Softmax() is the softmax function with linear transformation into output dimensions.

5.3 CVAE with Content Word Storage

The latent representation capacity in CVAE is limited. In normal CVAE, all words are embedded in a fixed-dimension latent representation. They include words that should be retained before and after the transfer. However, it is practically hard to achieve it by using only the limited capacity of the latent representation.

To exclude retained word information from the latent representation to effectively use the limited capacity, we propose CWS. This method explicitly defines the information to be retained as "content words" and handles them separately from the latent content feature $\boldsymbol{z}$. CWS is a method used for extracting information to be retained from the text input and transmitting it directly to the decoder. In this paper, we emphasize the

point that content words "have substantive meaning and can be independent elements". Therefore, we define four parts of speech as content words: nouns, verbs, adjectives, and adverbs.

We propose two types of CWS to extend CVAE. The first is CWS based on a bag-of-words (BoW) and the second is CWS based on the attention mechanism [109].

5.3.1 BoW as CWS

BoW is one of the classical methods of representing text features. It is created by adding together one-hot vectors that display the occurrence of each word in the text. The dimension of the BoW is equal to the vocabulary size.

A schematic of using BoW as CWS is shown in the lower part of Figure 5.1. To utilize BoW as CWS, we build a lexicon of content words from training data to define the dimension of BoW. Thus, our BoW has a dimension equal to the vocabulary size of training data. We refer to this method as CWS-BoW.

CWS-BoW can be incorporated into CVAE as follows. Eq. (5.6) is calculated to obtain the CWS feature $\boldsymbol{z}^{\mathrm{CWS}}$:

$$\boldsymbol{z}^{\mathrm{CWS}} = \mathrm{Linear}(\mathrm{BoW}(\boldsymbol{x}^{\mathrm{CW}})). \tag{5.6}$$

In CWS-BoW, only the content words are first extracted from the input word sequence to obtain the content word sequence $\boldsymbol{x}^{\mathrm{CW}}$. This sequence is input into the BoW function that transforms it to the BoW feature represented by the sum of one-hot vectors. We obtain the CWS feature $\boldsymbol{z}^{\mathrm{CWS}}$ by the linear transformation of the BoW feature and the CWS feature is then fed into the decoder along with the content latent variable $\boldsymbol{z}$.

5.3.2 Attention mechanism as CWS

The attention mechanism [109] is employed within the sequence-to-sequence conversion framework to align input and output sequences. It allows access to the entire input sequence to derive the hidden representation at each decoding time step. This contrasts with the original sequence-to-sequence framework [64], which utilizes a fixed encoded representation to decode the entire sequence.

We propose a CWS method using attention. We expect that the hidden representation in attention can function as CWS, assuming that attention selectively focuses on content words in input texts. We refer to this method as CWS-Attn.

A schematic of using attention as CWS is shown in Figure 5.1. The general workflow is as follows. Content words are extracted from the input based on the lexicon built for BoW. The content words are then converted into a content feature vector by attention at each decoding step. We use the context vector, which is the weighted sum of input sequences in the attention framework, as the content feature of CWS-Attn. Finally, the content feature vector is input into the decoder to produce the output texts.

Attention can be incorporated into CVAE as follows. CWS-Attn extracts the content words from the input word sequence to obtain the content word sequence $\boldsymbol{x}^{\mathrm{CW}}$. We embed this sequence as

$$\boldsymbol{e}^{\mathrm{CW}} = \mathrm{Embed}(\boldsymbol{x}^{\mathrm{CW}}) \tag{5.7}$$

and make this the subject of attention from the decoder. Eq. (5.8) is calculated to obtain the content feature based on CWS-Attn:

$$\boldsymbol{z}_n^{\mathrm{CWS}} = \boldsymbol{e}^{\mathrm{CW}} \cdot \mathrm{Softmax}(\boldsymbol{e}^{\mathrm{CW}} \odot \boldsymbol{h}_n^{(D)}). \tag{5.8}$$

The context vector $\boldsymbol{z}_n^{\mathrm{CWS}}$ at time step n can be obtained by multiplying context word vectors and attention weights $\mathrm{Softmax}(\boldsymbol{e}^{\mathrm{CW}} \odot \boldsymbol{h}_n^{(D)})$; the attention weights are calcu-

lated by taking the inner product of the embedded vector and the hidden layer of the decoder; The content feature is fed into the decoder at each time step.

The attention mechanism has advantages over BoW in that it can consider contexts in a sentence and its capacity is proportional to the output sequence length. Note that CWS-BoW uses fixed representation $\boldsymbol{z}^{\mathrm{CWS}}$ at all decoding time steps, whereas CWS-Attn uses a different representation $\boldsymbol{z}_n^{\mathrm{CWS}}$ at each time step.

5.3.3 Positional embedding in CWS-Attn

CWS-Attn can further be improved by using the position information of content words. CWS-Attn has no information about the order of input words. Therefore, when input text has more than one identical word, the word embeddings would be the same. This may generate problematic transferred texts ignoring the number and order of content words.

Therefore, we propose adding the positional embedding that explicitly indicates the word position to the embedding of the content word in the input text. It is expected that this will enable CWS-Attn to consider even the number and order of content words. In this paper, we use the absolute positional embedding (APE) with the trigonometric function used in Transformer [130]. Specifically, the position embedding vector $p_t^{(i)}$ expressed as

$$p_t^{(i)} = \begin{cases} \sin\left(t/10000^{i/d}\right) & \text{if } i = 2k \\ \cos\left(t/10000^{i/d}\right) & \text{if } i = 2k+1 \end{cases} \tag{5.9}$$

is added to a word vector series that masks all but the content words in the input text, where t is the word position and d is the number of embedding dimensions. We refer to this method as CWS-Attn + APE.

5.3.4 Cyclic learning

The CWSs are expected not to improve style control performance because they are mainly aimed at improving content preservation performance. The introduction of CWS may rather cause the degradation of style control performance, and the degradation may not be negligible for methods using reconstruction-based training without parallel data such as CVAE. Therefore, we introduce cyclic learning inspired by CycleVAE [149], a method used to improve transformation performance by cycle reconstruction from the pseudo-reference text to the original text, in addition to the original reconstruction. We refer to the method introducing cyclic learning as CycleCVAE + CWS-Attn + APE [1].

A schematic of CycleCVAE + CWS-Attn is shown in Figure 5.2. CycleCVAE + CWS-Attn has two steps. During the pre-inference step, the input text $\boldsymbol{x}$ with the source style s is converted into the text x' with the target style t by using the pretrained CVAE + CWS-Attn + APE. This text x' is treated as a pseudo-reference text. During the training step, the cycle reconstruction of x' is converted back to x with the source style. At the same time, the reconstruction of x is learned using cross-entropy loss. This is expected to improve the performance of TST while maintaining the condition of not using parallel data.

Simply combining the cyclic learning with APE is suspected not to work properly, because the input and output lengths are different in cycle reconstruction.

We, therefore, propose "content word absolute positional embedding (CWAPE)" to circumvent this issue. CWAPE extracts content words from the input text and embeds their positions to represent their pseudo-relative positions in the entire text, as shown

[1]Note that we call this model CycleCVAE for convenience only, and such a model differs from "CycleVAE" [149].

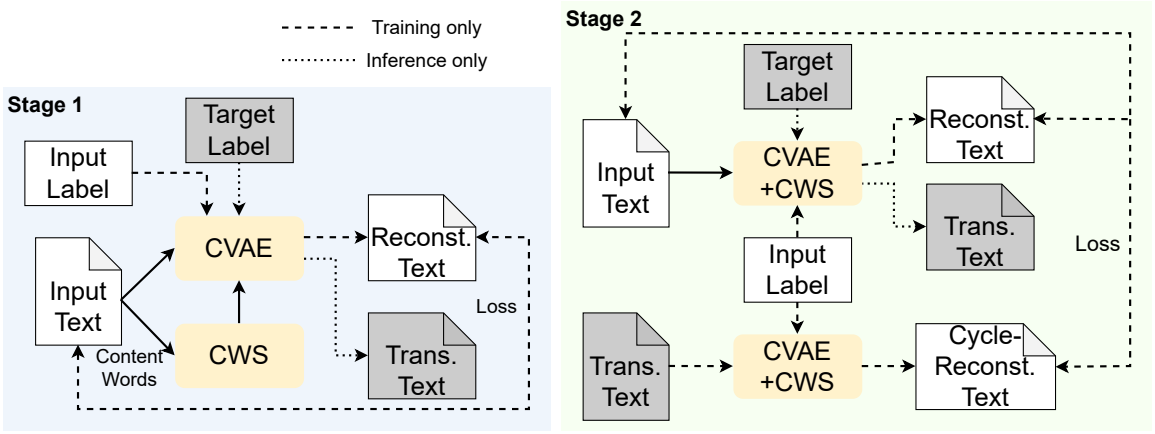

Figure 5.2: *Proposed method: CycleCVAE+CWS.*

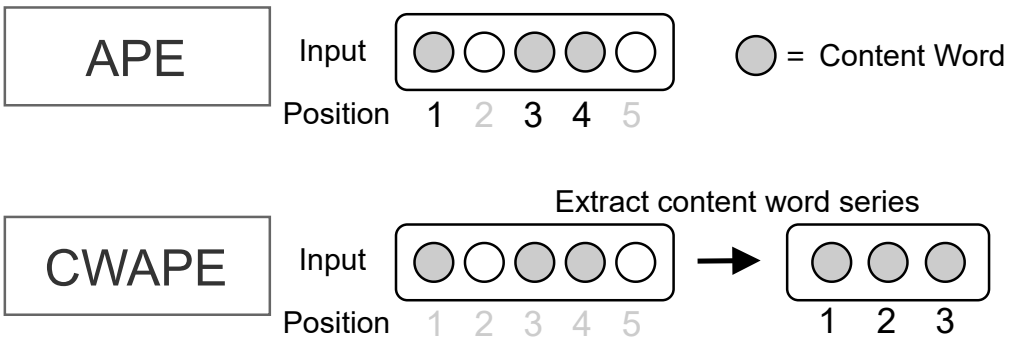

Figure 5.3: *Comparison between APE and CWAPE.*

in Figure 5.3. Because we assume in this paper that content words do not change before and after style transfer, CWAPE can effectively preserve the position of content words in cyclic learning.

5.4 Experimental Evaluation

To evaluate the effect of CWS and cyclic learning on content preservation and style control performances, we conducted two experiments on spoken style transfer targeting the following styles: (1) disfluency (fluent and disfluent) and (2) dialect (standard and

Kansai Japanese).

5.4.1 Datasets

For the disfluency transfer experiment, we used the corpus of the spontaneous Japanese (CSJ) dataset [49], which is a large-scale lecture speech corpus. We obtained transcriptions with and without disfluency, such as FPs and stutter words, based on the disfluency annotation in the CSJ. The transcription without disfluency was derived by removing words labeled disfluency. The total number of transcripts obtained was 375k sentences. We split the transcripts into 352,874, 15,016, and 7,508 sentences to construct training, validation, and test sets, respectively, at a ratio of 94:4:2. The vocabulary size of the training set was 34,293. We also defined nouns, verbs, adjectives, and adverbs as content words to build a lexicon of content words by using morphological analysis results of Mecab [150]. The vocabulary size of the lexicons was 31,210.

For the dialect transfer experiment, we used the corpus of Kansai Vernacular Japanese (KVJ) [151] in addition to the CSJ. The KVJ is a collection of sociolinguistic interviews of university students with family members who were born and raised around Osaka and those who moved to Osaka during adulthood. The Kansai region with Osaka as its largest city has a dialect distinct from the Tokyo dialect (standard Japanese). Note that the KVJ includes some texts that are in standard Japanese when viewed only in letters, e.g., “この結果から分かりますように (k o n o k e cl k a k a r a w a k a r i m a s U y o o n i)”. This text appears to be standard and formal Japanese in letters, but the accent is different when spoken in the Kansai dialect. This issue is discussed in Section 5.5.2. We used the KVJ as Kansai dialect data and the CSJ without disfluency as standard dialect data. The transcripts of the CSJ portion comprised 187k sentences and those of the KVJ portion comprised 153k sentences, with a total of 340k

sentences. Note that some texts in the KVJ contained disfluency words such as FPs, but their occurrence was negligible. We split the transcripts into 318,354, 15,138, and 6,806 sentences to construct training, validation, and test sets, respectively, at a ratio of 93:5:2. The vocabulary size of the training set was 42,527 and the lexicon of content words was 41,496.

5.4.2 Systems

We constructed two CVAE systems using CWS: CVAE + CWS-BoW and CVAE + CWS-Attn. In CWS-Attn, we constructed three additional variations: CVAE + CWS-Attn + APE, CycleCVAE + CWS-Attn + APE, and CycleCVAE + CWS-Attn + CWAPE. We also constructed a plain CVAE system without CWS as a baseline.

The parameters common to all systems were as follows: Max epoch of the training step of 100 and the learning rates of the encoder and decoder of 0.001 and 1.0, the input and hidden layer dimensions of the encoder of 256 and 1024, the input and hidden layer dimensions of the decoder of 128 and 1024, the number of the hidden layer of the encoder and decoder of 2 and 1, the dimensions of the latent representation $\boldsymbol{z}$ and the embedding vector of the style label y of 64 and 16, respectively.

We used a prompt-based system with LLMs (GPT-3.5 and GPT-4) as the latest comparison method. The checkpoint of GPT-3.5 is "gpt-3.5-turbo-0125", and that of GPT-4 is "gpt-4-turbo-2024-04-09". We used three types of prompts: zero-shot (zs), which only gave transfer instructions; nonparallel few-shot (fs-np), which gave some examples of text in each style along with instructions; and parallel few-shot (fs-p), which gave some examples of TST in each style along with instructions. Each prompt in disfluency transfer is shown in Table 5.2. Note that the explanation of disfluency at the beginning and the notes on transfer (When doing so, ...) are omitted in the

few-shot prompts because they are redundant. In the LLM-based model, 250 texts of each style were randomly selected from the test data. As a result, a total of 500 texts were transferred.

5.4.3 Objective evaluation

For objective evaluation, we calculated the automatic scores using three metrics: accuracy (AC), BLEU [104], cosine similarity (CS), and content word error rate (CWER). In addition, the execution time required to convert 500 samples was also measured to evaluate practical applicability to TTS.

AC

AC was used to evaluate the performance of spoken style control. To calculate AC, we used a pretrained style-classifier-based TextCNN [152], which had 97% AC on the test set of disfluency transfer and 96% AC on the test set of dialect transfer. We also computed AC excluding content words (AC w/o CW) with the classifier to measure the AC of purer-style transfer. We considered that this metric was preferable to AC particularly for dialect transfer, because the classifier could classify styles depending on particular words that did not represent a style but merely appeared in only one of the corpora when the transcripts of the two styles originated from different corpora, which resulted in low accuracy. The pretrained style classifier for AC w/o CW had 94% AC on both test sets of disfluency transfer and dialect transfer.

Table 5.2: *Prompts used in LLM's TST (original in Japanese, shown here in English translation).*

method	prompt
zero-shot (zs)	Disfluency is a disturbance in speech such as "FP" or "self-correction" that results from human hesitation or error. For the following tokenized disfluent (fluent) spoken text, please transfer the style to fluent (disfluent) spoken text. When doing so, preserve the meaning and expression of the original text as much as possible, and do not add extra expressions or change the original expression. Input: [input_text] Transferred:
nonparallel few-shot (fs-np)	Disfluency is ... [the same description as in zs] ... hesitation or error. Below is data in the form of "[Style Labels] [Tokenized Text]," with labels of 0 indicating disfluent style and 1 indicating fluent style. Please transfer the text to the appropriate style for those with labels 0 to 1, and 1 to 0. When doing so, ... [the same description as in zs] ... expression. Here are some examples of text for each style; 0 I actually became a resident of Akabane. 0 I produced the total number of moras for the lecture hours. 1 This is an in, inter, uh, internal breakdown. 1 Um, well, I'm going to take it out, but I'm also going to do the same process from Japanese to English... Please output the transferred result of the following input. Input: [input_text] Transferred:
parallel few-shot (fs-p)	Disfluency is ... [the same description as in zs] ... hesitation or error. Below is a tokenized disfluent (fluent) spoken style text. Please transfer the input to a fluent (disfluent) spoken style text. When doing so, ... [the same description as in zs] ... expression. Here are examples of transfer; Input: I actually become a resident of Akabane Output: Um, I actually become a resident of, like, Akahane Input: I produced the total number of mora for the lecture hours. Output: Er, also I produced the total number of mora for the lecture hours. Please output the transferred result of the following input. Input: [input_text] Transferred:

BLEU

BLEU [104] is a word-overlap-based measure of the content preservation of the entire text proposed in the field of machine translation. BLEU was also used to evaluate content preservation in TST [111, 119, 146]. We used the following two BLEU metrics: reference-BLEU (r-BLEU), calculated by comparison with the reference, and self-BLEU (s-BLEU), calculated by comparison with an input. For dialect transfer, we evaluated s-BLEU owing to a lack of reference transcripts.

Cosine similarity (CS)

CS [153] calculates the cosine similarity between the original and transferred sentence embeddings. In the original paper, sentence embedding consists of max, min, and mean pooling of pretrained word embedding. We used the Japanese implementation[2] of Sentence-BERT [105] as sentence embedding. This is expected to enable the measurement of semantic similarity rather than lexical similarity.

Content word error rate (CWER)

CWER is the word error rate of only a content word sequence for evaluating the content word preservation. In TST, it is desirable that the content words of the input and generated texts remain unchanged. Therefore, CWER is used to calculate WER for the "content word series of the generated text" and the "content word series of the input text."

We calculated the average values of automatic scores for all test samples. We also measured the scores for reference transcripts as well when reference transcripts were

[2]https://huggingface.co/sonoisa/sentence-bert-base-ja-mean-tokens-v2

available.

5.4.4 Subjective evaluation

For subjective evaluation, we conducted two types of web-based human evaluation tests. One is evaluated only with text, while the other is evaluated with synthesized speech with style-transferred text.

MOS test for Text style transfer

We recruited fourteen participants who were fluent Japanese speakers including native speakers in the disfluency transfer experiment. We included CVAE + CWS-Attn + APE and CycleCVAE + CWS-Attn + CWAPE as the proposed systems and CVAE as the baseline in the human evaluation test. We used 50 samples per transfer direction, 100 samples in all, for the evaluation. Each text sample was evaluated three times. We obtained 900 evaluations in total from the test. We also recruited 150 crowd workers who were Japanese speakers living in or from Kansai in the dialect transfer experiment. We used 200 samples per transfer direction, 400 samples in all, for the evaluation. Each text sample was evaluated four times. We obtained 4800 evaluations in total from the test. We prepared the Kansai dialect data for evaluation by selecting 200 samples that had a clear dialect easily judged only by texts. We determined the statistical significance using the Mann–Whitney U test in both disfluency and dialect transfer experiments.

The subjects were asked to evaluate the samples in three aspects: the degree of style transfer (ST), content preservation (CP), and naturalness (Nat) as in related studies [113, 116, 117, 154]. ST was rated on a scale of one to four. CP and Nat were

rated on a scale of one to five.

ABX-test for Text style transfer + Spoken TTS

To confirm TST effectiveness for TTS, we performed a human listening test in the disfluency TST + TTS. We recruited 27 participants who were native Japanese speakers. We prepared synthetic speech from original texts without disfluency and denoted as "`Ori`". We generated style-transferred speech by TTS after TST with the proposed system CycleCVAE + CWS-Attn + CWAPE and denoted as "`Prop`". The TTS system used here was a lecture-style spoken TTS trained at 1000 epochs with CSJ speech data using VITS [70] implemented in Espnet2 citeespnet2. In the listening test, evaluators are presented with natural speech with original text as a reference. After listening to the reference speech, the subjects evaluated synthetic speech from `Ori` and `Prop` by listening to them in random order. We selected 7 different speakers from the CSJ, took their respective continuous transcribed text samples, and synthesized speech to produce a 15–20 second speech sample `Ori` and a speech sample `Prop` with TST. We also selected reference speech samples of 10–15 seconds each. In this experiment, 189 evaluations were obtained.

The subjects were asked to evaluate "Which speech reproduces the style of the reference speech?" and "Which speech is more natural regardless of the reference speech?" The subjects selected from four options, "`Ori` (`Prop`) is better" and "`Ori` (`Prop`) is relatively better". The result of this evaluation is evaluated as a preference score: 0 is uniformly assigned when `Ori` is rated as good and 1 when `Prop` is rated as good, and the average of the two is calculated to indicate the percentage of "`Prop` rated as better than `Ori`". We determined the statistical significance using a binomial test.

Table 5.3: *Automatic evaluation results for disfluency.*

Method	AC ↑	r-BLEU ↑	CS ↑	CWER ↓
Oracle	98.35	100.00	94.69	0.00
CVAE	61.83	31.79	56.06	60.84
+ CWS-BoW	54.83	35.33	70.75	51.85
+ CWS-Attn	63.99	47.66	85.45	29.35
+ CWS-Attn + APE	67.54	**60.04**	**88.53**	**7.60**
CycleCVAE + CWS-Attn + APE	77.12	53.28	87.64	9.96
CycleCVAE + CWS-Attn + CWAPE	**80.51**	56.43	88.00	9.99
GPT-3.5 -zs	48.40	20.47	81.14	81.33
GPT-3.5 -fs-np	73.40	36.71	86.73	45.30
GPT-4 -zs	94.80	43.42	88.90	22.94
GPT-4 -fs-np	**95.40**	**61.57**	**92.65**	**12.33**
GPT-3.5 -fs-p	69.00	58.74	**92.21**	23.70
GPT-4 -fs-p	**97.80**	**59.54**	92.03	**10.75**

5.4.5 Results

Objective evaluation results

The results of the objective evaluation using the automatic metrics in disfluency transfer are shown on the left side of Tables 5.3 and 5.3 . In terms of content preservation, the proposed methods showed clear improvements in BLEU, CS, and CWER compared with the normal CVAE. This suggests that the introduction of CWS contributed to the preservation of content words. CWS-Attn had consistently better BLEU (47.66), CS (85.45), and CWER (29.35) than CWS-BoW (35.33, 70.75, 51.85). This indicates that CWS-Attn was a more powerful content preservation method than CWS-BoW. The model combining CWS-Attn with APE was found to be the best

Table 5.4: *Automatic evaluation results for dialect.*

Method	AC ↑	AC w/o CW ↑	s-BLEU ↑	CS ↑	CWER ↓
Oracle	–	–	–	–	–
CVAE	58.04	57.49	27.37	52.27	61.85
+ CWS-BoW	34.31	57.24	36.85	71.67	43.36
+ CWS-Attn	35.20	55.41	48.41	84.88	24.57
+ CWS-Attn + APE	41.92	63.86	**60.02**	**87.15**	**9.23**
CycleCVAE + CWS-Attn + APE	**61.71**	**73.98**	44.65	83.58	12.59
CycleCVAE + CWS-Attn + CWAPE	60.62	73.25	47.04	84.56	10.75
GPT-3.5 -zs	**59.60**	**62.73**	23.66	80.11	63.17
GPT-3.5 -fs-np	52.00	52.31	31.99	83.62	47.77
GPT-4 -zs	48.60	49.50	33.49	85.44	44.64
GPT-4 -fs-np	55.00	57.46	**38.24**	**86.71**	**39.54**
GPT-3.5 -fs-p	–	–	–	–	–
GPT-4 -fs-p	–	–	–	–	–

and the improvement in CWER (7.60) was particularly marked. We also found that CWAPE was more effective than APE in CycleCVAE due to superior BLEU, CS, and comparable CWER.

The improvement in CWER is also seen in Figure 5.4, which shows heat maps of the attention weights of the attention mechanisms. The vertical axis of the heat maps is the generated text and the horizontal axis is the input text. The heat maps should be nearly diagonal if the position and number of content words are correctly retained. In CVAE + CWS-Attn, the attention weight is not diagonal and some content words are generated multiple times. On the other hand, in CVAE + CWS-Attn + APE, the attention weight is perfectly diagonal.

In terms of style control performance, both CVAE + CWS-BoW and CVAE + CWS-Attn showed comparable ACs (54.83 and 63.99) to the normal CVAE (61.83), indicating that the introduction of CWS did not degrade the style control performance. The improvement in AC due to the introduction of positional embedding was also seen (67.54). It was because the positional embedding enables the CWS to store most of the information in the content word by itself, and this enables the encoder to acquire a latent representation that contains more information other than the content word. In addition, the introduction of cyclic learning was confirmed to significantly improve AC (80.51).

The bottom of Table 5.3 shows the objective evaluation of LLM-based methods. GPT-3.5 performed poorly in all metrics except CS. In contrast, GPT-4 obtained a very high AC under all conditions, indicating that it had a high style control performance. However, content preservation performance from GPT-4 was not prominent. Under zero-shot conditions, BLEU and CWER of GPT-4 were particularly poor. This is because even with the "preserve content" constraint in the prompt, the LLM tended to ignore the original text and generate text that was too long or rephrased, especially in the direction of fluent to disfluent. The sentences generated by GPT-4 with few-shot looked like they were very fluent and natural. However, the "spoken style" targeted in this study should ideally be able to transfer not only FPs but also stutter word, such as misspoken and self-correction. In this respect, we also found that "natural human error" is difficult to reproduce in LLMs that are trained only with fluent written language. Comparing the proposed method with the LLM-based method under non-parallel conditions, we first confirmed that the proposed method outperforms all items for systems using GPT-3.5. On the other hand, the proposed method outperforms the system using GPT-4 in CWER and is comparable in BLEU and CS, indicating that the

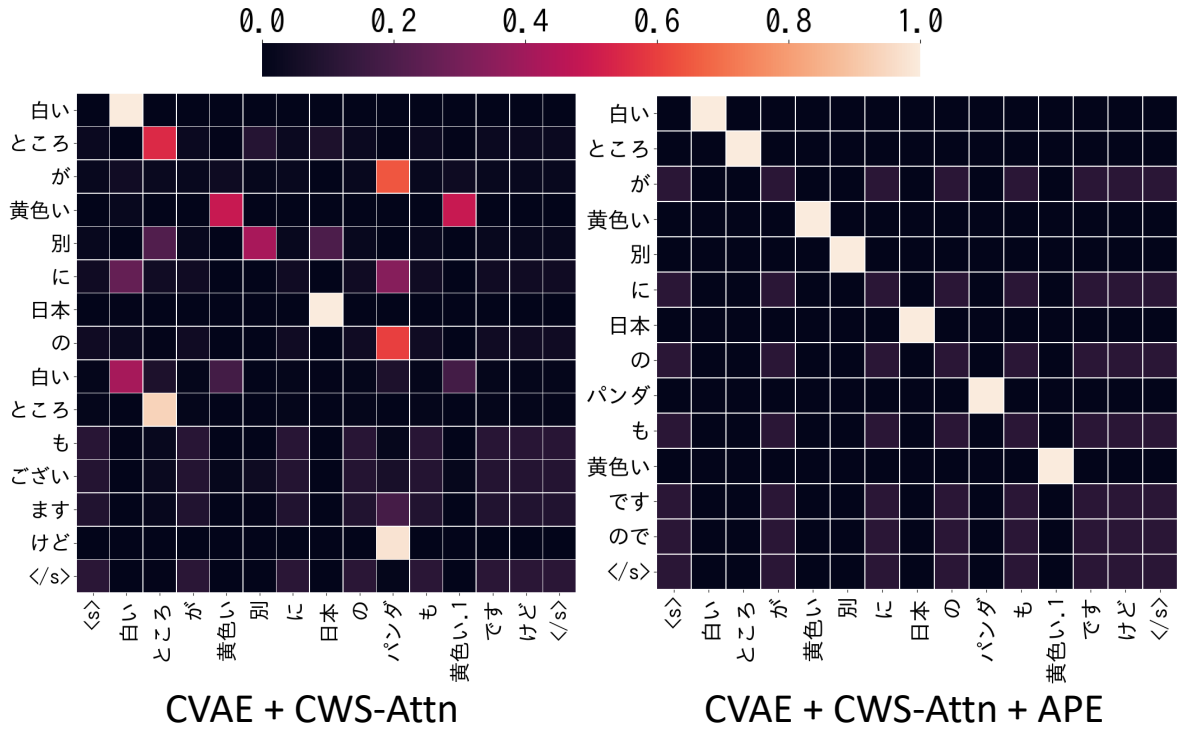

Figure 5.4: *Comparison of attention weight heatmaps. The vertical axis is the generated text and the horizontal axis is the input text.*

proposed method has an advantage in content preservation performance. The differences in performance between the LLM-based system for each condition were observed between zero-shot and few-shot.

The results of the objective evaluation of dialect transfer are shown in Table 5.4. The proposed methods outperformed the baseline in terms of content preservation metrics such as BLEU and CWER in this task as well. CWS-Attn had better BLEU (48.41), CS (84.88), and CWER (24.57) than CWS-BoW (36.85, 71.67, 43.36), and the model combining CWS-Attn with APE was found to be the best BLEU (60.02), CS (87.15), and CWER (9.23) in this task as well. In terms of style control performance, the proposed CVAE + CWS systems showed lower ACs than the baseline. We suspected

that this was caused by the problem inherent to AC, as described in Section 5.4.3. The CWS-BoW and CWS-Attn, however, had comparable ACs w/o CW (57.24 and 55.41) to the baseline (57.49). We, therefore, concluded that the proposed systems did not degrade the style control performance in this task as well. In addition, the introduction of cyclic learning with CWAPE was confirmed to improve AC w/o CW (73.25) as in the disfluency transfer. However, as the changes in letters are larger in dialect transfer than in disfluency, successful style control through cyclic learning could have resulted in lower BLEUs (47.04). On the other hand, the results showed that the value of CS, a content retention measure that is less affected by lexical change, was not compromised.

Tables 5.5 and 5.6 show the objective evaluation results arranged by transfer direction. Henceforth, CWS-Attn is denoted as CWS. In the disfluency transfer, a huge difference in AC between the transfer directions was observed: the direction from fluent to disfluent showed markedly lower ACs than the other direction. We inferred that the transfer in the disfluent to fluent direction was easier than in the opposite direction because the former simply required the deletion of a disfluent phrase that exists in the input text. In contrast, the transfer in the fluent to disfluent direction required a decision on what and where to insert disfluent words in the original text, which was a nondeterministic task in the nature of disfluency. In the dialect transfer task, the difference in AC was also as marked as in the disfluency task. In particular, the AC w/o CW in the direction from Kansai to standard Japanese was found to be higher. This is partly due to the fact that the Kansai dialect data contain texts that are also present in standard Japanese when viewed only as texts, and thus there are output texts that are in standard Japanese without transfer.

The bottom of Table 5.4 shows objective evaluation results from LLM-based methods. LLMs showed moderately high AC, but they performed very poorly in BLEU

Table 5.5: *Performance comparison between different style transfer directions in disfluency. (CWS-Attn is denoted as CWS in this table.)*

Method	**direction**	**AC ↑**	**r-BLEU ↑**
CVAE	fluent → disfluent	34.20	43.35
	disfluent → fluent	89.45	26.34
CVAE+CWS+APE	fluent → disfluent	54.40	73.07
	disfluent → fluent	81.67	55.74
CycleCVAE+CWS+CWAPE	fluent → disfluent	68.49	65.21
	disfluent → fluent	93.93	51.90

and CWER. It was partially caused by the cases where the LLM simply changed the original text too much. A more significant factor was that it could also convert dialects of the content words. Looking at the actual generated text, there were many cases where noun words in the content words were converted to dialect words with the same meaning, such as "母 (mother)" to "おかん (mom)". This means that the content preservation of the conversion results of LLM-based models cannot be fairly evaluated by BLEU or CWER. On the other hand, CS is capable of robust evaluation against lexical changes, but the accuracy depends on the domain of the training data for the model that produces the sentence embedding. Although the training data for Sentence-BERT used in this work is closed to the public, it is presumed to have been trained on written texts such as Wikipedia, so its performance for spoken text and dialects may not be sufficient. Therefore, a human evaluation should finally be conducted. Comparing the proposed method with the LLM-based method, our proposed methods showed higher or comparable AC and higher BLEU, CS, and CWER than that of LLMs. Despite the better objective metrics from our methods, the flexible

Table 5.6: *Performance comparison between different style transfer directions in dialect. (CWS-Attn is denoted as CWS in this table.)*

Method	direction	AC ↑	AC w/o CW ↑	s-BLEU ↑
CVAE	Standard → Kansai	56.31	26.34	27.78
	Kansai → Standard	60.16	73.00	26.62
CVAE+CWS+APE	Standard → Kansai	31.75	48.77	65.45
	Kansai → Standard	54.42	78.01	51.95
CycleCVAE+CWS+CWAPE	Standard → Kansai	62.60	67.22	50.73
	Kansai → Standard	58.19	80.68	41.48

Table 5.7: *Comparison of execution times, averaged over 500 sample runs.*

Method	Ours(CPU)	Ours(GPU)	GPT-3.5	GPT-4
Time (ms / sample)	172.3	12.7	824.6	1346.0

conversion observed in LLMs was lacking in our methods. This was one of the major differences in our approach from LLM.

Finally, a comparison of execution times is shown in Table 5.7. On both CPU and GPU, we found that our model was significantly faster than GPTs. The GPT-4, in particular, had a very long execution time in exchange for its high performance.

Subjective evaluation results

The human evaluation results of disfluency-transferred text are shown in the upper half of Table 5.8. CVAE + CWS + APE improved CP significantly. ST and Nat of CWS were comparable to the baseline. This indicates that CWS improved content

Table 5.8: *Human evaluation results of TST. Shown with 95% confidence interval. (CWS-Attn is denoted as CWS in this table.)*

Method	**ST ↑**	**CP ↑**	**Nat ↑**
	Disfluency		
CVAE	2.40 ± 0.15	2.46 ± 0.18	4.01 ± 0.24
CVAE+CWS+APE	2.36 ± 0.12	**3.84** ± 0.21	3.82 ± 0.19
CycleCVAE+CWS+CWAPE	**2.91** ± 0.18	3.67 ± 0.19	4.02 ± 0.18
	Dialect		
CVAE	1.72 ± 0.04	1.89 ± 0.05	2.60 ± 0.06
CVAE+CWS+APE	2.02 ± 0.04	**2.98** ± 0.06	2.91 ± 0.06
CycleCVAE+CWS+CWAPE	**2.25** ± 0.04	2.94 ± 0.06	3.09 ± 0.05

preservation without compromising style control or naturalness performance. In addition, the proposed model with cyclic learning significantly improved ST without the degradation of CP and Nat. The results of the subjective human evaluation of dialect-transferred text are shown in the bottom half of Table 5.8. This experiment shows the same trend as in the disfluency transfer experiment with lower values overall.

The results of the ABX-test of disfluency transferred TTS are shown in Table 5.9. The TTS combined with the proposed method (`Prop`) achieved a preference score of 65% for style reproducibility, compared to the case of performing a plain TTS (`Ori`). From the binomial test, the score for style reproducibility was not 50 at the more than 99% significance level, meaning that there was a significant difference. Furthermore, no significant difference was found for naturalness, and the evaluation value was close to 50. This indicates that the proposed method can reproduce the style without sacrificing

Table 5.9: *Preference scores of* `Prop` *for* `Ori` *in ABX test. Shown with 95% confidence interval.*

Metrics	Preference Score
Reproduction of Style	$65.08^{+6.78}_{-7.26}$
Naturalness	$52.38^{+7.30}_{-7.37}$

naturalness by using the proposed method as a preprocessing for TTS.

Summary of results

We summarize our experimental results in Figure 5.5. Hereafter, CycleCVAE + CWS is denoted as Cycle. While the normal CVAE had some style control performance, its performance in content preservation was notabley weak. By introducing CWS and positional embedding in the proposed methods, the content preservation performance was enhanced. On the other hand, Cycle has lower content preservation than CVAE + CWS + APE. Still, evaluation indicators of style control and word-overlap-based content preservation are trade-offs. If the proportion of style-transferred text is high, the evaluation value of BLEU and CWER will decrease. Nevertheless, the difference is much smaller than between the CVAE and CVAE + CWS + APE. Furthermore, the scatter plot of automatic evaluations in disfluency transfer shows that Cycle + CWAPE is the closest in direction (cosine similarity) and Euclidean distance to the ideal value, Oracle. There is no Oracle in the dialect transfer, but if we assume that the upper right end is Oracle as same as disfluency transfer, Cycle + CWAPE is the closest to the upper right. In addition, the same trend was observed in the human evaluations, and only significant differences were observed in ST and not in CP. From

this, we can conclude that Cycle + CWAPE improves content preservation and style control in a balanced manner.

It was also shown that the overall performance of the dialect transfer was lower than that of the disfluency transfer. This is partially due to the fact that the standard Japanese data used in the dialect conversion are "lecture speech" data and the Kansai dialect data are "daily conversation" data, which have different styles other than dialect, making the present dialect transfer difficult, and there is room for improvement in this respect as well.

Tables 5.10, 5.11, 5.12, and 5.13 shows transferred examples in fluent-to-disfluent, disfluent-to-fluent, standard-to-Kansai, and Kansai-to-standard direction, respectively. For the English translation of dialect transfer, the Scottish dialect is used as an example to transfer the nuance of the Kansai dialect. The summarized outcomes are evident in the transferred examples presented in these tables. The transferred examples from GPT showed a tendency for the transferred sentences to be completed as sentences even if the original sentence was ended in the middle. This may be due to the fact that LLM is basically learned in a completed written language, while transcription of spoken language often consists of a series of incomplete sentences with no punctuation.

5.5 Conclusions and Discussions

5.5.1 Summary

In this chapter, we proposed a method for improving content preservation in supervised nonparallel text style transfer (TST), addressing the second challenge presented in Section 1.2.2.

Section 5.2 introduced the *conditinal variational autoencoder* (CVAE)–based TTS

model as baseline. Section 5.3 extended CVAE with *content word storage* (CWS) to directly transmit content word information to the decoder. In addition, we introduced cyclic learning to improve the style control performance without parallel data.

Section 5.4 evaluated these methods on two spoken styles—disfluency and dialect. In both experiments, CWS improved content preservation, and CWS with an attention mechanism was more effective than that with bag-of-words (BoW). Furthermore, cyclic learning improved style control performance while minimizing degradation in content preservation.

Our objective evaluation also shows that the proposed method is comparable with LLM-based systems in terms of content preservation, yet much more efficient in terms of computational latency. In addition, using the proposed method as a preprocessing for TTS improved the style reproducibility of speech compared to the case where the proposed method is not used. Taken together, these two results suggest that our method could be useful as a pretreatment method for TTS.

5.5.2 Discussions

We discuss some limitations of the proposed method and directions for future research.

How to determine which words are content words?

The proposed method is very effective when it is clear which words in the text are content words. It means how to define content words is important. In this paper, content words are specified by parts of speech: nouns, verbs, adjectives, and adverbs are designated as content words based on morphological analysis. This is based on the

assumption that these four parts of speech would not be involved in the speech style and persona characteristics of target utterances. In practice, however, nouns, verbs, and adjectives may change in the case of dialect style transfer, for example. In the case of sentiment transfer, which is the benchmark for the TST of English datasets, verbs, adjectives, and adverbs are the main targets of conversion. To achieve a more flexible style transfer, we need context-dependent definitions of content words that do not solely depend on parts of speech. It would be interesting to investigate if LLMs could solve the context-dependent content word judgment.

Limitations of styles that can be handled

In terms of the "standard/Kansai Japanese dialect datasets" used in this paper, the corpus of the spontaneous Japanese (CSJ) is the lecture speech dataset and the corpus of Kansai Vernacular Japanese (KVJ) is the daily conversation dataset. In other words, these data include the "dialect" style as well as the "lecture and conversation" style. To deal with such cases, we believe that it is necessary to incorporate methods that treat style features as multi-hot labels or continuous values rather than binary labels.

In addition, the Kansai dialect text in the dialect dataset contains texts that are not different from standard Japanese in literal terms alone, although, in this paper, we use only the binary styles of "standard/Kansai dialect," as mentioned in the results section. To solve this, some form of accent information, not only surface information, should be taken into account.

Applicability of CWS to LLMs

High-performance LLM models typically suffer from processing speed bottlenecks. Meanwhile, lightweight LLM models have been developed that offer a trade-off: slightly

reduced performance in exchange for improved processing speed. Our proposed method offers a simple and effective way to incorporate CWS into CVAE models, which can be trained efficiently using nonparallel data. Given CWS's effectiveness in preserving content, we expect it to complement the limitations of lightweight LLMs. In order to enhance LLMs with CWS, efficient learning and fine-tuning strategies for LLMs are needed, which remains a challenge for future research.

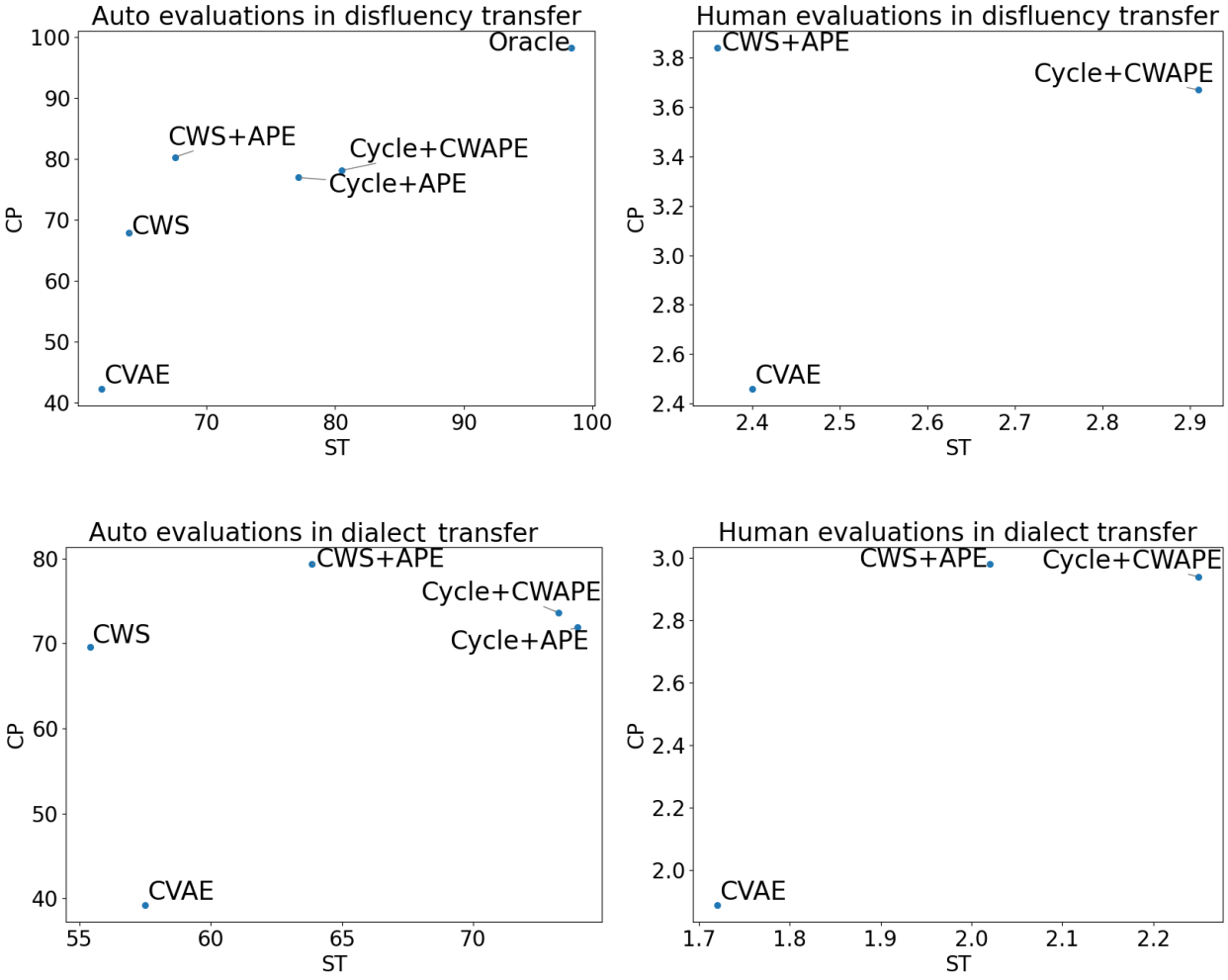

Figure 5.5: *Scatter plots of evaluation results. CP in automatic evaluation were calculated using the formula:* $\{\text{BLEU} + \text{CS} + (100 - \text{CWER})\}/3$. *(In this figure, CVAE + CWS is denoted as CWS, and CycleCVAE + CWS is denoted as Cycle.)*

Table 5.10: *Style-transferred text examples from fluent to disfluent. CVAE + CWS + APE is abbreviated as +CWS+APE, CycleCVAE + CWS + CWAPE as Cycle, and GPT-4 as GPT-4 with nonparallel few-shot.*

Original Text	行動ルールを与えることによってそれが全体としては
Translation	By giving rules of conduct, it is the whole...
CVAE	行動関数を得ることによってそれが
Translation	By obtaining an action function, it...
+CWS+APE	行動ルールを与えることによってそれが全体としては
Translation	By giving rules of conduct, it is the whole...
Cycle	行動ルールを与えることによってそれが全体としてえー
Translation	By giving rules of conduct, it is the whole, well, ...
GPT-4	行動ルールを与えることによってえっとそれが全体としては
Translation	By giving rules of conduct, well, it is the whole, ...

Table 5.11: *Style-transferred text examples from disfluent to fluent. CVAE + CWS + APE is abbreviated as +CWS+APE, CycleCVAE + CWS + CWAPE as Cycle, and GPT-4 as GPT-4 with nonparallel few-shot.*

Original Text	え算出手順をえ簡単に説明いたしますと
Translation	Uh, a brief explanation of the calculation procedure is, well, ...
CVAE	今回算出実験をこのようにします
Translation	This is how we will do the calculation experiment this time.
+CWS+APE	この算出手順を御簡単に説明いたしますと
Translation	A simple explanation of this calculation procedure is...
Cycle	この算出手順を極めて簡単に説明いたしますと
Translation	A very simple explanation of this calculation procedure is...
GPT-4	算出手順を簡単に説明いたします
Translation	The following is a brief explanation of the calculation procedure.

Table 5.12: *Style-transferred text examples from standard to Kansai Japanese. CVAE + CWS + APE is abbreviated as +CWS+APE, CycleCVAE + CWS + CWAPE as Cycle, and GPT-4 as GPT-4 with nonparallel few-shot.*

Original Text	平均誤差はあまり差は見られなかったんですけれども
Translation	The mean error did not show much difference, ...
CVAE	平均年収はあまり差は見られへん
Translation	Average annual income disnae show much difference.
+CWS+APE	平均誤差はあまり差は見られへんんですけれども
Translation	The mean error didnae shown much difference, ...
Cycle	平均誤差はあまり差は見られへんけどな
Translation	The mean error disnae show much difference.
GPT-4	平均誤差はあんまり差は見られへんかったんやけど
Translation	The mean error didnae show much difference, ...

Table 5.13: *Style-transferred text examples from Kansai to standard Japanese. CVAE + CWS + APE is abbreviated as +CWS+APE, CycleCVAE + CWS + CWAPE as Cycle, and GPT-4 as GPT-4 with nonparallel few-shot.*

Original Text	その人が言うことやから間違いないやろなーと思って
Translation	I thocht that sin he said it, it maun be richt ...
CVAE	その人が言うことだからきっと
Translation	Since he said it, it must be ...
+CWS+APE	その人が言うことでは間違いないでしょうか思って
Translation	I'm sure that what he says mast be true, ...
Cycle	その人って言うことは間違いないかと思って
Translation	I thought that what he says must be true.
GPT-4	その人が言うことだから間違いないでしょうねと思いました
Translation	I thought that since he said it, it must be true.

6 Disfluency Annotation for TST + TTS

This chapter addresses the first challenge introduced in Section 1.2.1: the *challenge of generating spontaneous speech from fluent text*. Building on Chapter 5, where we proposed CycleCVAE+CWS to improve nonparallel text style transfer (TST) while balancing content preservation and style controlability, we now integrate TST with a text-to-speech synthesis (TTS) back-end to enable the generation of *spontaneous speech* that includes disfluencies.

This chapter is organized as follows: Section 6.2 introduces related work on filled pause (FP) annotation for TTS. Section 6.3 proposes the TST–TTS cascaded method using disfluency annotation. Sections 6.4 through 6.6 describe our experiments to evaluate and compare the effectiveness of disfluency annotation in TST, TTS, and the cascaded method. Finally, Section 6.7 concludes our findings and discusses future work.

6.1 Introduction

With rapid improvements in speech synthesis and recognition performance driven by machine learning and deep learning, we are now entering an era in which spoken communication between "people and computers" and between "people via computers" is commonplace. In this context, research on spontaneous TTS is essential.

As noted in Section 3.1.1, conventional speech-synthesis research has emphasized

fluent, clear delivery. Human speech, however, is not always fluent; it frequently contains disfluencies that signal spontaneity and are essential for conversational systems and educational audio. Existing spontaneous TTS systems typically assume that such phenomena are already present in the input text, which limits their applicability.

To overcome this limitation, our approach first uses a TST module (CycleCVAE + CWS) to introduce disfluencies into otherwise fluent input, and then feeds the resulting text to a TTS module, thereby achieving *direct generation of natural, human-like spontaneous speech*. To address the third challenge identified in Section 1.2.3—inconsistent treatment of disfluency across modules—we employ a *disfluency annotation* scheme that explicitly represents the position and type of disfluencies in both TST and TTS.

6.2 Related Work: FP Annotation

In a related study, Székely *et al.* [6] performed speech synthesis using Tacotron2 [65] with annotations for two FPs: "uh" and "um" in English. Their research involved objective analysis and subjective perceptual evaluation. They examined the following: 1) the effects of the FPs "uh" and "um" in the context of a neural TTS system trained on a large single-speaker spontaneous speech corpus; 2) the degree of FP control in output speech resulting from varying levels of detail in FP annotation during training; and 3) the capability of TTS using probabilistic models to reproduce FP patterns from training data, along with the effects of different levels of FP control in output speech on perception.

The first TTS system introduced is AutoFP, which trains on speech that contains FPs, whereas the text omits these FPs. Owing to the nature of Tacotron's statistical speech synthesis, which probabilistically reproduces the most likely patterns learned from the data, FPs are automatically generated in the output speech when fluent text

is input, and the positions and types of FPs cannot be specified. The second system is CtrlFP, which trains FPs using text explicitly annotated with unique symbols <uh> and <um>. CtrlFP gives the user control over the placement of these FPs, similarly to how they would control the placement of regular words. Finally, GenFP serves as an intermediate system between AutoFP and CtrlFP, using a single generic symbol <FP> for both "uh" and "um." It learns only the positions of the FPs, not their specific types.

Objective evaluation results indicate that AutoFP learns to automatically reproduce patterns of FP locations and types similarly to those found in the training corpus. Subjective listening tests suggest that listeners generally prefer the FPs rendered by GenFP over those from CtrlFP, which specifies ground truth (GT) FPs. This result shows that human listeners perceive the FPs produced by GenFP as more authentically hesitant. From another perspective, the authors also demonstrate that the complete control of FPs by CtrlFP can slightly enhance the fluent speech synthesis performance of TTS models trained with a speech that contains disfluencies.

Furthermore, we confirm that Tacotron2 and FastSpeech2, both commonly used TTS systems in this study, struggle to reproduce spontaneous behaviors accurately when trained solely on a corpus of spontaneous Japanese speech. It is important to note that spontaneous speech has several types of disfluency besides FPs, and the alignment between the speech and its transcribed text is more unstable than when it does not contain disfluencies. On the basis of the proposed methodology, we plan to extend the annotations to include more disfluencies in Japanese using diffusion- and VAE-based TTS (DVT) [155], which performs effectively even when the alignment between input text and output speech is uncertain or incorrect.

6.3 Proposed Method

6.3.1 Overview

Using disfluency annotation, we propose a method to generate spontaneous speech with disfluency from text without disfluency. Figure 6.1 shows an overview of our proposed method. In the overall process, we first preprocess the transcript data to divide fluent and disfluent texts and to conduct disfluency annotation. Then, we train the bidirectional TST system using disfluency-annotated text and the TTS system using disfluency-annotated text and spontaneous speech. In TTS training, we conduct the customized grapheme-to-phoneme (g2p) for transforming disfluency-annotated text to a disfluency-annotated phoneme sequence described later in Section 6.5. Finally, we use the TST system to add disfluency to fluent text and the TTS system to generate spontaneous speech with the disfluency-added text as input.

We adopt the labeling method for disfluency annotation from prior work [6] with minor modifications. Our labeling method uses slightly different annotations between the TST and TTS. In Section 6.3.2, we explain our labeling method for each TST and TTS, as well as a combined labeling method for TST and TTS systems. Section 6.3.3 describes the TTS method for spontaneous speech synthesis that can render disfluency on the basis of our disfluency annotations.

Note that disfluency annotation can be implemented simply by adding a special token to each text or phoneme token's vocabulary. It does not require special processing for the annotated text and phoneme, and TST and TTS are not dependent on any particular model. Although we use existing models described in Chapter 5 and 6.3.3 for TST and TTS, the crux of our proposal is the disfluency annotation strategy and the integration of TST and TTS.

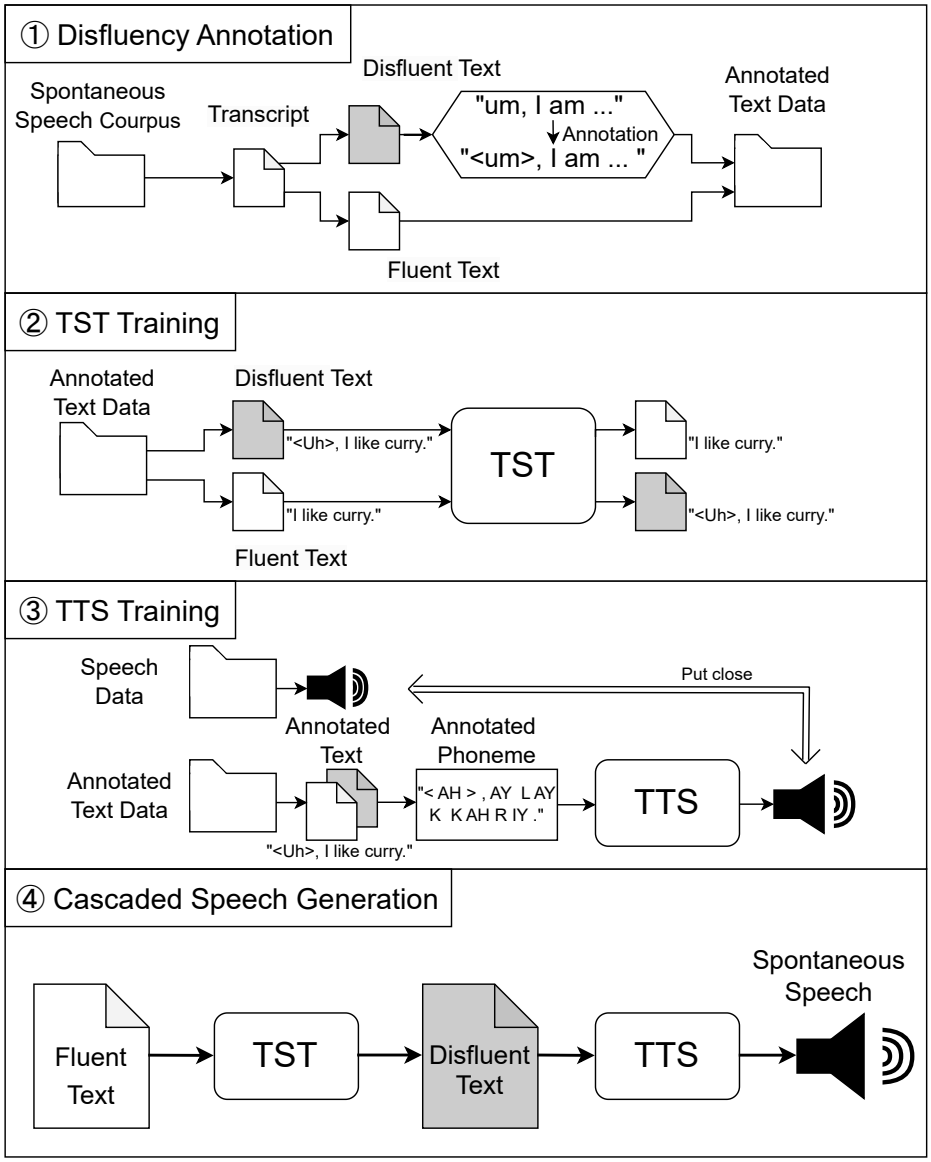

Figure 6.1: *Overview of our proposed method. We first preprocess the transcript data with disfluency annotation and make fluent and disfluent texts. Secondly, we train the bidirectional TST system using disfluency-annotated texts and the TTS system using disfluency-annotated texts and spontaneous speech. Finally, we use the TST system to add disfluency to fluent text and the TTS system to generate spontaneous speech from disfluency-added text.*

Table 6.1: *Summary of disfluency annotation for TST.*

Name	Annotation	Description
Plain	No annotation	Not explicitly considered.
Symbol	[FILLER]/[SLIP]	Only position and type are specified.
Tag	< えー > or (ん)	Specified with registered disfluency words.
S-Tag	＜ えー ＞ or （ ん ）	Specified with arbitrary words.

Table 6.2: *Summary of disfluency annotation for TTS.*

Name	Annotation	Description
Plain	No annotation	Not explicitly considered.
Symbol	$ or #	Only position and type are specified.
Tag	<ee> or (N)	Location, type, and word are specified.
Auto	No annotation	Texts don't include disfluencies.

6.3.2 Disfluency Annotation

Disfluency Annotation for TST

Table 6.1 shows disfluency annotation methods for TST. We propose the following three disfluency annotation methods for TST:

- **Plain**. The `Plain` method does not use annotation as a baseline. `Plain` cannot explicitly consider which word is disfluency.
- **Symbol**. The `Symbol` method converts disfluency words into unique symbol tokens corresponding to their respective types. Specifically, [FILLER] is used for FPs, and [SLIP] is used for word fragments caused by misspeaking or stuttering (stutter word). This method is expected to simplify training and improve the

overall performance of the TST model because the model needs to consider only the location and type of disfluency.

- **Tag**. The `Tag` method adds brackets around disfluency words. This method does not insert a space between the word and the parentheses and treats each disfluency as a unique token. Mountain brackets indicate FP words and round brackets indicate stutter words. This approach will enhance style control performance by allowing the TST model to learn which words are disfluencies.

- **Space-Tag (S-Tag)**. The `S-Tag` method adds brackets around disfluency words. A space between the word and parentheses is inserted to treat the brackets as independent tokens. Mountain brackets indicate FP words and round brackets indicate stutter words. This approach differs from the `Tag` method in that it does not distinguish between disfluency words and other words at the dictionary registration stage, increasing the transfer flexibility but also complicating learning.

Disfluency Annotation for TTS

Table 6.2 shows disfluency annotation methods for TTS. We propose the following three disfluency annotation methods. As shown later in Section 6.4.3, `S-TAG` significantly impairs content preservation by assigning parentheses to no disfluent words. For this reason, `S-TAG` was excluded from the methods using TTS.

- **Plain**. The `Plain` method does not use annotation as a baseline. `Plain` cannot explicitly consider which word is disfluency.

- **Symbol**. The `Symbol` method uses the special phonetic symbols corresponding to each type for the entire phoneme sequence corresponding to a disfluency word

as input representation. Specifically, FPs are converted to $ and stutter words are converted to #. This approach is expected to create a TTS system in which the user specifies the location of the disfluency, and the TTS model automatically synthesizes the appropriate disfluency.

- **Tag**. The `Tag` method puts the entire phoneme sequence corresponding to a disfluency word in special symbols for each type. Specifically, FPs are enclosed with <> and stutter words are enclosed with (). This approach is expected to allow a TTS system to account for acoustic differences between disfluency and nondisfluency words. In addition, the user has control over the location, type, and words of disfluency.

- **Auto**. The `Auto` method excludes all disfluency words from texts. This approach is expected to realize a TTS system that automatically synthesizes disfluencies into speech, even if the text input does not include disfluencies.

Disfluency Annotation Combinations for TST + TTS

We propose three ways to combine disfluency annotations in a combined TST and TTS system.

- **Plain+Plain (PP)**. Use a model with `Plain` in both TST and TTS as a baseline.

- **Symbol+Symbol (SS)**. Use a model with the `Symbol` annotations in both TST and TTS. This approach corresponds to determining the location of disfluencies on the TST system side and the type and words on the TTS system side.

- **Tag+Tag (TT)**. Use a model with the `Tag` annotations in both TST and TTS. This approach determines all locations, types, and words of disfluencies on the

TST system side.

- **None+Auto (NA)**. No style transfer is applied to the text, and spontaneous speech synthesis is performed on text that does not contain disfluencies. This approach determines all locations, types, and words of disfluencies on the TTS system side.

6.3.3 Base Model for TTS: DVT

DVT [155] is a TTS method using a diffusion probabilistic model. Figure 6.2 shows the model architecture of DVT. The method consists of three components: a waveform model consisting of an acoustic encoder and a waveform decoder, a latent acoustic model that converts language features into latent acoustic representations with diffusion, and an alignment model that considers the correspondence between the series of latent linguistic and acoustic representations.

During training, the waveform model predicts speech waveforms from acoustic features X via latent acoustic representations z_X. The acoustic encoder encodes acoustic features X into the latent acoustic representation z_X, which follows the Gaussian distribution with the approximated mean $\mu_\psi(X)$ and variance $\sigma^2_\psi(X)$. The waveform decoder decodes the waveform given z_X. The latent acoustic model predicts the latent acoustic representation z_X from the phoneme sequence Y via the latent linguistic representation z_Y. We obtain the phoneme sequence Y from the input text using the customized g2p described in Section 6.5. Unlike conventional diffusion models, which diffuse the mean from 0 to x_0 and the variance from 1 to 0, this model diffuses the mean from 0 to $\mu_\psi(X)$ and the variance from 1 to $\sigma^2_\psi(X)$, leveraging the known distribution of z_X. The approximate posterior is defined by setting $\mu_\psi(X)$ as the target and

interpolating variance from $\sigma_{\psi}^2(X)$ to 1. The prior distribution is assumed to be standard Gaussian, and the model function $f_\theta(x_t, t, z_Y)$ predicts the mean and variance, optimized by minimizing KL divergence, where t means diffusion time. The alignment model learns to align the latent acoustic representation z_X and linguistic representation z_Y. A monotonic path is searched as alignment in a trellis defined by distances between the two representations [156]. To measure the distances between the different representations, an alignment function $g_\phi(z_X) \mapsto z_Y$ is introduced to map z_X to z_Y. The parameter of the alignment function ϕ is optimized to minimize distances between z_Y and $g_\phi(z_X)$. The duration model is trained with the phoneme duration obtained from the alignment. DVT is trained in two stages: in the first stage, the waveform model is trained independently, and in the second stage, other models are trained with the parameters of the waveform model fixed.

During inference, the duration model first predicts the duration using the latent linguistic representation obtained from the text encoder and then upsamples the latent representation. The latent acoustic model predicts the latent acoustic representation using the upsampled latent linguistic representation. Finally, the waveform model generates speech waveforms from the latent acoustic representation.

Because the diffusion model adds noise to the input and removes it as a learning criterion, the synthesized speech will be clean and high-quality. Moreover, one of the characteristics of DVT is its robustness to input format. In general, TTS performs better with phoneme input than with character input. An original paper showed that DVT performs better on character input than phoneme input, whereas other comparative methods perform worse on character input [155]. Another paper showed that DVT can produce correct speech more robustly than other methods, even when the input text contains a large amount of noise derived from automatic speech recognition [157]. This

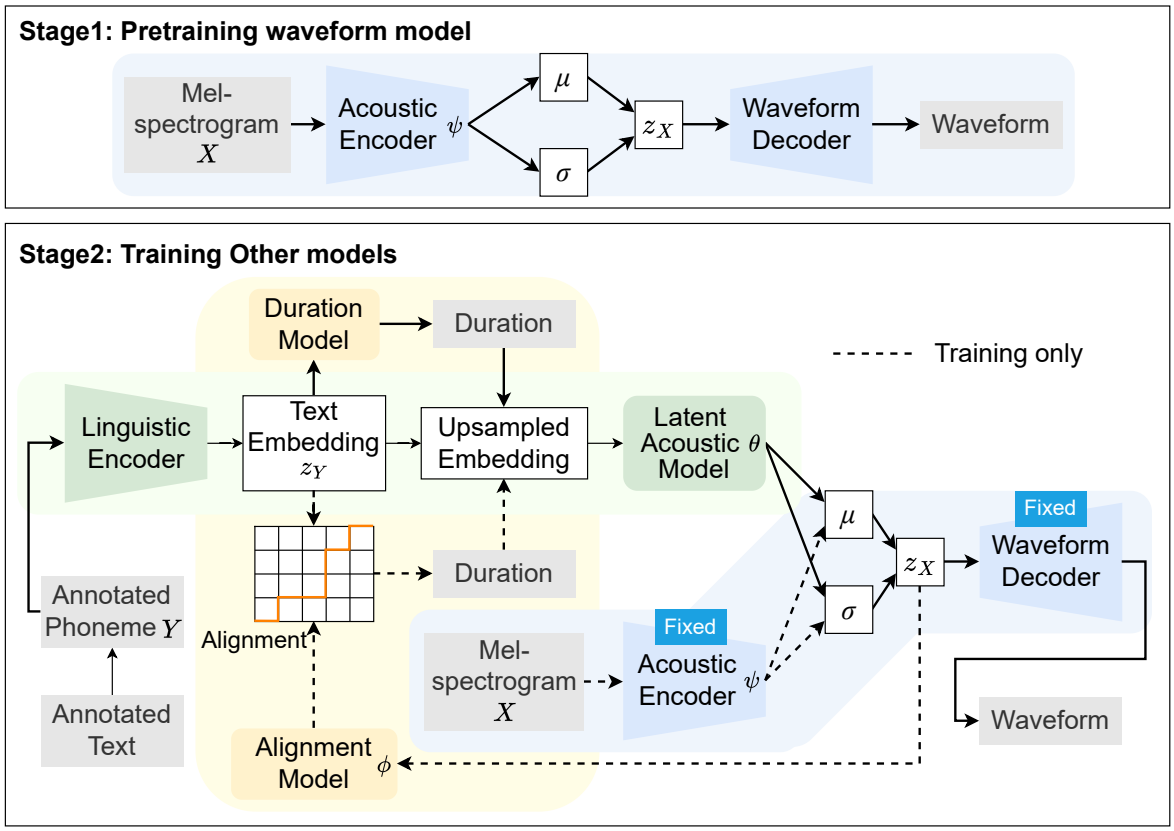

Figure 6.2: *Model architecture of DVT.*

finding suggests that DVT can effectively generate spontaneous speech that contains many difficult-to-model elements, including disfluency.

6.4 Experimental Evaluation 1: Disfluency Annotation for TST

6.4.1 Settings

To confirm the effectiveness of disfluency annotation in TST, we conducted an experiment with style transfer in both directions for "with and without disfluency." We used

CycleCVAE+CWS [158] as the TST method. The systems compared in the experiment were `Symbol`, `Tag`, and `S-Tag`, which were trained by applying each of the three proposed annotation methods. We also used `Plain` as a baseline, which was trained without disfluency annotation. In the objective evaluation experiment, the average of the entire test data was calculated for each evaluation metric described in Section 6.4.2.

For the experimental data, we used the corpus of spontaneous Japanese (CSJ) [49]. CSJ has manually annotated the word, for example, (F えー) (FP) and (D ん) (stutter word). We preprocess the transcripts from CSJ in the following steps.

1. **Devided the data** by labeling each text as "with" or "without" disfluency using CSJ's manually annotated word labels.
2. **Deleted manually annotated word labels** without (F) and (D) labels (CSJ includes some other labels, such as whisper (L) and laughing (笑)).
3. **Separated transcripts** into short units of about 10–20 words.
4. **Labeled each unit** according to whether it contained a disfluency.
5. **Processed by each annotation method**.

The specific process for each method is as follows: in `Plain`, the (F) and (D) labels were removed; in `Symbol`, the words labeled (F) and (D) were replaced with [FILLER] and [SLIP]; in `Tag` and `S-Tag`, the words labeled (F) and (D) were enclosed with <> and (). In `Auto`, which is not used in this experiment but will be used in Experiment 2, the words labeled (F) and (D) are deleted. The TST used in this paper does not support the transfer of long sentences, so we separated transcripts into short units. We obtained 349,983 fluent and 330,650 disfluent texts; the total is 680,633. We split them into 654,817, 17,222, and 8,594 texts to construct training, validation, and test datasets at a 96.0:2.5:1.5 ratio.

6.4.2 Metrics

The following three types of objective evaluation indicators were used:

- **AC**. Accuracy (AC) is the style accuracy rate used for evaluation to indicate style control performance. In the calculation, a CNN classifier for texts [152] was first trained using training data. Using this, we predicted the text style generated by each model and calculated the percentage of text that could be classified as being in the target style.

- **BLEU**. BLEU [104] is a metric proposed in machine translation. We measured the content preservation of the entire text by calculating the n-gram overlap (n = 1–4) of the generated and reference texts and taking the average of the overlaps. To maintain fairness with the evaluation in `Plain`, each special symbol was converted to the most frequent FP (えー) and stutter word (ん) in `Symbol`, and brackets were removed in `Tag` and `S-Tag` before calculation.

- **Content word error rate (CWER)**. CWER [158] is a metric similar to the word error rate (WER) used in speech recognition, specifically applied to content word sequences. In TST, the content words of input and generated texts should not change. Therefore, CWER was used to calculate WER for "content word series of generated text" and "content word series of input text" to measure the preservation of content words.

6.4.3 Results

Table 6.3 shows each experiment's objective evaluation results. From the results, the proposed method outperformed the baseline `Plain` method in each metric. In

Table 6.3: *Automatic evaluation results for TST.*

Method	**AC (%)** ↑	**BLEU** ↑	**CWER (%)** ↓
Oracle	93.73	100.00	0.00
Plain	61.31	**58.33**	7.68
Symbol	**95.36**	57.56	8.13
Tag	79.25	57.96	**7.62**
S-Tag	82.74	50.36	14.25

particular, AC was improved by all the disfluency annotation methods, and BLEU and CWER were comparable or improved by `Symbol` and `Tag`, respectively. From this, disfluency annotation enabled the TST model to significantly improve its style control performance without compromising content preservation. On the other hand, `S-Tag` showed a higher AC than `Plain` and `Tag`, but BLEU and CWER were degraded. This result was because the model added brackets to words other than those indicating disfluency, suggesting the need for more rigorous annotation that treats disfluency as an independent token rather than simply indicating which position in an existing word is disfluency. On the basis of this result, we will use `Symbol` and `Tag` in the TST + TTS experiment described in Section 6.6.

Table 6.4 shows the evaluation results for each transfer direction to confirm the results in more detail. Although `Plain` performed well in style control in the disfluent to fluent direction, its performance in the reverse direction was poor. This result indicated that `Plain` can add disfluency to only about 40% of the generated text. In contrast, the three proposed methods showed a higher AC than `Plain` in the direction of fluent to disfluent. Since this paper aims to generate disfluent speech from fluent

Table 6.4: *Performance comparison between different style transfer directions.*

Method	**Direction**	**AC (%) ↑**	**BLEU ↑**
Plain	fluent → disfluent	39.67	53.31
	disfluent → fluent	85.75	53.30
Symbol	fluent → disfluent	99.25	53.23
	disfluent → fluent	90.96	55.94
Tag	fluent → disfluent	70.25	53.72
	disfluent → fluent	90.49	54.66
S-Tag	fluent → disfluent	72.55	51.97
	disfluent → fluent	94.20	41.58

text, these results indicate that using disfluency annotation in TST was effective.

Finally, we qualitatively inspected the generated texts. For `S-Tag`, we observed failure cases such as (i) annotations incorrectly attached to content words , e.g., "非常 に ＜ 深い ＞ 人 で あり ます" ("(They are) very < deep >."), and (ii) malformed annotation markers where only the left bracket token ("<") was generated without the corresponding right token. These errors are consistent with the degradation in BLEU and CWER, suggesting that bracket-based marking is brittle unless disfluency is represented as an independent token with stricter constraints.

For `Plain` and `Symbol`/`Tag`, `Symbol` and `Tag` inserted disfluencies into a larger portion of utterances than `Plain`. However, the overall tendencies of the predicted disfluency positions (and the disfluency types in `Tag`) did not differ substantially across these methods.

6.5 Experimental Evaluation 2: Disfluency Annotation for TTS

6.5.1 Settings

We conducted a subjective evaluation experiment to confirm the effectiveness of disfluency annotation in TTS. First, as a preliminary experiment, we conducted a mean opinion score (MOS) test, which evaluated the overall naturalness of spontaneous speech to select a suitable model for Spontaneous TTS. We compared two acoustic models, FastSpeech2 and DVT, which were trained with CSJ under the `Plain` condition, plus human speech (GT), for three types of speech for evaluation. HiFiGAN was used for the waveform model in both FastSpeech2 and DVT. However, because the input features differ, we used different training models for the two methods. Ten male and female native Japanese speakers in their 20s listened to each voice individually and rated each on a five-point scale from 1 to 5. We used 50 samples of 2 to 12 s containing disfluency words randomly selected from the test data. Different participants rated each sample at least six times, obtaining 300 evaluations for each system. Finally, we conducted Mann–Whitney's U test to confirm the statistical significance of MOS.

The second experiment was an ABX test with reference speech to confirm the effectiveness of applying disfluency annotation. We used the DVT TTS model for this experiment. We compared three systems, `Auto`, `Symbol`, and `Tag`, trained by applying each proposed annotation method. We also used `Plain`, trained without disfluency annotation, as a baseline. We employed 100 native Japanese speakers as crowd workers and experimented with a web test format. First, we presented the participants with a description of the "disfluency" to be evaluated, sample human voices for each, and sample synthesized voices that both acoustically reproduced the disfluency and did

not. Then, the participants listened to a reference speech, X, followed by two synthesized speeches, A and B. They were instructed to choose the speech that more closely reproduced X's disfluency acoustically. Additionally, the participants evaluated which of the two speeches sounded more natural and spontaneous, regardless of X. Here, we aimed to assess the style reproducibility and naturalness of the synthesized speech. However, since the synthesized speech for each annotation method was predicted to differ in its textual content from the original speech, the participants were concerned that they might focus on textual similarities rather than acoustic or stylistic features, introducing noise into the evaluation. For this reason, we used speeches A and B synthesized from different texts of the same speaker as X in this Section and the next Section 6.6. We used 50 randomly selected 5 to 15 s samples containing disfluency words from the test data. Different participants rated each sample at least eight times, obtaining 400 evaluations for each of the six system combinations. We calculated the evaluation values as the preference score of voice B over voice A by calculating the average of the evaluations, which was 1 when voice B was selected as better than voice A and 0 when voice A was selected as better. Finally, we conducted a binomial test to confirm statistical superiority.

We used approximately 400k speech samples and their corresponding transcriptions from the CSJ [49] for our experimental data. It contained fluent and disfluent samples. Within the CSJ, about 7% of the total data is classified as "core data," which includes more detailed manual annotations such as accent and phoneme labels. However, in this study, we used the entire CSJ dataset. Additionally, in the next Section 6.6, we used style-transferred text that has no accent and phoneme information. Therefore, we employed a dictionary in the external text processing front-end tool, pyopenjtalk[1], to process the grapheme-to-phoneme conversion and extract accent labels for each

[1] `https://github.com/r9y9/pyopenjtalk`

phoneme. In this experiment, since we used disfluency-annotated text, we developed a custom pyopenjtalk, which has the function of converting and retaining a special text token to a special phoneme token. Specifically, we converted [FILLER] to $, [SLIP] to #. Parentheses, such as <> and (), were lost when converted from graphemes to phonemes in pyopenjtalk's default settings, so we changed the code to allow them to remain.

Furthermore, we used speaker labels during TTS training to build the multispeaker TTS model. Since CSJ does not have specific speaker labels, we utilized lecture IDs as pseudo-speaker labels. The number of lecture IDs was 3,224. These included a few lectures by the same speaker, but we did not consider the speaker duplicates and used all the lectures this time. We set the dimension of the speaker embedding to 256. Note that FastSpeech2 was implemented by ESPnet2 [159], which used a pretrained x-vector [160] with this pseudo-speaker label, whereas DVT trained the speaker embedding from scratch.

In the latent acoustic model of DVT, we set the number of diffusion steps to 100. We sampled diffusion time t uniformly and optimized KL divergence directly as in the original DVT settings [155]. We trained plain DVT up to 830k steps and FastSpeech2 up to 1M steps. We trained DVT for `Symbol` up to 738k steps, `Tag` up to 773k steps, and `Auto` up to 818k steps. The system used to synthesize each speech remained undisclosed to participants throughout both experiments. The audio samples used in Experiments 2 and 3 are available at the URL in the notes[2].

Table 6.5: *Evaluation results for MOS in TTS.*

Method	**MOS ↑**
GT	4.69 ± 0.07
FastSpeech2	2.78 ± 0.14
DVT	**3.01 ± 0.12**

6.5.2 Results

Table 6.5 shows MOS evaluation results. DVT showed better results than FastSpeech2, and the U-test showed significant differences between the two systems and between each system and GT. Speech from FastSpeech2 was characterized by beeps and mechanical noises throughout, especially in sound prolongations. This could be attributed to the fact that FastSpeech2 uses soft attention for teacher alignment. Soft attention makes it difficult to align the transcript with the speech with phonological stretches. This effect is a problem in spontaneous speech, often including FPs and hesitations. Although DVT could synthesize disfluent speech without noise, it also had unnatural accents and intonations. DVT has good signal quality since noise removal is the learning criterion. Nonetheless, random sampling in DVT generates diverse intonation patterns, which may contribute to the unnatural quality of certain samples. Additionally, although accent labels are provided, they may not be accurately rendered. This is likely because DVT learned the speaker vector simultaneously, making the training process more complex owing to interspeaker variations, even for identical accent labels. The overall naturalness score of DVT was higher than that of FastSpeech2, and we judged DVT to be more in line with our goal of speech synthesis,

[2] `https://dyoshioka-555.github.io/SponTTS-samples/audio_samples.html`

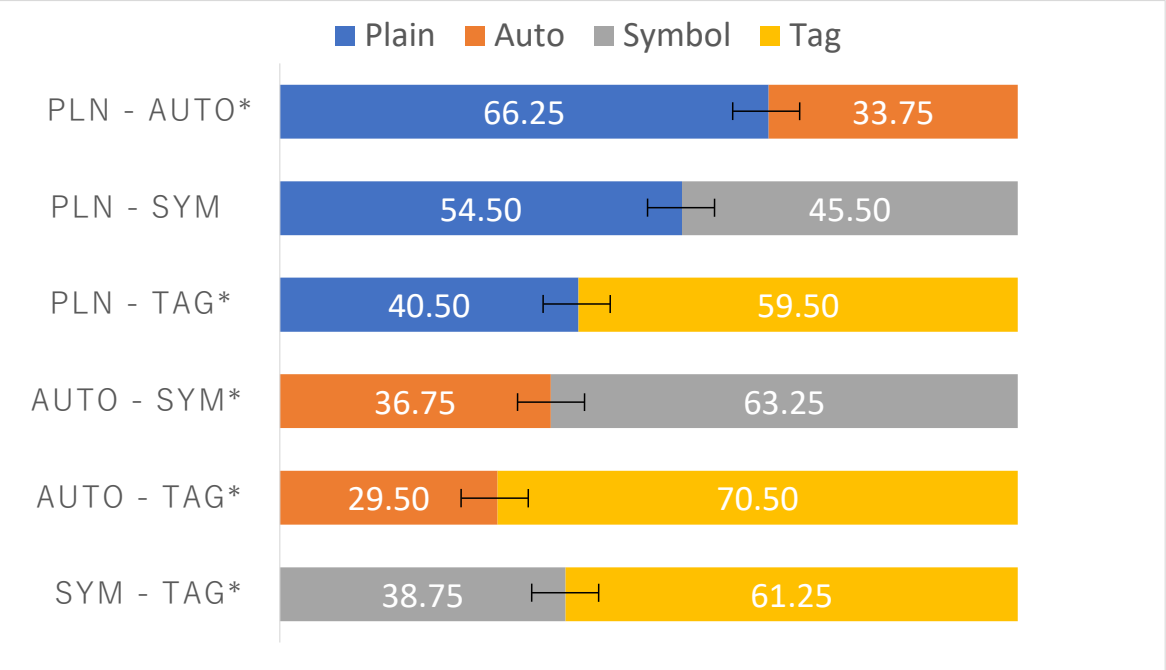

Figure 6.3: *ABX test results on the preference for style reproducibility in TTS. Pairs marked with an asterisk indicate significant differences.*

focusing on disfluency. Therefore, we decided to use DVT in subsequent experiments.

Figures 6.3 and 6.4 show the results of the ABX test and the presence or absence of significant differences by the binomial test. Pairs marked with an asterisk indicate significant differences. The results confirmed that `Auto` is significantly inferior to all other systems in style reproducibility and that `Symbol` is superior to `Auto` but significantly inferior to `Tag`. These results differ from previous studies [6], which concluded that listeners generally prefer the disfluencies selected by `Symbol` over those specified GT disfluencies by `Tag`.

This difference suggests that when the number of disfluency types increases, the TTS system may find it challenging to automatically select the appropriate disfluency location and content.

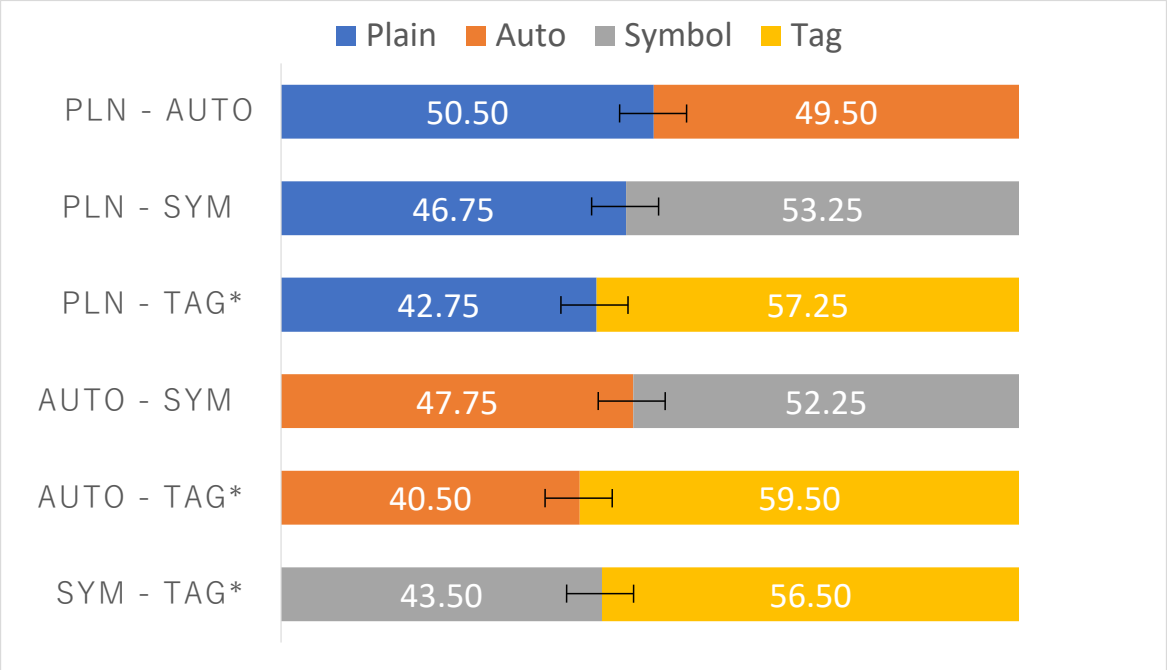

Figure 6.4: *ABX test results on the preference for naturalness in TTS. Pairs marked with an asterisk indicate significant differences.*

On the other hand, the most detailed annotated TTS system, `Tag`, significantly outperforms all other systems regarding style reproducibility and overall naturalness. This result suggests that detailed annotation can reproduce disfluency more naturally for the TTS system.

Finally, Table 6.6 compares the transcriptions obtained from the utterances synthesized by TTS using the annotations of each proposed method. For `Auto`, the output exhibits only a slight lengthening at the beginning of the utterance, and the system fails to automatically render disfluencies in most cases. This observation helps explain why `Auto` is substantially worse than the other methods in style reproducibility. Next, for `Symbol`, the system does not generate diverse realizations for each `[FILLER]` token and outputs almost exclusively a single filler (“えー”). Although not shown in the

Table 6.6: *Transcrptions obtained from the utterances synthesized by TTS using the annotations of each proposed method.*

Plain input	でこれ英語でまずうー考えられたあのー概念でしてえー…
Transcrption	でこれ英語でまずう考えられたあのー概念でしてえー…
Translation	This is a concept that, that was uh first conceived in English, um…
Auto input	でこれ英語でまず考えられた概念でして…
Transcrption	でーこれ英語でまず考えられた概念でして…
Translation	Thiiis is a concept that was first conceived in English,…
Symbol input	でこれ英語でまず [FILLER] 考えられた [FILLER] 概念でして [FILLER]…
Transcrption	でこれ英語でまずえー考えられたえー概念でしてえー…
Translation	Well, this is a concept, well, that was first conceived in English, well,…
Tag input	でこれ英語でまず < うー > 考えられた < あのー > 概念でして < えー >…
Transcrption	でこれ英語でまずうー考えられたあのー概念でしてえー…
Translation	This is a concept that, that was uhh first conceived in English, um…

example, we also observed cases where (i) the realized duration of each disfluency was shorter than expected and (ii) consecutive symbols were compressed and spoken as a single disfluency event. We believe these issues contribute to the inferior ABX results of `Symbol` compared with `Tag`. For `Tag`, the transcribed text appears similar to that of `Plain`, suggesting that many differences are not visible at the text level. However, when listening to the synthesized speech, `Tag` produces more disfluency-like temporal patterns (e.g., more appropriate prolongation and cutoff behaviors) and sometimes includes voice-quality phenomena such as creaky voice [161], resulting in speech that sounds more natural and spontaneous.

6.6 Experimental Evaluation 3: Disfluency Annotation for TST + TTS

6.6.1 Settings

We conducted subjective evaluation experiments, an ABX test as in Section 6.5, to confirm the effectiveness of disfluency annotation in a combined TST and TTS system. We used CycleCVAE+CWS as the TST model and DVT as the TTS model. We compared the three proposed ways of combining the annotation method: `NA`, `SS`, and `TT`. We also used `PP`, trained without disfluency annotation, as the baseline. We employed 100 native Japanese speakers as crowd workers and experimented with a web test format. First, we presented the participants with a description of the "disfluency" to be evaluated and sample human voices for each. Then, the participants listened to a reference speech, X, followed by two synthesized speeches, A and B, synthesized from different texts of the same speaker as X. The participants selected the speech that better reproduced X's overall disfluency style. We also evaluated which of the two voices was more natural and spontaneous, regardless of X. We used 50 randomly selected 5 to 15 s samples that did not contain disfluency words from the test data and applied TST to each transcript.

Different participants evaluated each sample at least eight times, obtaining 400 evaluations for each of the six system combinations. We calculated the evaluation values as the preference score of voice B over voice A by calculating the average of the evaluations, as in Section 6.5.1. Finally, we conducted a binomial test to confirm statistical significance.

For the experimental data, we used about 400k speech data and its transcription in CSJ [49] as in Sections 6.4 and 6.5. As in Section 6.5, we built the multispeaker TTS

model using pseudo-speaker labels.

6.6.2 Results

Figures 6.5 and 6.6 show the results of the ABX test and the presence or absence of significant differences by the binomial test. Pairs marked with an asterisk indicate significant differences. The results confirmed that `NA` is significantly inferior to all other systems regarding style reproducibility. There is no significant difference between `PP` and `SS` and `SS` and `TT`, but the preference scores are in the order `TT` > `SS` > `PP`, and there is a significant difference between `PP` and `TT`. This result showed that the use of `Tag` annotations in TST and TTS improves style reproducibility compared with the use of `Plain`.

Regarding naturalness, we found no significant difference among `NA`, `PP`, and `TT`, but `SS` was significantly inferior to all other systems. `SS` was inferior in naturalness even to `NA`, rated as having the lowest style reproducibility, and there was no significant difference between `Symbol` and `Auto` in Experiment 2. These results suggest a problem with TST using `Symbol`. We discuss the detailed causes in Section 6.6.3.

6.6.3 Discussion

We conducted an experimental evaluation of the application of disfluency annotation to TST, TTS, and a combination of the two. Naturally, the method using `Tag`, the most detailed annotation, improves in all aspects over `Plain`. Even when using the same detailed annotations, the mainstream method for spontaneous speech synthesis in Mandarin is to input detailed annotated labels as separate inputs from the text [94,162]. In contrast, we directly annotate the input text without implementing new inputs into

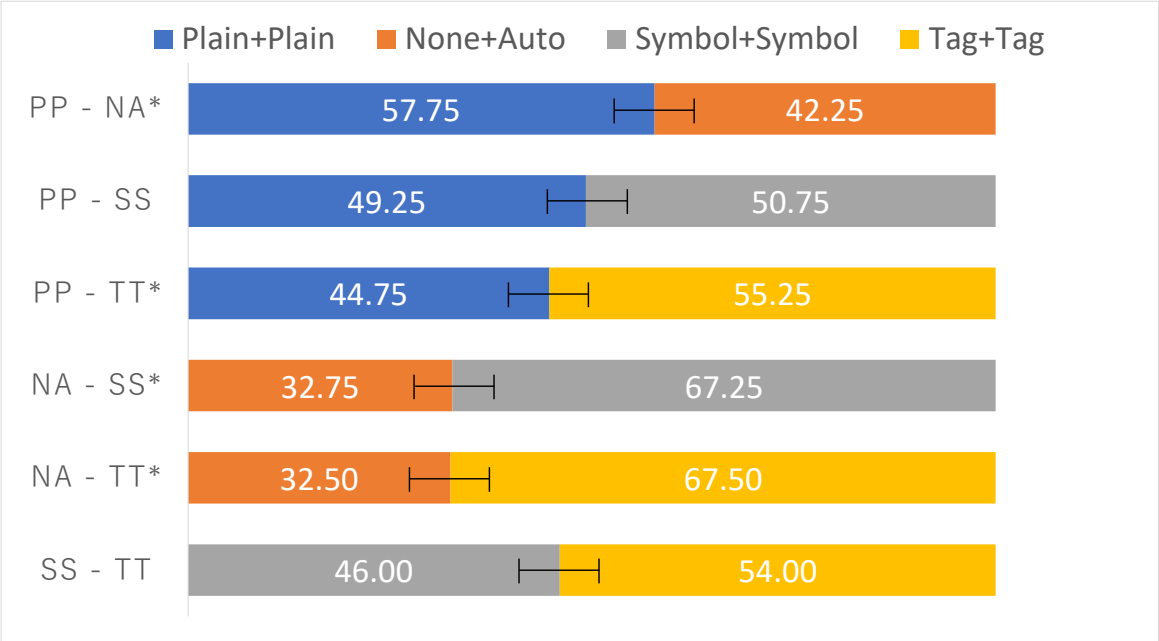

Figure 6.5: *ABX test results on the preference for style reproducibility in TST + TTS. Pairs marked with an asterisk indicate significant differences.*

the existing TTS model. One of our following tasks is to compare our method with the models using labels as separate input with the text.

In English-focused work [6], the method corresponding to `Symbol` had an effect equal to or greater than that of the method corresponding to `Tag`. However, our experimental results showed that the method using `Symbol` is inferior to `Tag` and equal to or inferior to even `Plain`. One possible reason for this is the difference in the number of types of disfluency between English and Japanese. In a large-scale corpus of British English [163], five types of FPs were identified, but in terms of pronunciation, they can be summarized into two series: nasal (erm/um) and non-nasal (eh/uh) [164]. Székely *et al.* [6] also dealt with only two types of FPs: *uh* and *um*. In contrast, Japanese FPs are more varied than English FPs [165]. In addition, since we were also

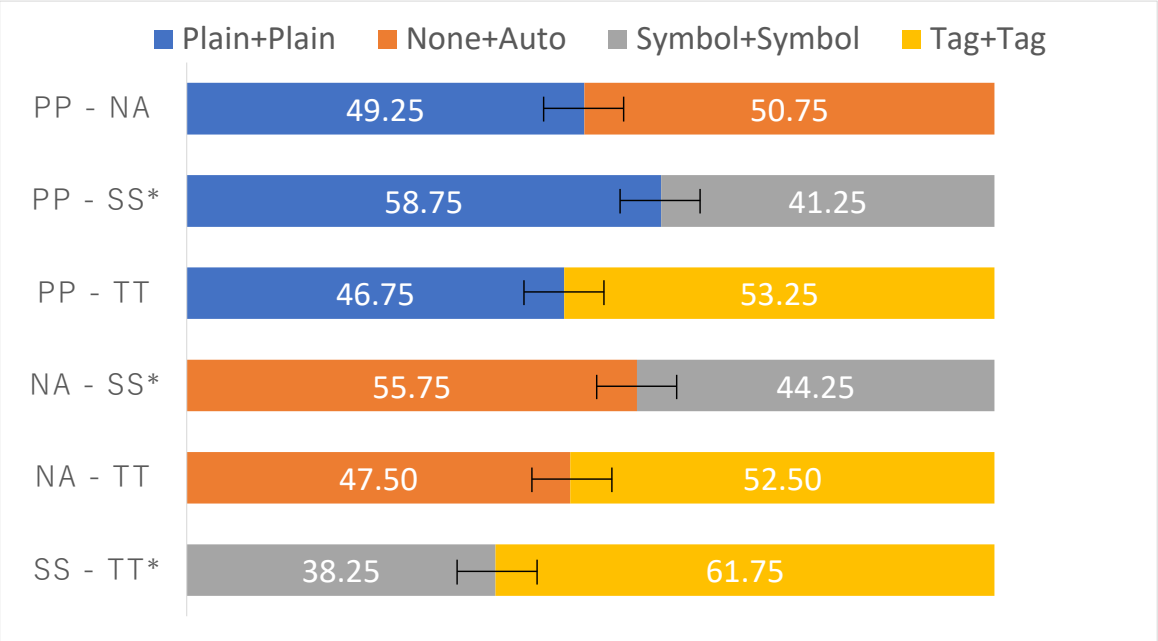

Figure 6.6: *ABX test results on the preference for naturalness in TST + TTS. Pairs marked with an asterisk indicate significant differences.*

dealing with stutter words in this study, it is quite possible that the `Symbol` were not being converted into appropriate disfluency during speech synthesis.

Here, we discuss the factors that prevented `Symbol` with partial annotations from significantly outperforming `Plain` and `Tag` in terms of the naturalness of TST + TTS. First, let us look at the TTS part in Experiment 2. To isolate the problem once, we examined the performance of synthesizing fluent speech from the TTS model using disfluency annotation. We calculated three metrics, Mel-cepstral distortion (MCD), F_0 route mean squared error (F_0 RMSE), and character error rate (CER), to examine the acoustic and prosodic differences and the intelligibility of fluent speech from each method. In calculating CER, we transcribed the synthesized fluent speech using pretrained automatic speech recognition (ASR) implemented by ESPnet2 [159] with

Table 6.7: *Mel-cepstral distortion (MCD) and F_0 route mean squared error (F_0 RMSE) for fluent synthesis speech to original speech, and character error rate (%) for fluent synthesis speech.*

Method	**MCD** ↓	F_0 **RMSE** ↓	**CER (%)** ↓
Test data	–	–	4.47
Plain	5.44	72.47	3.45
Symbol	5.60	65.61	4.14
Tag	5.42	68.05	4.03
Auto	5.50	66.80	5.68

the same data (CSJ). This ASR's average recognition rate for the test data, including disfluency, is about 4.47%. The results of each metric are shown in Table 6.7. `Symbol`'s MCD was slightly higher than the other methods, and its F_0 RMSE was slightly lower. `Plain`, which does not use disfluent annotation, had the best CER, followed by `Tag` and `Symbol`. `Symbol`'s CER was inferior to `Plain` but comparable to `Tag`. The results in CER exceed those in the test data because the test data contains disfluency, but the speech assessed here does not. The results suggest that `Symbol` was acoustically and prosodically comparable to `Plain` and `Tag`, and its acoustic and prosodic features were not a factor that significantly impaired the perception of naturalness.

What about the disfluent part of synthetic speech from `Symbol`? We investigated `Symbol`'s synthesized speech, focusing on the disfluent parts, and did not notice any particular acoustic or prosodic discomfort. On the other hand, in Experiment 2, `Symbol`'s style reproducibility for disfluency was inferior to that of `Plain` and `Tag`. To ascertain the cause of this, we manually examined the ASR results of the synthetic speech containing disfluency used in the experiments and found the following features for decoding

Table 6.8: *The average length of each synthesis speech (sec). Whole is the length of the synthesized speech used in Section 6.5, and Fluent is synthesized speech from transcription without disfluency. Disfluent is the difference between the lengths of Whole and Fluent.*

Method	**Whole (sec)**	**Fluent (sec)**	**Disfluent (sec)**
Test data	6.56	5.86	0.70
Plain	6.00	5.34	0.66
Symbol	5.86	5.26	0.60
Tag	6.18	5.32	0.86
Auto	5.35	5.23	0.22

disfluent symbols into speech: Stutter word symbols may be ignored; FP symbols may be output as other disfluencies, such as stutter words, which are generally not considered as FPs; when there is a sequence of disfluent symbols, some of them may be ignored. In addition, disfluencies synthesized using `Symbol` tend to have shorter durations than those synthesized using other methods. These can be seen by looking at the average of the pseudo disfluency lengths by calculating the difference between the length of the entire synthesized speech used in Section 6.5 and the length of the fluent synthesized speech excluding the disfluency, as shown in Table 6.8. In test data, Whole is measured using original speech, and Fluent is measured using speech that has had disfluency manually removed. We can assume that these factors made it difficult for listeners to perceive disfluency and caused `Symbol` to be rated equal or inferior to `Plain` regarding style reproducibility.

Then, we will look at the TST part in Experiments 1 and 3. By comparing the style-transferred text with `Symbol` among the samples used in this experiment with

those of `Plain` and `Tag` for detailed analysis, we found scattered cases where the TST system output disfluent symbols instead of content words, which was different from the case with `Plain` and `Tag`. Moreover, in Japanese, words such as "あの (that)" and "その (it)" are used as both pronouns and FPs. However, in `Tag`, the accidental transfer of these pronouns to FPs does not markedly affect pronunciation. In contrast, when `Symbol` replaces these words with [FILLER], the TTS system may pronounce them as entirely different FPs (such as "uh," etc.), which could be one of the reasons for the reduced naturalness.

On the basis of these results of our analysis, we predict that to successfully train `Symbol`, which is less expensive than `Tag`, it would be helpful to add part-of-speech information from morphological analysis to supplement information for inferring relationships with surrounding words and to add duration constraints when decoding disfluent symbols into speech.

Another critical factor is the proportion of disfluency in the synthesized text and speech, and the proportion of FPs and stutter words. We have trained the TST and TTS systems to handle these two types of words, and for TST, we have found that the percentage of stuttered word output in the conversion is extremely low. This is because the number of stutter words in the training data is lower than that of FPs, but `Symbol` tends to output a relatively large number of stutter words. Table 6.9 shows the proportion of disfluency in the training data with disfluency and that of the fluent to disfluent transferred text for each model. The results suggest that the low naturalness ratings for `Symbol` in Section 6.6 are due to the high proportion of stutter words included. Possible solutions to the imbalance include training on balanced data or using the same data but repeatedly generating and evaluating it until both types of disfluency are produced.

Table 6.9: *Proportion of disfluency in the text and the proportion of FPs and stutter words (number of words per sentence).*

Method	**Disfluency**	**Filler**	**Stutter**
Train data	1.55	1.28	0.27
Tag	0.75	0.72	0.03
S-Tag	0.75	0.71	0.04
Symbol	1.10	0.97	0.13

Stutter words are generally susceptible to negative ratings in the "naturalness" and "likability" indices, but they are disfluent phenomena that always occur in the pursuit of "human-like" and thus are something we would like to reproduce. It is necessary to isolate the issue and investigate the impact of "stutter words" as a stand-alone phenomenon that elicits the perception and recall of spontaneous speech.

6.7 Conclusions and Discussions

In this chapter, we proposed an integrated method for text style transfer (TST) and text-to-speech synthesis (TTS) using disfluency annotation, addressing the first and third challenges presented in Sections 1.2.1 and 1.2.3.

Section 6.3.3 introduced filled pauses (FPs) annotation method for TTS as related work. Section 6.3 proposed three types of disfluency annotation: `Symbol`, `Tag`, and `S-TAG` for TST, and `Symbol`, `Tag`, and `Auto` for TTS. We also proposed combinations of disfluency-annotated TST and TTS.

We conducted three experiments to evaluate disfluency annotation by comparing the condition without disfluency annotation (`Plain`): Automatic evaluation for bidirectional style-transferred text, ABX test for TTS in terms of style reproducibility and naturalness, and ABX test for TST + TTS in terms of style reproducibility and naturalness.

Section 6.4 showed that disfluency annotations could improve TST's style controlability and content preservation. `S-Tag` treated all words as disfluency during transfer and indicated the need for an annotation that treats disfluency as an independent token. Section 6.5 showed that using `Tag`, which specifies the location, type, and word of disfluency in detail, could improve the style reproducibility and naturalness in spontaneous speech synthesis compared with using other annotations and not using disfluency annotation. `Auto` and `Symbol`, methods with lower annotation costs than `Tag`, were inferior to `Plain` in terms of style reproducibility and had difficulty automatically rendering disfluencies for fluent text or text with only disfluency positions when there were many types of disfluency. Section 6.6 showed that a combination of `Tag` could improve the style reproducibility in spontaneous speech synthesis compared with `Plain`, and `Symbol`'s naturalness is inferior to all other systems. Additional statistical analysis and discussion revealed that `Symbol`'s TST had shortcomings that were difficult to see from the automatic evaluation and that `Symbol`'s TTS had a shorter duration of disfluency than `Tag`'s.

In this study, we did not conduct a test to investigate the effect of disfluency on recall by creating an actual whole lecture speech, as has been performed in previous studies; we only evaluated impressions of speech units. However, we aim to investigate the effect of perception and recall for the whole lecture speech and to produce a more natural and spontaneous style by annotating other spontaneous behaviors, such as pauses and

nonverbal emotional expressions. In addition, since we did not perform multispeaker learning on the TST side, we would like to introduce speaker labels and the like on the TST side to realize spontaneous speech synthesis that reflects more individuality.

7 Conclusions and General Discussions

7.1 Summary of This Thesis and Contributions

This thesis aims to realize *monolog-oriented spontaneous speech synthesis*—in particular, synthesis that incorporates disfluencies such as filled pauses (FPs)—*directly from fluent written text*. To this end, we advanced a *spoken-text processing* approach that transforms written text into spoken-style text with controllable spontaneity and then synthesizes it with a TTS back-end.

In the Introduction (Section 1.2), we identify three challenges for spontaneous speech generation in monologs, and this thesis addresses them as follows:

- **Problem 1 (written-to-spontaneous mismatch)**. We establish a practical pathway to generate spontaneous speech from written text by introducing a *TST–TTS cascade* framework (Chapters 5 and 6). This framework reduces reliance on manual conversion of written scripts into spoken-style text.

- **Problem 2 (nonparallel TST performance)**. We improve TST under nonparallel conditions by proposing *CycleCVAE+CWS*, which strengthens content preservation while enabling controllable transfer of spontaneous styles (Chapter 5).

- **Problem 3 (inconsistent disfluency handling)**. We introduce a shared *Dis-*

fluency Annotation interface and integrate it into both TST and TTS, enabling consistent representation and realization of disfluencies across modules (Chapter 6).

Across Chapters 2–4, we provide background knowledge and survey related work on spontaneous speech and text style transfer (TST). The technical contributions are organized around two pillars.

Pillar I: Advancing nonparallel text style transfer (TST). In Chapter 5, we propose *CycleCVAE+CWS* to balance content preservation with strong yet natural style transfer. By combining cyclic learning (to learn not only reconstruction but also style transfer) with content word storage (CWS), an explicit mechanism that protects content words, the method enables accurate transfer of spontaneous styles such as disfluencies and dialectal variants under nonparallel supervision. In addition, the approach is computationally efficient compared with LLM-based methods, making it practical as a text-side front-end for speech synthesis.

Pillar II: An integrated framework for spontaneous speech via TST–TTS. In Chapter 6, we integrate the TST module with a TTS back-end through a shared *Disfluency Annotation* method that specifies the positions and types of disfluencies such as FPs and word fragments. This unified pipeline enables text-level control of disfluency placement while improving the acoustic realization of disfluent speech. In our lecture-speech generation experiments, the framework was shown to achieve a more favorable balance between perceived naturalness and spontaneity than conventional baselines.

Taken together, this thesis formalizes a spoken-text processing approach that makes spontaneity controllable at the text level via *CycleCVAE+CWS* and exposes it to TTS

through a shared *Disfluency Annotation* interface, thereby providing a concrete answer to the central question posed in the Introduction: *how to generate spontaneous and natural speech directly from written text.*

7.2 Integration of Findings

At first glance, the two pillars of this thesis—*CycleCVAE+CWS* for nonparallel text style transfer (TST) and the *TST–TTS integration with Disfluency Annotation*—may appear to be separate research strands. In fact, they are tightly coupled by a common end goal: generating spontaneous speech from written text.

Concretely, the TST method in Chapter 5 supplies the text-side mechanism for injecting spontaneity. For example, it can insert FPs and stutter words into otherwise fluent lecture scripts, producing text that better reflects human conversational style while preserving propositional content. The integration in Chapter 6 then realizes these text-level decisions acoustically: the TTS back-end consumes the style-transferred text and renders the specified disfluencies in the waveform.

By combining the two, we obtain a unified *spoken-text processing framework* that handles the full pathway from text-level control to speech-level realization. Unlike conventional TTS pipelines that assume already-spontaneous text as input, our approach creates spontaneous text from written text via TST and then synthesizes it, establishing a new paradigm for end-to-end spontaneity.

This integration yields several concrete benefits:

- **Consistency through cascading.** A shared Disfluency Annotation method ensures that positions and types of disfluencies decided by TST are faithfully realized by TTS, reducing mismatches between text plans and acoustic output.

- **Balanced control.** CycleCVAE+CWS preserves content words by localizing edits to non-content tokens and enables effective control of spontaneous style within the text; downstream, the TTS back-end learns to map these symbolic cues to natural timing and prosody, improving spontaneity.

- **Evaluation alignment.** Beyond MOS, future work can jointly analyze text-side measures (e.g., placement accuracy) and speech-side outcomes (e.g., perceived listener effort), enabling more sensitive evaluation of controllable spontaneity.

- **Modularity and extensibility.** The interface permits future upgrades on either side (e.g., stronger TST controllers or prosody-aware TTS) without redesigning the entire pipeline.

Accordingly, the primary contribution of this thesis goes beyond isolated technical advances: it *positions TST and TTS as an integrated stack* and provides an application-focused foundation for spontaneous speech generation, charting a path from traditional reading-style synthesis to controllable, human-like spontaneity.

7.3 Potential Applications

The proposed *spoken-text processing framework* has significance beyond its academic contributions; it enables a wide range of practical applications. Representative domains include:

1. **Education**

    - **Lecture and textbook augmentation.** Starting from written materials, the framework can generate spontaneous, listener-friendly lecture videos and

classes with virtual teachers, inserting FPs and timing cues that mirror human delivery.

- **Learning outcomes.** Appropriate hesitation and pausing can reduce perceived pace, facilitate chunking, and improve comprehension and recall for complex content. This is particularly valuable for online education and automated e-learning content.
- **Personalization and pacing.** The degree of spontaneity (e.g., FP rate, pause duration) can be adjusted to learner proficiency, enabling accessible variants for different cognitive loads or languages.

2. **Healthcare and assistive communication**

- **Humanlike voice interfaces.** For users with speech impairments, the system can provide alternative voices whose spontaneity better reflects everyday conversational norms, improving social acceptance and comfort.
- **Therapy and practice.** In rehabilitation settings, controllable disfluencies can be used to simulate realistic interlocutors and support graded practice of turn-taking and repair strategies.

3. **Conversational agents and assistants**

- **Naturalistic small talk.** Smart speakers and virtual agents can move beyond mechanical reading-style delivery, inserting brief hesitations, repair, and repetition that increase perceived human-like warmth and relatability.
- **Interaction management.** Disfluency timing can signal floor-holding or uncertainty, improving turn-taking and grounding in multi-turn dialog system.
- **Persona control.** By enabling more granular manipulation of style param-

eters (spontaneity intensity, FP inventory, and repair type), diverse persona designs can be achieved for each application.

To realize these applications across various domains, there exist multiple directions for technological advancement along with challenges that must be addressed, as described in the next section.

7.4 Future Directions

We outline several directions to further develop this work.

7.4.1 Technical developments

- **Integration with large language models (LLMs).** Incorporating LLMs into the TST component could enable more context-sensitive, discourse-aware spontaneous text generation. A practical challenge is latency (fast inference is required). Moreover, because Dall *et al.* [90] demonstrated via perceptual tests that there are indeed *appropriate* and *inappropriate* positions for FP insertion, future systems should explicitly examine whether LLM-based controllers learn such placement constraints, rather than assuming they emerge implicitly.

- **Expanding to more disfluencies and spontaneous phenomena.** Beyond the FPs and word fragments considered here, useful targets include discourse markers, repetitions/restarts, repairs, prolongations, breaths, backchannels, and laughter. Broadening the inventory can improve realism and task fit.

- **Multimodal integration.** Coupling speech with nonverbal channels (e.g., gaze, nods, facial expressions, gestures) can support more human-like timing, floor-

holding, and repair behavior. This will require audio–visual corpora with aligned disfluency/prosody labels and models that coordinate cross-modal cues.

- **Personalized adaptation.** Introduce mechanisms to tailor spontaneity to speaker identity and audience characteristics (e.g., user-adjustable FP rate, duration, and type), ideally with privacy-preserving adaptation (on-device or secure federated learning).

7.4.2 Open research problems

- **Balancing spontaneity and clarity.** Excessive disfluencies can impede comprehension; effective control is required (e.g., selecting domain-appropriate FP, constraining generation frequency, or using comprehension-aware objectives).
- **Multilingual coverage.** Validation should extend beyond Japanese to typologically and culturally diverse languages, where spontaneous phenomena (vocabulary, placement, timing) differ.
- **Ethical considerations.** As synthetic speech becomes more human-like, risks of misuse (e.g., deceptive content or fraud) increase. Establishing usage guidelines, disclosure (e.g., watermarking or audible icons), consent-aware data curation, and bias audits are essential.

These directions are key to opening the next phase of spontaneous speech synthesis research, moving from prototypes toward reliable, controllable, and responsible real-world systems.

Acknowledgments

I would like to express my deepest gratitude to my advisor, Prof. Tomoki Toda of Nagoya University, for his warm guidance throughout the preparation of this research. From shaping the initial ideas to structuring the arguments and revising the manuscript, Prof. Toda provided invaluable insights. Their advice, always offered from perspectives I had not considered, helped clarify the direction of the research and elevate its quality. I am also sincerely grateful for his thoughtful mentorship regarding my career path and for his many forms of support in my student life.

I am especially grateful to Dr. Yusuke Yasuda, formerly a Specially Appointed Assistant Professor of Nagoya University and now at the National Institute of Informatics, for practical and concrete support in the day-to-day progress of this research. In particular, I benefited greatly from assistance with the logical exposition of the proposed model and with building the experimental environment for the listening test evaluations, as well as from numerous comments during manuscript preparation. Beyond technical matters, Dr. Yasuda offered many lessons on research attitude and mindset, for which I am deeply thankful.

I also thank the examination committee members for their insightful comments and constructive suggestions, which substantially deepened this work. I am particularly indebted to Prof. Katsuhiko Toyama of Nagoya University, who supervised me during my undergraduate studies and provided opportunities to cultivate the foundational

skills that ultimately led to this research.

I am grateful to Mr. Yamato Ohtani (now with the National Institute of Information and Communications Technology) for guidance and collaboration during my internship and joint research in the first half of my doctoral program. From problem formulation and data provision to many discussions in search of better methods, the advice grounded in real-world perspectives was invaluable in clarifying the applied significance of this study.

Part of this work was conducted as joint research with Mr. Yuuto Nakata, who joined our laboratory as an intern from National Institute of Technology, Tokuyama College. Their diligent and swift execution of experiments and analyses, together with active discussions, yielded important findings and often prompted me to notice aspects I had overlooked. I extend my sincere thanks.

This research has been supported since the latter half of my doctoral program by THERS Make New Standards Program, a doctoral support project of Nagoya University. I gratefully acknowledge this support.

I also wish to thank all members of my laboratory and our collaborators in the Takeda Laboratory for daily discussions and assistance with experimental data collection. Finally, I am profoundly grateful to my family and friends for their unwavering support of my studies and daily life.

References

[1] K. Maekawa, "Spontaneous speech in the light of linguistics." *Proc. Spring Meet. Acoust. Soc. Jpn.*, vol. 2001, no. 1, pp. 19–22, 2001.

[2] F. Tree and J. E., "Listeners' uses of um and uh in speech comprehension," *Memory & Cognition*, vol. 29, no. 2, pp. 320–326, 2001. [Online]. Available: https://link.springer.com/article/10.3758/BF03194926

[3] J. E. Arnold, M. K. Tanenhaus, R. J. Altmann, and M. Fagnano, "The old and thee, uh, new: Disfluency and reference resolution," *Psychological science*, vol. 15, no. 9, pp. 578–582, 2004.

[4] M. Watanabe, K. Hirose, Y. Den, and N. Minematsu, "Filled pauses as cues to the complexity of upcoming phrases for native and non-native listeners," *Speech Communication*, vol. 50, no. 2, pp. 81–94, 2008.

[5] L. Schettino, A. Origlia, and F. Cutugno, "*Though this be hesitant, yet there is method in 't*: Effects of disfluency patterns in neural speech synthesis for cultural heritage presentations," *Comput. Speech Lang.*, vol. 85, p. 101585, 2024.

[6] É. Székely, G. E. Henter, J. Beskow, and J. Gustafson, "How to train your fillers: uh and um in spontaneous speech synthesis," in *Proc. SSW*, M. Pucher, Ed. ISCA, 2019, pp. 245–250.

[7] H. Guo, S. Zhang, F. K. Soong, L. He, and L. Xie, "Conversational end-to-end TTS for voice agents," in *Proc. SLT*. IEEE, 2021, pp. 403–409.

[8] W. Li, P. Yang, Y. Zhong, Y. Zhou, Z. Wang, Z. Wu, X. Wu, and H. Meng, "Spontaneous style text-to-speech synthesis with controllable spontaneous behaviors based on language models," in *25th Annual Conference of the International Speech Communication Association, Interspeech 2024, Kos, Greece, September 1-5, 2024*, I. Lapidot and S. Gannot, Eds. ISCA, 2024. [Online]. Available: https://doi.org/10.21437/Interspeech.2024-1989

[9] H. Lameris, S. Mehta, G. E. Henter, J. Gustafson, and É. Székely, "Prosody-controllable spontaneous TTS with neural HMMS," in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*. IEEE, 2023, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICASSP49357.2023.10097200

[10] H. Li, X. Zhu, L. Xue, Y. Song, Y. Chen, and L. Xie, "Spontts: Modeling and transferring spontaneous style for TTS," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*. IEEE, 2024, pp. 12 171–12 175. [Online]. Available: https://doi.org/10.1109/ICASSP48485.2024.10445828

[11] J. E. Tree, "The effects of false starts and repetitions on the processing of subsequent words in spontaneous speech," *Journal of Memory and Language*, vol. 34, no. 6, pp. 709–738, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0749596X85710327

[12] H. Maclay and C. E. Osgood, "Hesitation phenomena in spontaneous english speech," *WORD*, vol. 15, no. 1, pp. 19–44, 1959. [Online]. Available: https://doi.org/10.1080/00437956.1959.11659682

[13] Y. Den and M. Watanabe, "Some functions of disfluency in speech communication," *Journal of the Phonetic Society of Japan*, vol. 13, no. 1, pp. 53–64, 2009.

[14] E. A. Schegloff, G. Jefferson, and H. Sacks, "The preference for self-correction in the organization of repair in conversation," *Language*, vol. 53, no. 2, pp. 361–382, 1977. [Online]. Available: http://www.jstor.org/stable/413107

[15] W. J. Levelt, "Monitoring and self-repair in speech," *Cognition*, vol. 14, no. 1, pp. 41–104, 1983.

[16] E. R. Blacfkmer and J. L. Mitton, "Theories of monitoring and the timing of repairs in spontaneous speech," *Cognition*, vol. 39, no. 3, pp. 173–194, 1991.

[17] H. H. Clark and T. Wasow, "Repeating words in spontaneous speech," *Cognitive Psychology*, vol. 37, no. 3, pp. 201–242, 1998.

[18] S. R. Rochester, "The significance of pauses in spontaneous speech," *Journal of Psycholinguistic Research*, vol. 2, no. 1, pp. 51–81, 1973.

[19] D. Schiffrin, *Discourse Markers*, ser. Studies in Interactional Sociolinguistics. Cambridge University Press, 1987.

[20] D. T. Miller, "The effect of dialect and ethnicity on communicator effectiveness," *Speech Monographs*, vol. 42, no. 1, pp. 69–74, 1975.

[21] N. Coupland, *Style: Language Variation and Identity*, ser. Key Topics in Sociolinguistics. Cambridge University Press, 2007. [Online]. Available: https://books.google.co.jp/books?id=oJE462b0kv4C

[22] E. K. Acton, "On gender differences in the distribution of um and uh," *University of Pennsylvania Working Papers in Linguistics*, vol. 17, no. 2, pp. 1–9, 2011, selected Papers from NWAV 39. [Online]. Available: https://repository.upenn.edu/pwpl/vol17/iss2/2

[23] G. Tottie, "Uh and um as sociolinguistic markers in british english," *International Journal of Corpus Linguistics*, vol. 16, no. 2, pp. 173–197, 2011. [Online]. Available: https://www.jbe-platform.com/content/journals/10.1075/ijcl.16.2.02tot

[24] W. Chafe, *Discourse, Consciousness, and Time: The Flow and Displacement of Conscious Experience in Speaking and Writing*, ser. Discourse, Consciousness, and Time: The Flow and Displacement of Conscious Experience in Speaking and Writing. University of Chicago Press, 1994. [Online]. Available: https://books.google.tt/books?id=j-UFqNz8D28C

[25] D. Biber, S. Johansson, G. Leech, S. Conrad, and E. Finegan, *Grammar of Spoken and Written English*. John Benjamins Publishing Company, 2021. [Online]. Available: https://books.google.co.jp/books?id=qSlHEAAAQBAJ

[26] F. Goldman-Eisler, "Pauses, clauses, sentences," *Language and Speech*, vol. 15, no. 2, pp. 103–113, 1972.

[27] B. Butterworth, "Hesitation and semantic planning in speech," *Journal of Psycholinguistic Research*, vol. 4, no. 1, pp. 75–87, 1975.

[28] R. Eklund, "Prolongations: A dark horse in the disfluency stable," in *ISCA Tutorial and Research Workshop (ITRW) on Disfluency in Spontaneous Speech, DiSS 2001, Edinburgh, Scotland, UK, August 29-31, 2001*. ISCA, 2001, pp. 5–8. [Online]. Available: https://www.isca-archive.org/diss_2001/eklund01_diss.html

[29] P. Howell and K. Kadi-Hanifi, "Comparison of prosodic properties between read and spontaneous speech material," *Speech Communication*, vol. 10, no. 2, pp. 163–169, 1991.

[30] K. R. Scherer, "Vocal communication of emotion: A review of research paradigms," *Speech Communication*, vol. 40, no. 1–2, pp. 227–256, 2003.

[31] E. Grabe, "Variation adds to prosodic typology," in *Speech Prosody 2002*, 2002, pp. 127–132.

[32] P. Warren, "Issues in the study of intonation in language varieties," *Language and Speech*, vol. 48, no. 4, pp. 345–358, 2005.

[33] J. Ehret, A. Bönsch, L. Aspöck, C. T. Röhr, S. Baumann, M. Grice, J. Fels, and T. W. Kuhlen, "Do prosody and embodiment influence the perceived naturalness of conversational agents' speech?" *ACM Transactions on Applied Perception*, vol. 18, no. 4, 2021.

[34] E. Jacewicz, R. A. Fox, and L. Wei, "Between-speaker and within-speaker variation in speech tempo of american english," *The Journal of the Acoustical Society of America*, vol. 128, no. 2, pp. 839–850, 2010.

[35] M. Breen, E. Fedorenko, M. Wagner, and E. Gibson, "Acoustic correlates of information structure," *Language and Cognitive Processes*, vol. 25, no. 7–9, pp. 1044–1098, 2010.

[36] E. Szekely, J. Higginbotham, and F. Possemato, "Voice and choice: Investigating the role of prosodic variation in request compliance and perceived politeness using conversational TTS," in *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, T. Kawahara, V. Demberg, S. Ultes, K. Inoue, S. Mehri, D. Howcroft, and K. Komatani, Eds. Kyoto, Japan: Association for Computational Linguistics, Sep. 2024, pp. 466–476. [Online]. Available: https://aclanthology.org/2024.sigdial-1.40/

[37] P. Glenn, *Laughter in Interaction*, ser. Studies in Interactional Sociolinguistics. Cambridge University Press, 2003.

[38] E. M. Hoey, "Sighing in interaction: Somatic, semiotic, and social," *Research on Language and Social Interaction*, vol. 47, no. 2, pp. 175–200, 2014. [Online]. Available: https://doi.org/10.1080/08351813.2014.900229

[39] F. Torreira, S. Bögels, and S. C. Levinson, "Breathing for answering: the time course of response planning in conversation," *Frontiers in Psychology*, vol. 6, p. 284, 2015. [Online]. Available: https://doi.org/10.3389/fpsyg.2015.00284

[40] M. Wlodarczak and M. Heldner, "Respiratory belts and whistles: A preliminary study of breathing acoustics for turn-taking," in *17th Annual Conference of the International Speech Communication Association, Interspeech 2016, San Francisco, CA, USA, September 8-12, 2016*, N. Morgan, Ed. ISCA, 2016, pp. 510–514. [Online]. Available: https://doi.org/10.21437/Interspeech.2016-344

[41] J. Trouvain, R. Werner, and B. Möbius, "An acoustic analysis of inbreath noises in read and spontaneous speech," in *Speech Prosody 2020*, 2020, pp. 789–793.

[42] Y. V. H., "On getting a word in edgewise," *Chicago Linguistics Society, 6th Meeting, 1970*, pp. 567–578, 1970. [Online]. Available: https://cir.nii.ac.jp/crid/1572824500867512320

[43] W. Nigel and T. Wataru, "Prosodic features which cue back-channel responses in english and japanese," *Journal of Pragmatics*, vol. 32, no. 8, pp. 1177–1207, 07 2000. [Online]. Available: https://cir.nii.ac.jp/crid/1361699994520775296

[44] A. Gravano and J. Hirschberg, "Turn-taking cues in task-oriented dialogue," *Comput. Speech Lang.*, vol. 25, no. 3, pp. 601–634, 2011. [Online]. Available: https://doi.org/10.1016/j.csl.2010.10.003

[45] S. H. Fraundorf and D. G. Watson, "The disfluent discourse: Effects of filled pauses on recall," *Journal of Memory and Language*, vol. 65, no. 2, pp. 161–175, 2011.

[46] B. Muhlack, M. Elmers, H. Drenhaus, J. Trouvain, M. van Os, R. Werner, M. Ryzhova, and B. Möbius, "Revisiting recall effects of filler particles in german and english," in *Proc. Interspeech 2021*, H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, and P. Motlícek, Eds. ISCA, 2021, pp. 3979–3983.

[47] T. NAGURA, "Hesitations (discourse markers) in japanese," 世界の日本語教育. 日本語教育論集, vol. 7, pp. 201–218, 06 1997. [Online]. Available: https://cir.nii.ac.jp/crid/1390572174835946368

[48] E. A. Schegloff, "Some other "uh(m)"s 1," *Discourse Processes*, vol. 47, no. 2, pp. 130–174, 2010. [Online]. Available: https://doi.org/10.1080/01638530903223380

[49] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous Speech Corpus of Japanese," in *Proc. LREC*, 2000, pp. 947–952.

[50] J. Godfrey, E. Holliman, and J. McDaniel, "Switchboard: telephone speech corpus for research and development . acoustics,," in *IEEE International Conference on Speech, and Signal Processing, ICASSP-92*, vol. 1, 1992, pp. 517–520.

[51] Y. Ferstl and R. McDonnell, "Iva: Investigating the use of recurrent motion modelling for speech gesture generation," in *IVA '18 Proceedings of the 18th International Conference on Intelligent Virtual Agents*, Nov 2018. [Online]. Available: https://trinityspeechgesture.scss.tcd.ie

[52] Z. Yang, Y. Chen, L. Luo, R. Yang, L. Ye, G. Cheng, J. Xu, Y. Jin, Q. Zhang, P. Zhang, L. Xie, and Y. Yan, "Open source magicdata-ramc: A rich annotated mandarin conversational(ramc) speech dataset," in *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022, Incheon, Korea, September 18-22, 2022*, H. Ko and J. H. L. Hansen, Eds. ISCA, 2022, pp. 1736–1740. [Online]. Available: https://doi.org/10.21437/Interspeech.2022-729

[53] M. A. Pitt, L. Dilley, K. Johnson, S. Kiesling, W. Raymond, E. Hume, and E. Fosler-Lussier, "Buckeye corpus of conversational speech (2nd release)," http://www.buckeyecorpus.osu.edu, Columbus, OH, 2007, distributor.

[54] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus: A pre-announcement," in *Machine Learning for Multimodal Interaction (MLMI 2005)*, ser. Lecture Notes in Computer Science, S. Renals and S. Bengio, Eds., vol. 3869. Springer, 2006, pp. 28–39.

[55] A. Taylor, M. Marcus, and B. Santorini, *The Penn Treebank: An Overview*. Dordrecht: Springer Netherlands, 2003, pp. 5–22. [Online]. Available: https://doi.org/10.1007/978-94-010-0201-1_1

[56] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot International*, vol. 5, no. 9/10, pp. 341–345, 2001.

[57] F. Eyben, M. Wöllmer, and B. Schuller, "opensmile – the munich versatile and fast open-source audio feature extractor," in *Proceedings of the 18th ACM International Conference on Multimedia (MM '10)*. Florence, Italy: ACM, 2010, pp. 1459–1462.

[58] F. Eyben, F. Weninger, F. Groß, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM International Conference on Multimedia (MM '13)*. Barcelona, Spain: ACM, 2013, pp. 835–838.

[59] M. Ding, "A systematic review on the development of speech synthesis," in *8th International Conference on Computer and Communication Systems, ICCCS 2023, Guangzhou, China, April 21-23, 2023*. IEEE, 2023, pp. 28–33. [Online]. Available: https://doi.org/10.1109/ICCCS57501.2023.10150729

[60] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1, 1996, pp. 373–376 vol. 1.

[61] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden markov models," *Proc. IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013. [Online]. Available: https://doi.org/10.1109/JPROC.2013.2251852

[62] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7962–7966.

[63] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *The 9th ISCA Speech Synthesis Workshop, SSW 2016, Sunnyvale, CA, USA, September 13-15, 2016*, A. W. Black, Ed. ISCA, 2016, p. 125. [Online]. Available: https://www.isca-archive.org/ssw_2016/vandenoord16_ssw.html

[64] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NeurIPS*, 2014, pp. 3104–3112.

[65] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *18th Annual Conference of the International Speech Communication Association, Interspeech 2017, Stockholm, Sweden, August 20-24, 2017*, F. Lacerda, Ed. ISCA, 2017, pp. 4006–4010. [Online]. Available: https://doi.org/10.21437/Interspeech.2017-1452

[66] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *Proc. ICASSP*. IEEE, 2018, pp. 4779–4783.

[67] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 6706–6713. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33016706

[68] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 3165–3174. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/f63f65b503e22cb970527f23c9ad7db1-Abstract.html

[69] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=piLPYqxtWuA

[70] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. ICML*, vol. 139, 2021, pp. 5530–5540.

[71] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NeurIPS*,

2014, pp. 2672–2680.

[72] R. Yamamoto, E. Song, and J. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, 2020, pp. 6199–6203. [Online]. Available: https://doi.org/10.1109/ICASSP40776.2020.9053795

[73] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/c5d736809766d46260d816d8dbc9eb44-Abstract.html

[74] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Commun.*, vol. 50, no. 5, pp. 434–451, 2008. [Online]. Available: https://doi.org/10.1016/j.specom.2008.01.002

[75] K. Ito and L. Johnson, "The LJ Speech Dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[76] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," in *20th Annual Conference of the International Speech Communication Association, Interspeech 2019, Graz, Austria, September 15-19, 2019*, G. Kubin

and Z. Kacic, Eds. ISCA, 2019, pp. 1526–1530. [Online]. Available: https://doi.org/10.21437/Interspeech.2019-2441

[77] R. Sonobe, S. Takamichi, and H. Saruwatari, "JSUT corpus: free large-scale japanese speech corpus for end-to-end speech synthesis," *CoRR*, vol. abs/1711.00354, 2017. [Online]. Available: http://arxiv.org/abs/1711.00354

[78] J. Fong, P. O. Gallegos, Z. Hodari, and S. King, "Investigating the robustness of sequence-to-sequence text-to-speech models to imperfectly-transcribed training data," in *20th Annual Conference of the International Speech Communication Association, Interspeech 2019, Graz, Austria, September 15-19, 2019*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 1546–1550. [Online]. Available: https://doi.org/10.21437/Interspeech.2019-1824

[79] X. Zhang, I. Vallés-Pérez, A. Stolcke, C. Yu, J. Droppo, O. Shonibare, R. Barra-Chicote, and V. Ravichandran, "Stutter-tts: Controlled synthesis and improved recognition of stuttered speech," 2022. [Online]. Available: https://www.amazon.science/publications/stutter-tts-controlled-synthesis-and-improved-recognition-of-stuttered-speech

[80] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 5167–5176. [Online]. Available: http://proceedings.mlr.press/v80/wang18h.html

[81] T. Xie, Y. Rong, P. Zhang, W. Wang, and L. Liu, "Towards controllable speech synthesis in the era of large language models: A survey," *arXiv preprint arXiv:2412.06602*, 2024.

[82] ITU-T, "Recommendation itu-t p.800: Methods for subjective determination of transmission quality," International Telecommunication Union, Tech. Rep., Aug. 1996. [Online]. Available: https://www.itu.int/rec/t-rec-p.800-199608-i

[83] A. Kirkland, S. Mehta, H. Lameris, G. E. Henter, É. Székely, and J. Gustafson, "Stuck in the MOS pit: A critical analysis of MOS test methodology in TTS evaluation," in *12th ISCA Speech Synthesis Workshop, SSW 2023, Grenoble, France, August 26-28, 2023*, G. Bailly, T. Hueber, D. Lolive, N. Obin, and O. Perrotin, Eds. ISCA, 2023, pp. 41–47. [Online]. Available: https://doi.org/10.21437/SSW.2023-7

[84] Y. Yasuda and T. Toda, "Limits of mos-based subjective evaluation and new opportunity of large-scale preference test," *THE JOURNAL OF THE ACOUSTICAL SOCIETY OF JAPAN*, vol. 80, no. 7, pp. 393–400, 2024.

[85] É. Székely, G. E. Henter, J. Beskow, and J. Gustafson, "Breathing and speech planning in spontaneous speech synthesis," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, 2020, pp. 7649–7653. [Online]. Available: https://doi.org/10.1109/ICASSP40776.2020.9054107

[86] E. Ekstedt, S. Wang, Éva Székely, J. Gustafson, and G. Skantze, "Automatic evaluation of turn-taking cues in conversational speech synthesis," in *Interspeech 2023*, 2023, pp. 5481–5485.

[87] S. Wang, É. Székely, and J. Gustafson, "Contextual interactive evaluation of TTS models in dialogue systems," in *25th Annual Conference of the International Speech Communication Association, Interspeech 2024, Kos, Greece, September 1-5, 2024*, I. Lapidot and S. Gannot, Eds. ISCA, 2024. [Online]. Available: https://doi.org/10.21437/Interspeech.2024-1008

[88] J. Adell, A. Bonafonte, and D. E. Mancebo, "Synthesis of filled pauses based on a disfluent speech model," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2010, 14-19 March 2010, Sheraton Dallas Hotel, Dallas, Texas, USA*. IEEE, 2010, pp. 4810–4813. [Online]. Available: https://doi.org/10.1109/ICASSP.2010.5495136

[89] J. S. Andersson, J. Yamagishi, and R. A. J. Clark, "Utilising spontaneous conversational speech in hmm-based speech synthesis," in *The Seventh ISCA Tutorial and Research Workshop on Speech Synthesis, SSW 2010, Kyoto, Japan, September 22-24, 2010*, Y. Sagisaka and K. Tokuda, Eds. ISCA, 2010, pp. 173–178. [Online]. Available: https://www.isca-archive.org/ssw_2010/andersson10_ssw.html

[90] R. Dall, M. Tomalin, M. Wester, W. J. Byrne, and S. King, "Investigating automatic & human filled pause insertion for speech synthesis," in *15th Annual Conference of the International Speech Communication Association, INTERSPEECH 2014, Singapore, September 14-18, 2014*, H. Li, H. M. Meng, B. Ma, E. Chng, and L. Xie, Eds. ISCA, 2014, pp. 51–55. [Online]. Available: https://doi.org/10.21437/Interspeech.2014-11

[91] É. Székely, G. E. Henter, J. Beskow, and J. Gustafson, "Spontaneous conversational speech synthesis from found data," in *20th Annual Conference of*

the International Speech Communication Association, Interspeech 2019, Graz, Austria, September 15-19, 2019, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 4435–4439. [Online]. Available: https://doi.org/10.21437/Interspeech.2019-2836

[92] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[93] R. Qader, G. Lecorvé, D. Lolive, and P. Sébillot, "Disfluency Insertion for Spontaneous TTS: Formalization and Proof of Concept," in *SLSP 2018 - 6th International Conference on Statistical Language and Speech Processing*. Mons, Belgium: Springer, Oct. 2018, pp. 1–12. [Online]. Available: https://inria.hal.science/hal-01840798

[94] W. Li, S. Lei, Q. Huang, Y. Zhou, Z. Wu, S. Kang, and H. Meng, "Towards spontaneous style modeling with semi-supervised pre-training for conversational text-to-speech synthesis," in *Proc. Interspeech*, N. Harte, J. Carson-Berndsen, and G. Jones, Eds. ISCA, 2023, pp. 3377–3381.

[95] M. Wester, M. P. Aylett, M. Tomalin, and R. Dall, "Artificial personality and disfluency," in *Proc. INTERSPEECH*. ISCA, 2015, pp. 3365–3369.

[96] J. Gustafson, J. Beskow, and É. Székely, "Personality in the mix - investigating the contribution of fillers and speaking style to the perception of spontaneous speech synthesis," in *Proc. Speech Synthesis Workshop, SSW*, G. Németh, Ed. ISCA, 2021, pp. 48–53.

[97] A. Kirkland, H. Lameris, É. Székely, and J. Gustafson, "Where's the uh, hesitation? the interplay between filled pause location, speech rate and fundamental

frequency in perception of confidence," in *Proc. Interspeech*, H. Ko and J. H. L. Hansen, Eds. ISCA, 2022, pp. 4990–4994.

[98] R. Dall, M. Wester, and M. Corley, "The effect of filled pauses and speaking rate on speech comprehension in natural, vocoded and synthetic speech," in *15th Annual Conference of the International Speech Communication Association, INTERSPEECH 2014, Singapore, September 14-18, 2014*, H. Li, H. M. Meng, B. Ma, E. Chng, and L. Xie, Eds. ISCA, 2014, pp. 56–60. [Online]. Available: https://doi.org/10.21437/Interspeech.2014-12

[99] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*. IEEE, 2019, pp. 3617–3621. [Online]. Available: https://doi.org/10.1109/ICASSP.2019.8683143

[100] Y. Matsunaga, T. Saeki, S. Takamichi, and H. Saruwatari, "Empirical study incorporating linguistic knowledge on filled pauses for personalized spontaneous speech synthesis," in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2022, pp. 1898–1903.

[101] T. Shen, T. Lei, R. Barzilay, and T. S. Jaakkola, "Style transfer from non-parallel text by cross-alignment," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 6830–6841. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/2d2c8394e31101a261abf1784302bf75-Abstract.html

[102] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *CVPR*, 2016, pp. 2414–2423.

[103] M. Toshevska and S. Gievska, "A review of text style transfer using deep learning," *IEEE Trans. Artif. Intell.*, p. 1, 2021.

[104] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proc. ACL*, 2002, pp. 311–318.

[105] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proc. EMNLP-IJCNLP 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 3980–3990.

[106] H. Jhamtani, V. Gangal, E. H. Hovy, and E. Nyberg, "Shakespearizing modern language using copy-enriched sequence-to-sequence models," in *Proc. the Workshop on Stylistic Variation*, 2017, pp. 10–19.

[107] K. Carlson, A. Riddell, and D. Rockmore, "Evaluating prose style transfer with the bible," *Royal Society Open Science*, vol. 5, no. 10, 2018.

[108] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. NeurIPS*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 2692–2700.

[109] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, Eds., 2015, pp. 1412–1421.

[110] J. Li, R. Jia, H. He, and P. Liang, "Delete, retrieve, generate: a simple approach to sentiment and style transfer," in *Proc. NAACL-HLT Volume 1 (Long Papers)*, M. A. Walker, H. Ji, and A. Stent, Eds., 2018, pp. 1865–1874.

[111] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Toward controlled generation of text," in *Proc. ICML*, D. Precup and Y. W. Teh, Eds., vol. 70, 2017, pp. 1587–1596.

[112] S. Prabhumoye, Y. Tsvetkov, R. Salakhutdinov, and A. W. Black, "Style transfer through back-translation," in *Proc. ACL*, I. Gurevych and Y. Miyao, Eds., 2018, pp. 866–876.

[113] V. John, L. Mou, H. Bahuleyan, and O. Vechtomova, "Disentangled representation learning for non-parallel text style transfer," in *Proc. ACL*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds., 2019, pp. 424–434.

[114] J. J. Zhao, Y. Kim, K. Zhang, A. M. Rush, and Y. LeCun, "Adversarially regularized autoencoders," in *Proc. ICML*, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 5897–5906.

[115] Y. Tian, Z. Hu, and Z. Yu, "Structured content preservation for unsupervised text style transfer," *CoRR*, vol. abs/1810.06526, 2018.

[116] Y. Huang, W. Zhu, D. Xiong, Y. Zhang, C. Hu, and F. Xu, "Cycle-consistent adversarial autoencoders for unsupervised text style transfer," in *Proc. COLING*, 2020, pp. 2213–2223.

[117] L. Laugier, J. Pavlopoulos, J. Sorensen, and L. Dixon, "Civil rephrases of toxic texts with self-supervised transformers," in *Proc. EACL*, P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds., 2021, pp. 1442–1461.

[118] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2020.

[119] P. Xu, J. C. K. Cheung, and Y. Cao, "On variational learning of controllable representations for text without supervision," in *Proc. ICML*, vol. 119, 2020, pp. 10 534–10 543.

[120] F. Locatello, S. Bauer, M. Lucic, G. Rätsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Reproducibility in Machine Learning, ICLR 2019 Workshop, New Orleans, Louisiana, United States, May 6, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=Byg6VhUp8V

[121] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023.

[122] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Proc. NeurIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.

[123] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Proc.*

NeurIPS, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022.

[124] OpenAI, "GPT-4 technical report," *CoRR*, vol. abs/2303.08774, 2023.

[125] E. Reif, D. Ippolito, A. Yuan, A. Coenen, C. Callison-Burch, and J. Wei, "A recipe for arbitrary text style transfer with large language models," in *Proc. ACL*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., 2022, pp. 837–848.

[126] M. Suzgun, L. Melas-Kyriazi, and D. Jurafsky, "Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models," in *Proc. EMNLP*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., 2022, pp. 2195–2222.

[127] Q. Liu, J. Qin, W. Ye, H. Mou, Y. He, and K. Wang, "Adaptive prompt routing for arbitrary text style transfer with pre-trained language models," in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, M. J. Wooldridge, J. G. Dy, and S. Natarajan, Eds. AAAI Press, 2024, pp. 18 689–18 697. [Online]. Available: https://doi.org/10.1609/aaai.v38i17.29832

[128] N. Wu, Y. Hu, L. Chen, and Z. Ling, "Anchored monotonic alignment and representation substitution for rare spontaneous behaviors in spontaneous speech synthesis," in *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*. IEEE, 2025, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICASSP49660.2025.10887968

[129] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[130] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5998–6008.

[131] V. Zayats, M. Ostendorf, and H. Hajishirzi, "Disfluency detection using a bidirectional LSTM," in *17th Annual Conference of the International Speech Communication Association, Interspeech 2016, San Francisco, CA, USA, September 8-12, 2016*, N. Morgan, Ed. ISCA, 2016, pp. 2523–2527. [Online]. Available: https://doi.org/10.21437/Interspeech.2016-1247

[132] S. Wang, W. Che, Y. Zhang, M. Zhang, and T. Liu, "Transition-based disfluency detection using LSTMs," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, M. Palmer, R. Hwa, and S. Riedel, Eds. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 2785–2794. [Online]. Available: https://aclanthology.org/D17-1296/

[133] P. J. Lou and M. Johnson, "Improving disfluency detection by self-training a self-attentive model," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 3754–3763. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.346

[134] M. Ihori, N. Makishima, T. Tanaka, A. Takashima, S. Orihashi, and R. Masumura, "Zero-shot joint modeling of multiple spoken-text-style conversion tasks using switching tokens," in *Proc. Interspeech*, 2021, pp. 776–780.

[135] F. Wang, W. Chen, Z. Yang, Q. Dong, S. Xu, and B. Xu, "Semi-supervised disfluency detection," in *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, E. M. Bender, L. Derczynski, and P. Isabelle, Eds. Association for Computational Linguistics, 2018, pp. 3529–3538. [Online]. Available: https://aclanthology.org/C18-1299/

[136] J. Yang, D. Yang, and Z. Ma, "Planning and generating natural and diverse disfluent texts as augmentation for disfluency detection," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 1450–1460. [Online]. Available: https://aclanthology.org/2020.emnlp-main.113/

[137] B. Marie, "Disfluency generation for more robust dialogue systems," in *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, A. Rogers, J. L. Boyd-Graber, and N. Okazaki, Eds. Association for Computational Linguistics, 2023, pp. 11 479–11 488. [Online]. Available: https://doi.org/10.18653/v1/2023.findings-acl.728

[138] A. Dimakis, J. Pavlopoulos, and A. Anastasopoulos, "Dialect normalization using large language models and morphological rules," in *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds.

Association for Computational Linguistics, 2025, pp. 23 696–23 714. [Online]. Available: https://aclanthology.org/2025.findings-acl.1215/

[139] "Cojads." [Online]. Available: https://www2.ninjal.ac.jp/cojads/index.html

[140] K. Lasek, M. Ptaszynski, and F. Masui, "Towards japanese dialect-aware chatbot: Adapting nlp models for japanese dialect variation," in *Proceedings of the 9th Linguistic and Cognitive Approaches to Dialog Agents Workshop*, ser. CEUR Workshop Proceedings, vol. 3862. CEUR-WS.org, 2024, pp. 30–44.

[141] D. Rao, D. Yarowsky, A. Shreevats, and M. Gupta, "Classifying latent user attributes in twitter," in *Proc. 2nd International Workshop on SMUC*, 2010, pp. 37–44.

[142] H. A. Schwartz, J. C. Eichstaedt, M. L. Kern, L. Dziurzynski, S. M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M. E. P. Seligman, and L. H. Ungar, "Personality, gender, and age in the language of social media: The open-vocabulary approach," *PLOS ONE*, vol. 8, no. 9, pp. 1–16, 09 2013.

[143] R. L. Boyd, A. Ashokkumar, S. Seraj, and J. W. Pennebaker, "The development and psychometric properties of liwc-22," *Austin, TX: University of Texas at Austin*, pp. 1–47, 2022.

[144] M. A. Walker, G. I. Lin, and J. Sawyer, "An annotated corpus of film dialogue for learning and characterizing character style," in *Proc. LREC*, N. Calzolari, K. Choukri, T. Declerck, M. U. Dogan, B. Maegaard, J. Mariani, J. Odijk, and S. Piperidis, Eds., 2012, pp. 1373–1378.

[145] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. ICLR*, Y. Bengio and Y. LeCun, Eds., 2014.

[146] G. Lample, S. Subramanian, E. M. Smith, L. Denoyer, M. Ranzato, and Y. Boureau, "Multiple-attribute text rewriting," in *Proc. ICLR*, 2019.

[147] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. NeurIPS*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 3581–3589.

[148] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997.

[149] P. L. Tobing, Y. Wu, T. Hayashi, K. Kobayashi, and T. Toda, "Non-parallel voice conversion with cyclic variational autoencoder," in *Proc. Interspeech 2019*, G. Kubin and Z. Kacic, Eds., 2019, pp. 674–678.

[150] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to Japanese morphological analysis," in *Proc. EMNLP*, Jul. 2004, pp. 230–237.

[151] H. Kevin, "An introduction to the Kansai dialect corpus," *Journal of Policy Studies*, no. 41, pp. 157–164, 2012.

[152] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP*, 2014, pp. 1746–1751.

[153] Z. Fu, X. Tan, N. Peng, D. Zhao, and R. Yan, "Style transfer in text: Exploration and evaluation," in *Proc. the Thirty-Second AAAI Conference on Artificial Intelligence*, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018, pp. 663–670.

[154] N. Dai, J. Liang, X. Qiu, and X. Huang, "Style transformer: Unpaired text style transfer without disentangled latent representation," in *Proc. ACL*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds., vol. 1, 2019, pp. 5997–6007.

[155] Y. Yasuda and T. Toda, "Text-to-speech synthesis based on latent variable conversion using diffusion probabilistic model and variational autoencoder," in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*. IEEE, 2023, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICASSP49357.2023.10094298

[156] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," in *NeurIPS*, 2020.

[157] J. Feng, Y. Yasuda, and T. Toda, "Exploring the robustness of text-to-speech synthesis based on diffusion probabilistic models to heavily noisy transcriptions," in *Interspeech 2024*, 2024, pp. 4408–4412.

[158] D. Yoshioka, Y. Yasuda, and T. Toda, "Nonparallel spoken-text-style transfer for linguistic expression control in speech generation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 33, pp. 333–346, 2025.

[159] T. Hayashi, R. Yamamoto, T. Yoshimura, P. Wu, J. Shi, T. Saeki, Y. Ju, Y. Yasuda, S. Takamichi, and S. Watanabe, "ESPnet2-TTS: Extending the edge of TTS research," *arXiv preprint arXiv:2110.07840*, 2021.

[160] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[161] T. Raitio, J. Kane, T. Drugman, and C. Gobl, "Hmm-based synthesis of creaky voice," in *Interspeech 2013*, 2013, pp. 2316–2320.

[162] J. Cong, S. Yang, N. Hu, G. Li, L. Xie, and D. Su, "Controllable context-aware conversational speech synthesis," in *Proc. Interspeech 2021*, H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, and P. Motlícek, Eds. ISCA, 2021, pp. 4658–4662.

[163] R. Love, C. Dembry, A. Hardie, V. Brezina, and T. McEnery, "The spoken bnc2014: Designing and building a spoken corpus of everyday conversations," *International Journal of Corpus Linguistics*, vol. 22, no. 3, pp. 319–344, 2017.

[164] M. Kirjavainen, L. Crible, and K. Beeching, "Can filled pauses be represented as linguistic items? investigating the effect of exposure on the perception and production of um," *Language and Speech*, vol. 65, no. 2, pp. 263–289, 2022.

[165] M. Watanabe, Y. Den, K. Hirose, and N. Minematsu, "The effects of filled pauses on native and non-native listeners^2 speech processing," in *ISCA Tutorial and Research Workshop (ITRW) on Disfluency in Spontaneous Speech, DiSS 2005, Aix-en-Provence, France, September 10-12, 2005*, J. Véronis and E. Campione, Eds. ISCA, 2005, pp. 169–172. [Online]. Available: https://www.isca-archive.org/diss_2005/watanabe05_diss.html

List of Publications

Journal Papers

1. D. Yoshioka, Y. Yasuda, T. Toda, "Nonparallel Spoken-Text-Style Transfer for Linguistic Expression Control in Speech Generation," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 33, pp. 333–346, 2025.

2. D. Yoshioka, Y. Nakata, Y. Yasuda, T. Toda, "Text-to-speech synthesis, Text style transfer, Spontaneous speech, Disfluency," APSIPA Transactions on Signal and Information Processing, Accepted, 2025.

International Conferences

1. D. Yoshioka, Y. Yasuda, N. Matsunaga, Y. Ohtani, T. Toda, "Spoken-Text-Style Transfer with Conditional Variational Autoencoder and Content Word Storage," Proc. Interspeech, 2022, pp. 4576–4580.

2. Y. Nakata, D. Yoshioka, W.-C. Huang, T. Toda, "Disfluency disentanglement enhancement in spoken-text-style transfer for spontaneous speech synthesis," Proc. APSIPA ASC, 2025, pp. 1098–1103.

Domestic Conferences

1. 吉岡 大貴, 安田 裕介, 松永 悟行, 大谷 大和, 戸田 智基, ”注意機構付き VAE を用いたテキスト発話スタイル変換の改良,” 日本音響学会講演論文集, 1-8-16, pp. 1583-1584, Sep. 2022.

2. 吉岡 大貴, 安田 裕介, 松永 悟行, 大谷 大和, 戸田 智基, ”内容語保存機構を備えた変分自己符号化器に基づくテキスト発話スタイル変換,” 情報処理研報, Vol. 2022-SLP-144, No. 8, pp. 1-6, Nov. 2022.

3. 吉岡 大貴, 安田 裕介, 松永 悟行, 大谷 大和, 戸田 智基, ”サイクル学習を用いた注意機構付き VAE によるテキスト発話スタイル変換,” 日本音響学会講演論文集, 2-3Q-12, pp. 911-912, Mar. 2023.

4. 吉岡 大貴, 安田 裕介, 戸田 智基, ”注意機構付き VAE を用いたテキスト発話スタイル変換における少量パラレルデータの活用,” 日本音響学会講演論文集, 2-Q-31, pp. 1249-1250, Sep. 2023.

5. 吉岡 大貴，安田 裕介，戸田 智基, ”テキストスタイル変換を用いた話し言葉音声合成,” 日本音響学会講演論文集, 1-Q-28, pp. 903-904, Mar. 2024.

6. 中田 優翔, 吉岡 大貴, ホワン ウェンチン, 戸田 智基, ”話し言葉音声合成のためのテキスト発話スタイル変換の改良, ” 情報処理研報, Vol. 2024-SLP-154, No. 6, pp. 1-6, Dec. 2024.

Awards

1. Corporate Award (Yahoo! JAPAN), Information Processing Society of Japan , SIG-SLP, Nov., 2022