APSIPA Transactions on Signal and Information Processing, 2025, 14, e26

Original Paper

Text- and Speech-style Control for Lecture Speech Generation Focusing on Disfluency

Daiki Yoshioka[1*], Yuuto Nakata[2], Yusuke Yasuda[1] and Tomoki Toda[1]

[1] *The Graduate School of Informatics, Nagoya University, Aichi, Japan*
[2] *National Institute of Technology, Tokuyama College, Yamaguchi, Japan*

ABSTRACT

In this paper, we propose text style transfer (TST) and text-to-speech synthesis (TTS) using disfluency annotation for the application of "spontaneous speech synthesis using the written text." TTS technology has progressed significantly, achieving human-like naturalness in reading-style speech generation. However, it is still developing when it comes to producing more spontaneous human-like speech. Moreover, for existing spontaneous speech synthesizers, it is assumed that the input text contains spontaneous parts such as disfluencies. Therefore, we aim to synthesize spontaneous speech with disfluency on the basis of written materials without disfluent parts. Specifically, we train the TST and TTS systems for lecture speech generation by tagging disfluencies with special symbols or converting disfluencies into special symbols to enhance each model's linguistic and acoustic control over disfluencies. We combine the TST and TTS systems using disfluency annotation to create a lecture speech generation system and demonstrate the effectiveness of our method by comparing the results of objective

*Corresponding author: Daiki Yoshioka, yoshioka.daiki@g.sp.m.is.nagoya-u.ac.jp. This work was supported by JST SPRING, through "THERS Make New Standards Program for the Next Generation Researchers" under Grant Number JPMJSP2125. This work was

Received 04 February 2025; revised 19 June 2025; accepted 04 August 2025
ISSN 2048-7703; DOI 10.1561/116.20250005

and subjective evaluation experiments with those obtained without disfluency annotation.

Keywords: Text-to-speech synthesis, text style transfer, spontaneous speech, disfluency

1 Introduction

With the rapid improvement in speech synthesis and recognition performance through the development of machine learning and deep learning technologies, we are now entering an era in which spoken communication between "people and computers" and between "people via computers" is a matter of course. In this context, research on text-to-speech synthesis (TTS), a technology for generating appropriate speech using natural language text as input, is developing significantly. In particular, "TTS for reading-style speech" using written text can produce speech as natural as human speech [29]. As the next challenge, there is a growing interest in research on "spontaneous speech synthesis," using spoken text to generate more natural and human-like speech.

The characteristics of spontaneous speech compared with those of reading-style speech include the following: 1) the linguistic form and content of speech are not predetermined, and no practice time is provided during recording; 2) it includes nonverbal phenomena called spontaneous behaviors, e.g., laughter, coughing, interjections, pauses, and disfluencies caused by hesitating, misspeaking or slurring of words; 3) the presence of listeners may affect the speaker in some way [21]. These characteristics make spontaneous speech more challenging to collect data than reading-style speech, and the spontaneous behaviors make modeling spontaneous speech more challenging.

Although spontaneous behaviors are challenging to replicate in speech synthesis, many studies have shown that these behaviors, particularly disfluencies, can affect the speaker's perceived impression and enhance the listener's memory, comprehension, and concentration [23, 1, 36, 8]. These studies indicate that spontaneous behaviors are vital in enhancing the naturalness and effectiveness of spontaneous speech.

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. Yokoyama *et al.* [40] use a conversational dataset [24]

supported by JST CREST under Grant Number JPMJCR22D1, and JSPS KAKENHI under Grant 21H05054.

annotated with six types of paralinguistic information, such as pleasantness and arousal, to enhance neural speech synthesis and control various speaking styles. Guo *et al.* [9] proposed a method for generating more natural prosody in conversations by introducing a context encoder that considers conversational history, along with an auxiliary encoder based on BERT [6] for integrating statistical features from text input. Li *et al.* [19] introduced contextual encoders and a method for predicting spontaneous behaviors from text through the semi-supervised pretraining of a label predictor. The predictor utilizes pseudo-labeled data from a multimodal label detector trained on high-quality spontaneous speech data.

In contrast, research on speech synthesis for monologs is currently centered around synthesizing reading-style speech, and the technology for replicating spontaneous speech production remains in its early stages of development. Moreover, spontaneous behaviors are not only effective in dialog. In monologs, such as a lecture speech, incorporating spontaneous behaviors can effectively retain the listener's memory and create a favorable impression of the speaker, enhancing the audience's attentiveness [8, 28]. Given these considerations, the importance of spontaneous behaviors in monologs deserves greater scholarly attention.

In most research on spontaneous speech synthesis, it is assumed that the text contains parts representing spontaneous behaviors. However, when we want to generate a lecture and explanatory speech that is a monolog, the original written text does not usually include spontaneous elements. Therefore, we focus on spontaneous speech synthesis from written text, aiming to synthesize a more natural and human-like speech [41]. We can accomplish this task by combining text style transfer (TST) [12], which converts only the text style, including sentiment and fluency, to another style while preserving the meaning, with a TTS system that supports spontaneous speech. This combination makes it possible to synthesize lecture and explanatory speech on the basis of existing written materials, thereby reducing the cost and effort of creating new speech scripts.

To achieve this goal, we focus on "disfluency" among spontaneous behaviors and attempt to improve the linguistic and acoustic controllability of disfluency in each model by introducing disfluency annotation to each of the TST and TTS systems. We describe the details of disfluency in Section 2.1. Furthermore, we combine the TST and TTS systems with disfluency annotations to generate speech from fluent text. The contributions of this paper are summarized as follows.

- We show that disfluency annotation improves style controllability in TST systems.
- We investigate TTS systems suitable for spontaneous speech synthesis.

- We show that disfluency annotation improves the reproducibility of disfluency in TTS systems.
- We show that combining the TST and TTS systems with disfluency annotation improves disfluency's linguistic and acoustic controllability in lecture speech generation.

2 Related Works

2.1 Disfluency Analysis

As mentioned in Section 1, nonverbal phenomena, e.g., pauses, disfluency, laughter, and coughing, are some characteristics of spontaneous speech. Among these spontaneous behaviors, we focus on disfluency in this paper. Although numerous studies discuss the definition of disfluency, no standardized definition has yet been established. We use the following definition; "disfluency" can be defined as a phenomenon that interrupts the flow of speech and does not add any propositional content [34]. There are also various ways of categorizing types of disfluency. The main types of disfluency are listed as follows; filled pauses (FPs or fillers) such as "uh" and "like"; repetitions, in which some or all of the same words are repeated; repairs, which occurs when one word is misspoken as another word; and prolongations, which extend the phoneme at the end of a word. There are also cases where silent pauses are included. Repetitions or repairs at the beginning of an utterance are sometimes called false starts. FPs, silent pauses, and repetitions are often collectively described as hesitation. In this paper, we deal with FP, widely studied in previous research, and stutter words, fragments of words that occur due to repetitions and repairs.

In the earliest studies, disfluency was examined mainly from a medical perspective, such as stuttering or aphasia, and in relation to language development in young children. Conversely, disfluency in spontaneous speech in healthy adults was treated as a "redundant and useless element" and was excluded from the scope of linguistic research [5]. Since the mid-1980s to 1990s, in psychology and psycholinguistic studies, analyzing the mechanisms and cognitive processes of human language production has been attempted by capturing spontaneous speech as it is [18, 2, 3]. In addition, with the progress of AI research since the 2010s, the effects of disfluency have been actively studied not only in human–human communication described in Section 2.1.1 but also in human–machine communication described in Section 2.1.2.

2.1.1 In Human–human Communication

Several studies have shown that disfluency in human–human communication positively affects cognition and recall. Arnold *et al.* [1] demonstrated that article fluency (including "thee uh" and "the") affects how a listener interprets the following noun by monitoring the listener's eye movement toward the display object. With fluent articles, listeners were biased toward previously mentioned objects; with disfluent articles, they were biased toward unmentioned objects. These results suggested that listeners use disfluency as a predictive cue about the newness or oldness of information in subsequent elements.

Following this, Watanabe *et al.* [36] investigated whether FPs affect listeners' predictions about the complexity of the following phrase in Japanese. The participant's task was to listen to sentences describing simple and complex shapes on a computer screen and press a button as soon as they identified the corresponding shape. An FP, or a silent pause of equal length or no pause, was placed immediately before the description. Results for native Japanese and non-native Chinese listeners showed that listeners' reaction times to complex shapes were shorter when FPs preceded the phrase describing the shape than when they were absent. This result provided evidence that FPs are a good cue for complex phrases. The response times of non-native listeners with the lowest proficiency level were not affected by the presence or absence of FPs, suggesting that the effect of FPs depends on their language proficiency level.

Fraundorf and Watson [8] showed through an experiment of listening to a recorded story-telling and recalling it orally that FPs facilitate recall at not only the utterance level but also the discourse level. Participants who listened to a pattern containing FPs were compared with those who listened to a pattern containing silence or coughing of the same length, which indicated that the group who listened to the pattern containing FPs showed better recall. On the other hand, a similar experiment conducted on the web, but not in the laboratory, in Germany showed opposite results [25]. This study suggested that the results may be affected by differences in language and experimental design, such as the web-based experiment, where it is more difficult to control for distractors than in the laboratory experiment.

2.1.2 In Human–machine Communication

In response to these results, many researchers have focused on using speech synthesis to reproduce disfluency in human–machine communication and its possible effects. Various studies have been conducted in the context of a dialog, including those described in Section 1. In the context of monolog, Yamashita *et al.* [38] showed that the performance of a deep neural network-

based statistical speech synthesizer could be improved by integrating a rich annotation of contextual linguistic features at the morphological, prosodic and phonetic levels (including disfluent phenomena) in Japanese. Schettino *et al.* [28] conducted a comparative study of Italian tourist guide speech synthesis with and without disfluency (FPs, lengthenings, or silent pauses). In the first experiment, they conducted an AB test, and a significantly large number of participants selected the voice with disfluency as the more natural voice. However, a significantly large number of them also selected the voice without disfluency as the more suitable voice for the virtual avatar. In a second experiment, they conducted a parallel written recall test and an impression test, and the average score of participants who listened to the disfluent voice in the written test was significantly higher. Moreover, the synthesis of the hesitation did not negatively affect the perception of quality or liking.

However, different studies have produced conflicting results regarding the effect of FPs on the perception of personality traits in synthetic speech. Wester *et al.* [37] trained a unit selection system on acting speech containing fillers such as "I mean," "you know," "like," "uh," and "uhm," and found that the presence of these fillers made the synthetic speech sound more nervous, less open, less conscientious, and less extroverted. On the basis of this finding, Gustafson *et al.* [10] tested the effect of filler insertion in synthesized speech on personality evaluation. They also found that in reading-style English speech, filler insertion makes the sound more nervous, less open, less conscientious, and slightly less extroverted. In contrast, they found that Swedish personality ratings had no effect except a perceived increase in spontaneity. Kirkland *et al.* [17] investigated the effects of filled pause placement, speech rate, and F_0 frequency on the speaker's perception of confidence. When a filled pause was inserted, the perceived confidence level decreased, especially when inserted in the middle of the utterance, compared with when inserted at the beginning.

It is important to note that, in common with both human–human and human–machine communication, different results are obtained depending on the language and the conditions assumed, and it is worthwhile to conduct experiments in various languages and conditions. From this perspective, we now examine the effects of disfluency in Japanese lecture speech on human–machine communication.

2.2 Filled Pause Annotation

In a related study, Székely *et al.* [32] performed speech synthesis using Tacotron2 [29] with annotations for two filled pauses (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 text-to-speech (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) [39], which performs effectively even when the alignment between input text and output speech is uncertain or incorrect.

2.3 TST-TTS Integration and Disfluency-aware Dialogue Systems

The study on TST-TTS integration is mentioned by Yoshioka *et al.* [41]. Still, this study focuses almost exclusively on the TST side, and they have not adequately verified the actual combination of TST and TTS. As for the others, there have been studies on using TST as pre-processing (data preparation) [26] and post-processing [31] for other NLP-related tasks, such as chatbot systems.

On the other hand, we focus on the novel framework of performing TST and then TTS to produce the spontaneous speech from written text.

Among the studies on spontaneous speech synthesis for dialogue systems, Cong *et al.* [4] and Li *et al.* [19] focus on disfluencies such as fillers and prolongations in Mandarin conversations. They develop a spontaneous label predictor that enables appropriate spontaneous speech synthesis from disfluent text without manual labeling. Furthermore, Li *et al.* [19] also proposes a multimodal pseudo-label predictor that generates pseudo-labeled data from low-quality corpora, thereby further improving performance through semi-supervised pre-training of the spontaneous label predictor.

3 Proposed Method

3.1 Overview

Using disfluency annotation, we propose a method to generate spontaneous speech with disfluency from text without disfluency. Figure 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 4.2. 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 [32] with minor modifications. Our labeling method uses slightly different annotations between the TST and TTS. We explain our labeling method in Section 3.2. In Section 3.2.1, we describe our labeling method for TST. In Section 3.2.2, we describe our labeling method for TTS. In Section 3.2.3, we explain a combined labeling method for TST and TTS systems. In Section 3.3, we describe our approach's TST method to convert fluent texts into disfluent texts. Section 3.4 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 Sections 3.3 and 3.4 for TST and TTS, the crux of our proposal is the disfluency annotation strategy and the integration of TST and TTS.

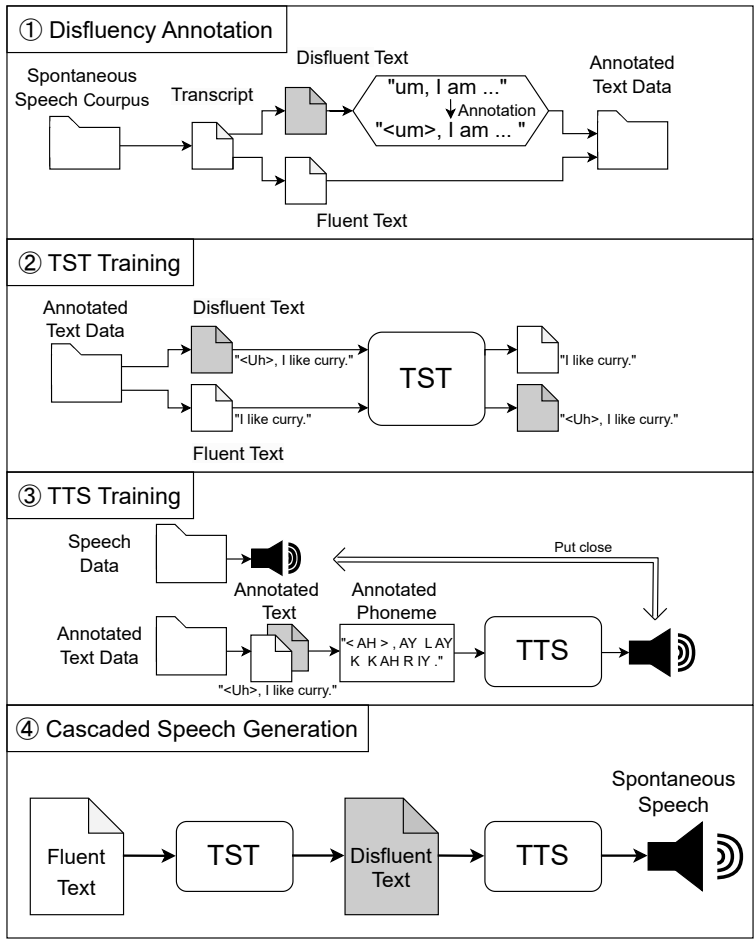

Figure 1: Overview of our proposed method. We first preprocess the transcript data with disfluency annotation and make fluent and disfluent texts. Secondly, we train the bidirectional TST system using disfluency-annotated texts and the TTS system using disfluency-annotated texts and spontaneous speech. Finally, we use the TST system to add disfluency to fluent text and the TTS system to generate spontaneous speech from disfluency-added text.

3.2 Disfluency Annotation

3.2.1 Disfluency Annotation for TST

Table 1 shows disfluency annotation methods for TST. We propose the following three disfluency annotation methods for TST:

Table 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	<> or ()	Specified with registered disfluency words.
S-Tag	or	Specified with arbitrary words.

- 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 fillers, 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 filler 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 filler 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.

3.2.2 Disfluency Annotation for TTS

Table 2 shows disfluency annotation methods for TTS. We propose the following three disfluency annotation methods. As shown later in Section 4.1.3, `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.

Table 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.

- 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, fillers 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, fillers 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.

3.2.3 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.

3.3 Base Model for TST: CycleCVAE+CWS

CVAE [15], a probabilistic model, can be used to implement TST with relative simplicity [12]. In TST, CVAE conditions the decoder by content features obtained from input text to the encoder and style features obtained from class labels. Since only reconstructions are learned during training, CVAE does not require GT-transferred data for the style, whereas the labels must be known. The content word storage (CWS) mechanism explicitly defines the information retained during style conversion as "content words," separating them from the content features and directly forwarding them to the decoder. CWS uses an attention mechanism to calculate the embedded representation of content words in the input sentence and the attention weights for each step of the decoder, and the product of the attention weights and the embedded representation of content words is directly forwarded with the decoder output as a context vector for word prediction. When combined, CVAE+CWS can improve content preservation during generation [41].

To further improve style control performance, CycleCVAE+CWS has been proposed [41]. Figure 2 shows the model architecture of CycleCVAE+CWS. CycleCVAE+CWS uses the style-transferred text synthesized by CVAE+CWS as pseudo-parallel data and simultaneously reconstructs and reconverts the pseudo-parallel data back to the original style. CycleCVAE+CWS takes two types of input, the original text and style-transferred text, and outputs the reconstructed text from the original text and cycle-reconstructed text restored to the original style from the style-transferred text. It also calculates the reconstruction loss between inputs and outputs for each.

CycleCVAE+CWS has improved style control performance and high content preservation while maintaining the condition without parallel data. Note that this differs from CycleGAN [42] and CycleVAE [33], which output the style-transferred text from the original text and further output the text back to the original style to calculate the loss in a single model.

3.4 Base Model for TTS: DVT

DVT [39] is a TTS method using a diffusion probabilistic model. Figure 3 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

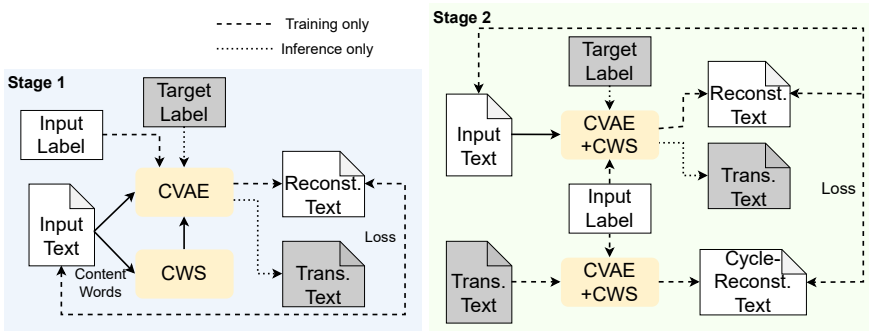

Figure 2: Model architecture of CycleCVAE+CWS.

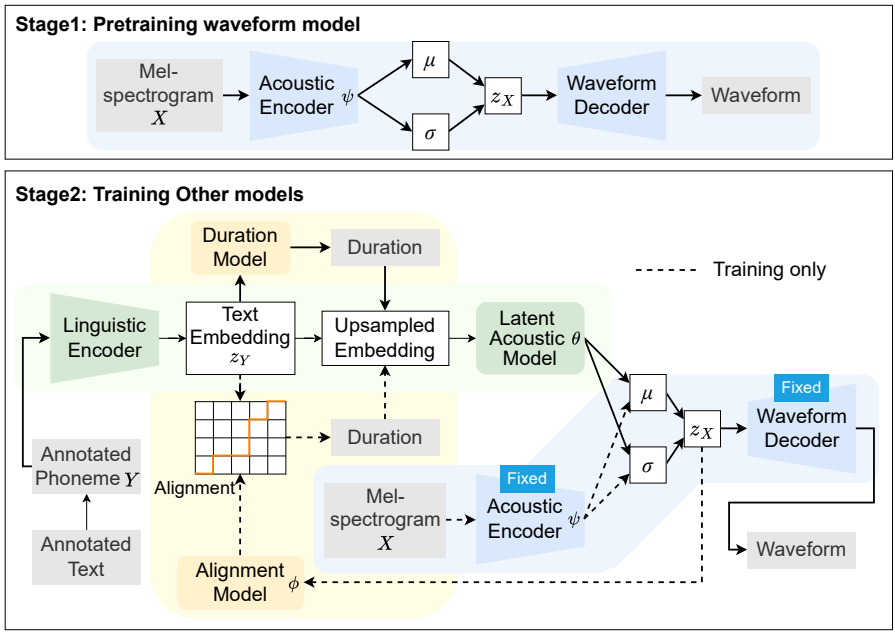

Figure 3: Model architecture of DVT.

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 4.2. 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 $\sigma^2_\psi(X)$ 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 [13]. 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 [39]. 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 [7]. This finding suggests that DVT can effectively generate spontaneous speech that contains many difficult-to-model elements, including disfluency.

4 Experimental Evaluations

4.1 Experiment 1: Disfluency Annotation for TST

4.1.1 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 [41] 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 4.1.2.

For the experimental data, we used the corpus of spontaneous Japanese (CSJ) [22]. CSJ has manually annotated the word, for example, (F) (filler) and (D) (stutter word). We preprocess the transcripts from CSJ in the following steps;

- divided the data by labeling each text as "with" or "without" disfluency using CSJ's manually annotated word labels,
- deleted manually annotated word labels without (F) and (D) labels (CSJ includes some other labels, such as whisper (L) and laughing ()),
- separated transcripts into short units of about 10–20 words,
- labeled each unit according to whether it contained a disfluency,
- 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.

4.1.2 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 [14] 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 [27] 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 filler () and stutter word () in `Symbol`, and brackets were removed in `Tag` and `S-Tag` before calculation.
- Content word error rate (CWER): CWER [41] 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.

4.1.3 Results

Table 3 shows each experiment's objective evaluation results. From the results, the proposed method outperformed the baseline `Plain` method in each index. In 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 4.3.

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

Table 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

Table 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

4.2 Experiment 2: Disfluency Annotation for TTS

4.2.1 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 4.3. 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 [22] 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 4.3, 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 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

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

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 [11], which used a pretrained x-vector [30] 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 [39]. 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]

4.2.2 Results

Table 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 phonological stretches. 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 fillers 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, focusing on disfluency. Therefore, we decided to use DVT in subsequent experiments.

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

Table 5: Evaluation results for MOS in TTS.

Method	MOS
GT	4.69 ± 0.07
Fastspeech2	2.78 ± 0.14
DVT	**3.01 ± 0.12**

Figures 4 and 5 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 [32], 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.

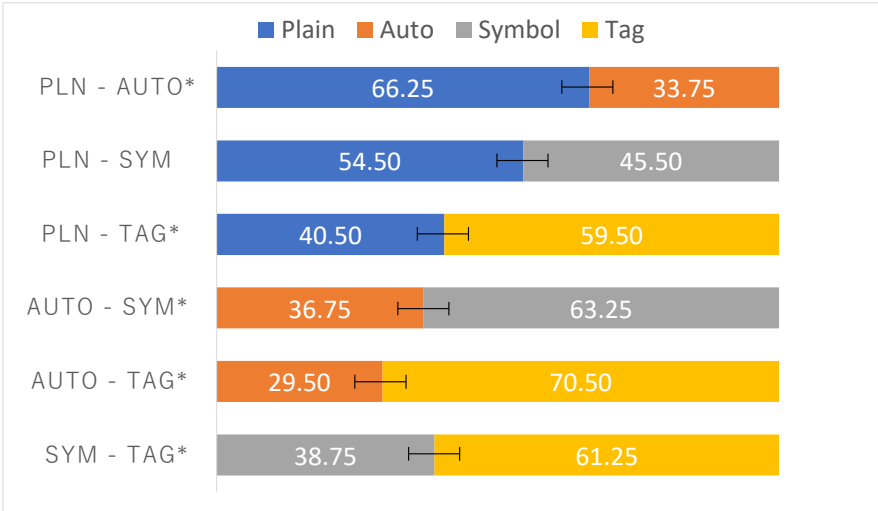

Figure 4: ABX test results on the preference for style reproducibility in TTS. Pairs marked with an asterisk indicate significant differences.

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.

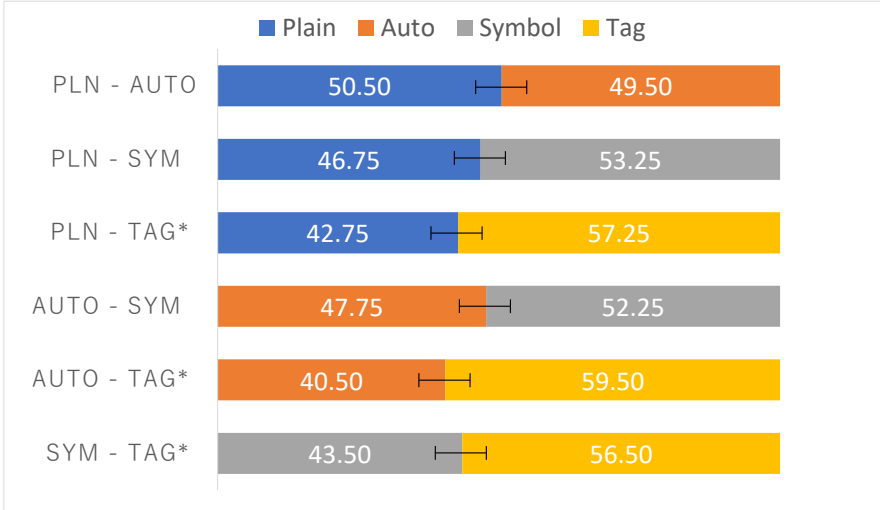

Figure 5: ABX test results on the preference for naturalness in TTS. Pairs marked with an asterisk indicate significant differences.

4.3 Experiment 3: Disfluency Annotation for TST + TTS

4.3.1 Settings

We conducted subjective evaluation experiments, an ABX test as in Section 4.2, 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 4.2.1. 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 [22] as in Sections 4.1 and 4.2. As in Section 4.2, we built the multispeaker TTS model using pseudo-speaker labels.

4.3.2 Results

Figures 6 and 7 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`.

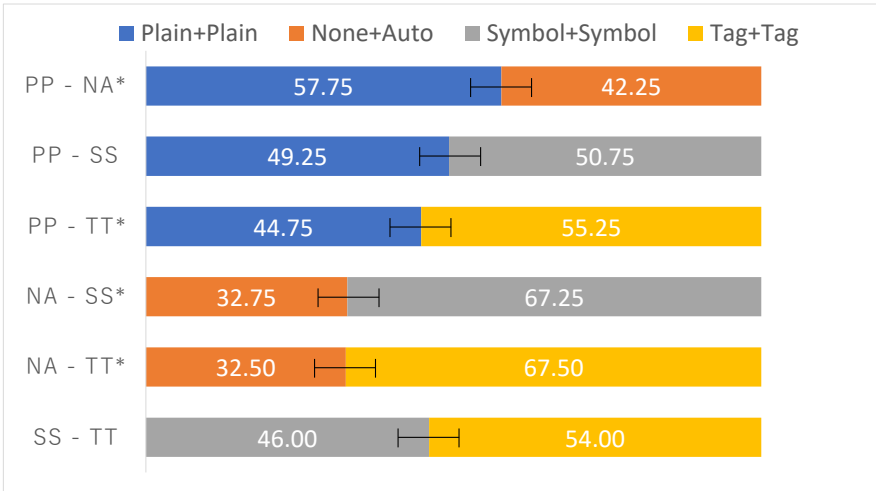

Figure 6: ABX test results on the preference for style reproducibility in TST + TTS. Pairs marked with an asterisk indicate significant differences.

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 4.4.

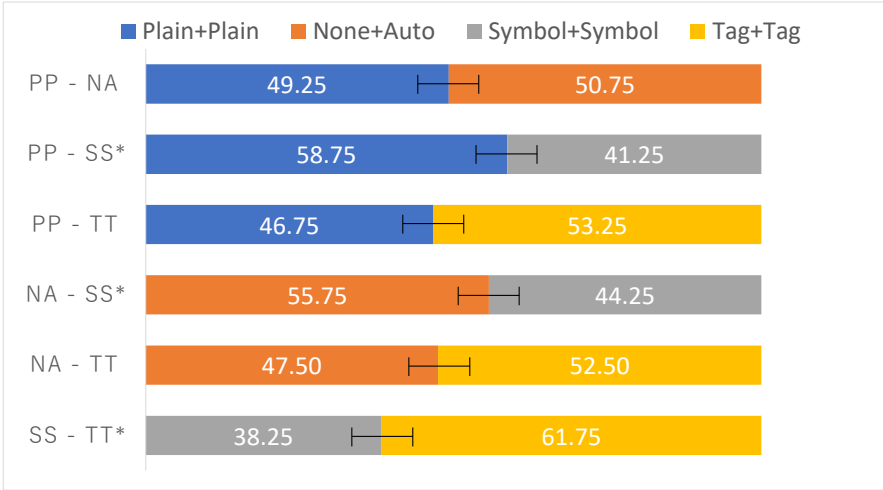

Figure 7: ABX test results on the preference for naturalness in TST + TTS. Pairs marked with an asterisk indicate significant differences.

4.4 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 [4, 19]. In contrast, we directly annotate the input text without implementing new inputs into 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 [32], 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 [20], five types of fillers were identified, but in terms of pronunciation, they can be summarized into two series: nasal (erm/um) and non-nasal (eh/uh) [16]. Székely *et al.* [32] also dealt with only two types of fillers: uh and um. In contrast, Japanese fillers are more varied than English fillers [35]. In addition, since we were also 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 [11] with 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. `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.

Table 6: 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

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 disfluent symbols into speech: Stutter word symbols may be ignored; filler symbols may be output as other disfluencies, such as stutter words, which are generally not considered as fillers; 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 4.2 and the length of the fluent synthesized speech excluding the disfluency, as shown in Table 7. 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.

Table 7: The average length of each synthesis speech (sec). Whole is the length of the synthesized speech used in Section 4.2, 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

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 "(that)" and "(it)" are used as both pronouns and fillers. However, in `Tag`, the accidental transfer of these pronouns to fillers does not markedly affect pronunciation. In contrast, when `Symbol` replaces these words with [FILLER], the TTS system may pronounce them as entirely different fillers (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 fillers 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 fillers, but `Symbol` tends to output a

relatively large number of stutter words. Table 8 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 4.3 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 8: Proportion of disfluency in the text and the proportion of fillers 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.

5 Conclusion

In this paper, we proposed three types of disfluency annotation: `Symbol`, `Tag`, and `S-TAG` for text style transfer (TST) and `Symbol`, `Tag`, and `Auto` for text-to-speech (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.

The results of the first experiment 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. The results of the second experiment 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. The results of the third experiment 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.

This research has applications in the educational field, such as lecture videos, classes with virtual teachers, and a more human-like reading-aloud function for online news articles. The limitation of this paper is that the experiments were only conducted in 'limited language (Japanese)' and 'limited spontaneous behavior (filler and stutter words).' In particular, we need to test our proposed method in other languages to determine whether the reasons for `Symbol`'s failure were derived from the model or language differences.

References

[1] J. E. Arnold, M. K. Tanenhaus, R. J. Altmann, and M. Fagnano, "The old and thee, uh, new: Disfluency and reference resolution", *Psychological science*, 15(9), 2004, 578–82.

[2] E. R. Blacfkmer and J. L. Mitton, "Theories of monitoring and the timing of repairs in spontaneous speech", *Cognition*, 39(3), 1991, 173–94, ISSN: 0010-0277, DOI: https://doi.org/10.1016/0010-0277(91)90052-6.

[3] H. H. Clark and T. Wasow, "Repeating Words in Spontaneous Speech", *Cognitive Psychology*, 37(3), 1998, 201–42, ISSN: 0010-0285, DOI: https://doi.org/10.1006/cogp.1998.0693.

[4] J. Cong, S. Yang, N. Hu, G. Li, L. Xie, and D. Su, "Controllable Context-Aware Conversational Speech Synthesis", in *Proc. Interspeech 2021*, ed. H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, and P. Motlícek, ISCA, 2021, 4658–62.

[5] Y. Den and M. Watanabe, "Some Functions of Disfluency in Speech Communication", *Journal of the Phonetic Society of Japan*, 13(1), 2009, 53–64, DOI: 10.24467/onseikenkyu.13.1_53.

[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", in *Proc. NAACL-HLT Volume 1 (Long and Short Papers)*, ed. J. Burstein, C. Doran, and T. Solorio, 2019, 4171–86.

[7] 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, 4408–12, DOI: 10.21437/Interspeech.2024-2337.

[8] S. H. Fraundorf and D. G. Watson, "The disfluent discourse: Effects of filled pauses on recall", *Journal of Memory and Language*, 65(2), 2011, 161–75, ISSN: 0749-596X.

[9] 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, 403–9.

[10] 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*, ed. G. Németh, ISCA, 2021, 48–53.

[11] 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.

[12] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Toward Controlled Generation of Text", in *Proc. ICML*, ed. D. Precup and Y. W. Teh, Vol. 70, 2017, 1587–96.

[13] 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.

[14] Y. Kim, "Convolutional Neural Networks for Sentence Classification", in *Proc. EMNLP*, 2014, 1746–51.

[15] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised Learning with Deep Generative Models", in *Proc. NeurIPS*, ed. Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, 2014, 3581–9.

[16] 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*, 65(2), 2022, 263–89.

[17] 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*, ed. H. Ko and J. H. L. Hansen, ISCA, 2022, 4990–4.

[18] W. J. Levelt, "Monitoring and self-repair in speech", *Cognition*, 14(1), 1983, 41–104, ISSN: 0010-0277, DOI: https://doi.org/10.1016/0010-0277(83)90026-4.

[19] 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*, ed. N. Harte, J. Carson-Berndsen, and G. Jones, ISCA, 2023, 3377–81.

[20] 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*, 22(3), 2017, 319–44.

[21] K. Maekawa, "Spontaneous speech in the light of linguistics.", *Proc. Spring Meet. Acoust. Soc. Jpn.*, 2001(1), 2001, 19–22.

[22] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous Speech Corpus of Japanese", in *Proc. LREC*, 2000, 947–52.

[23] C. Martin and H. R. J., "Hesitation in speech can um help a listener understand.", *Proc. CogSci*, 25, 2003, 276–81.

[24] H. Mori, T. Satake, M. Nakamura, and H. Kasuya, "Constructing a spoken dialogue corpus for studying paralinguistic information in expressive conversation and analyzing its statistical/acoustic characteristics", *Speech Commun.*, 53(1), 2011, 36–50.

[25] 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*, ed. H. Hermansky, H. Cernocký, L. Burget, L. Lamel, O. Scharenborg, and P. Motlícek, ISCA, 2021, 3979–83.

[26] S. Mukherjee, V. Hudecek, and O. Dusek, "Polite Chatbot: A Text Style Transfer Application", in *Proc. EACL*, ed. E. Bassignana, M. Lindemann, and A. Petit, Association for Computational Linguistics, 2023, 87–93.

[27] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation", in *Proc. ACL*, 2002, 311–8.

[28] 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.*, 85, 2024, 101585, ISSN: 0885-2308.

[29] 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, 4779–83.

[30] 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, 5329–33, DOI: 10.1109/ICASSP.2018.8461375.

[31] Y. Su, Y. Wang, D. Cai, S. Baker, A. Korhonen, and N. Collier, "PROTOTYPE-TO-STYLE: Dialogue Generation With Style-Aware Editing on Retrieval Memory", *IEEE ACM Trans. Audio Speech Lang. Process.*, 29, 2021, 2152–61.

[32] 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*, ed. M. Pucher, ISCA, 2019, 245–50.

[33] P. L. Tobing, Y. Wu, T. Hayashi, K. Kobayashi, and T. Toda, "Non-Parallel Voice Conversion with Cyclic Variational Autoencoder", in *Proc. Interspeech 2019*, ed. G. Kubin and Z. Kacic, 2019, 674–8.

[34] J. E. Tree, "The Effects of False Starts and Repetitions on the Processing of Subsequent Words in Spontaneous Speech", *Journal of Memory and Language*, 34(6), 1995, 709–38, ISSN: 0749-596X, DOI: https://doi.org/10.1006/jmla.1995.1032, https://www.sciencedirect.com/science/article/pii/S0749596X85710327.

[35] 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*, ed. J. Véronis and E. Campione, ISCA, 2005, 169–72, https://www.isca-archive.org/diss%5C_2005/watanabe05%5C_diss.html.

[36] 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*, 50(2), 2008, 81–94, ISSN: 0167-6393.

[37] M. Wester, M. P. Aylett, M. Tomalin, and R. Dall, "Artificial personality and disfluency", in *Proc. INTERSPEECH*, ISCA, 2015, 3365–9.

[38] Y. Yamashita, T. Koriyama, Y. Saito, S. Takamichi, Y. Ijima, R. Masumura, and H. Saruwatari, "Investigating Effective Additional Contextual Factors in DNN-Based Spontaneous Speech Synthesis", in *Proc. Interspeech*, ed. H. Meng, B. Xu, and T. F. Zheng, ISCA, 2020, 3201–5.

[39] 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, 1–5, DOI: 10.1109/ICASSP49357.2023.10094298, https://doi.org/10.1109/ICASSP49357.2023.10094298.

[40] M. Yokoyama, T. Nagata, and H. Mori, "Effects of Dimensional Input on Paralinguistic Information Perceived from Synthesized Dialogue Speech with Neural Network", in *Proc. Interspeech*, ed. B. Yegnanarayana, ISCA, 2018, 3053–6.

[41] 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*, 33, 2025, 333–46, DOI: 10.1109/TASLPRO.2024.3522757.

[42] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks", in *Proc. ICCV*, 2017, 2242–51.