Phonikud: Hebrew Grapheme-to-Phoneme Conversion for Real-Time Text-to-Speech

Yakov Kolani[1] **Maxim Melichov**[2] **Cobi Calev**[1] **Morris Alper**[3]

[1]Independent Researcher [2]Reichman University [3]Tel Aviv University

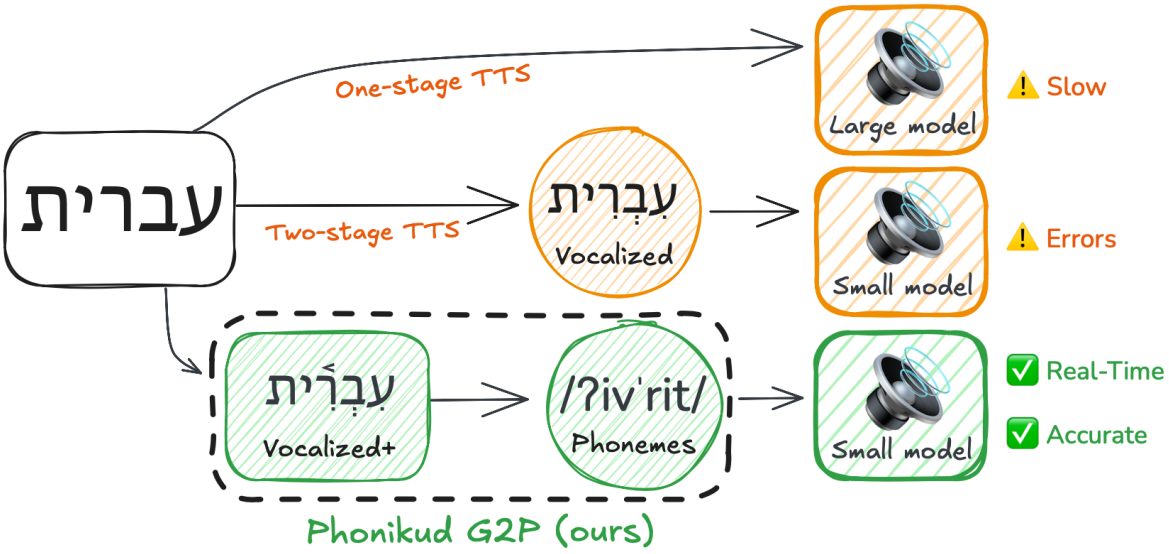

Figure 1: **Phonikud: Hebrew G2P conversion for fast, phonetically accurate Hebrew TTS.** Hebrew writing normally omits vowels, creating a speed-accuracy tradeoff for TTS: large models (orange, top) trained on raw Hebrew text are slow, while small models trained on vocalized text (orange, middle) produce pronunciation errors since added vowel marks still omit critical features like stress. Phonikud (green) achieves fast and phonetically accurate TTS via enhanced vocalization (*Vocalized+* above), augmenting standard vowel marks with additional symbols indicating phonetic features such as stress, followed by conversion to standard IPA phonemes. This enables training small TTS models with phonetically accurate outputs suitable for real-time applications.

Abstract

Real-time text-to-speech (TTS) for Modern Hebrew is challenging due to the language's orthographic complexity. Existing solutions ignore crucial phonetic features such as stress that remain underspecified even when vowel marks are added. To address these limitations, we introduce *Phonikud*, a lightweight, open-source Hebrew grapheme-to-phoneme (G2P) system that outputs fully-specified IPA transcriptions. Our approach adapts an existing diacritization model with lightweight adaptors, incurring negligible additional latency. We also contribute the *ILSpeech* dataset of transcribed Hebrew speech with IPA annotations, serving as a benchmark for Hebrew G2P and as training data for TTS systems. Our results demonstrate that Phonikud G2P conversion more accurately predicts phonemes from Hebrew text compared to prior methods, and that this enables training of effective real-time Hebrew TTS models with superior speed-accuracy trade-offs. We release our code, data, and models at `https://phonikud.github.io`.

1 Introduction

Despite the Modern Hebrew language being spoken by approximately nine million people (Lewis, 2009), it currently lacks an open-source real-time text-to-speech (TTS) system with adequate performance. TTS systems for important applications such as screen readers for visually impaired users and for smart home technology must run locally in real-time on resource-constrained devices. However, applying standard techniques to Hebrew is challenging due to the language's opaque orthography, which is difficult to parse directly for the small TTS models needed to achieve low latency.

The Hebrew script omits phonetic features such as vowel sounds, leaving them to be inferred from

context. For instance, in Hebrew the word סְפַר may be read as /ˈsefer/ (“book”), /saˈpar/ (“barber”), /saˈfar/ (“he counted”), or /sfar/ (“suburb”). A system of optional diacritics (*nikud*) may be used to indicate these features, but they are mostly confined to pedagogical texts such as dictionaries. Moreover, the pronunciation of a Hebrew word cannot be unambiguously determined even when vowel diacritics are provided. For example, בִּירָה may be read as either /ˈbira/ (“beer”) or /biˈra/ (“capital city”). This *phonetic underspecification* challenges TTS systems, which must receive normal (unvocalized) Hebrew text as input and output correctly-pronounced Hebrew audio.

One approach maps unvocalized Hebrew text directly to audio (Roth et al., 2024; Zeldes et al., 2025). However, the large models needed to capture the complexities of Hebrew orthography incur high latency, making them unsuitable for real-time applications. Conversely, small TTS models struggle to predict accurate pronunciation from unvocalized Hebrew. Existing approaches predict vowel diacritics directly (Sharoni et al., 2023; Pratap et al., 2024), but this does not fully resolve ambiguity (as in the example above), leading to inaccurate pronunciations in TTS outputs.

To bridge this gap, we propose a lightweight grapheme-to-phoneme (G2P) pipeline, *Phonikud*, to resolve the phonetic ambiguities in written Hebrew. Specifically, we adapt an existing state-of-the-art (SOTA) model for predicting Hebrew vowel diacritics (Shmidman et al., 2023), adding lightweight adaptors to efficiently predict additional phonetic features such as stress and *shva* realization (see Section 2) needed for disambiguation. A rule-based module converts these outputs into the International Phonetic Alphabet (IPA). We show that this allows effectively training small, real-time-capable TTS models. During inference, these models accept IPA input directly for precise phonetic control; additionally, they may be applied to unvocalized Hebrew text by applying our G2P conversion with Phonikud.

As an additional step to this goal, we contribute the novel *ILSpeech* dataset and benchmark, consisting of high-quality Hebrew speech recordings along with Hebrew text and expert-annotated IPA transcriptions. This serves both as an additional training resource for Hebrew TTS, which currently has a dire lack of available open data, as well as a benchmark for evaluating the novel task of G2P for Hebrew text.

In summary, our key contributions are:

- A lightweight, open-source G2P model augmenting an existing Hebrew diacritizer to accurately transcribe Hebrew text in IPA.
- Results demonstrating that G2P is beneficial for training real-time TTS systems for Hebrew, along with comparisons to existing systems.
- *ILSpeech*, a novel dataset and benchmark of Hebrew speech recordings, Hebrew and IPA transcriptions, enabling TTS training and benchmarking Hebrew G2P.

We release[1] our data, code, and trained models to spur development of open-source real-time Hebrew TTS systems.

2 Phonetic Underspecification in Hebrew

Hebrew is normally written without vowel marks (*unvocalized text*), but even when these are added (*vocalized text*) it is still underspecified for various phonetic features that are needed for accurate TTS. These may be split into three primary issues:

Stress. Lexical stress is only partially predictable from word shape and part of speech in Hebrew (Graf and Ussishkin, 2003). As illustrated by the minimal pair /ˈtxina/ (“tahini”) vs. /txiˈna/ (“grinding”), both spelled טְחִינָה, stress is not indicated in the orthography even when vowel marks are provided.

Shva. The vowel mark known as *shva* is polyvalent, being either silent or pronounced as /e/. Its pronunciation depends on complex morpho-phonological rules with many irregularities (Weinberg, 1966). For example, in בְּלוֹנְדוֹן /beˈlondon/ (“in London”) the shva vowel between the first two consonants is pronounced, while in בְּלוֹנְדִינִי /blonˈdini/ (“blonde”) it is silent.

Irregular words. Infrequently, words may deviate from regular pronunciation rules. A notable case is loanwords containing the phoneme /w/, written identically to /v/. For example, פִּינְגְּוִין /ˈpingwin/ (“penguin”) is indistinguishable from the hypothetical form */ˈpingvin/. Other examples of irregular spellings include יַאלָלָה /ˈjala/ (“come on”) and יִשָּׂשכָר /jisaˈxar/ (“Issachar”).

These ambiguities motivate our approach of augmenting existing diacritization with additional phonetic disambiguation before converting to IPA.

[1] https://phonikud.github.io

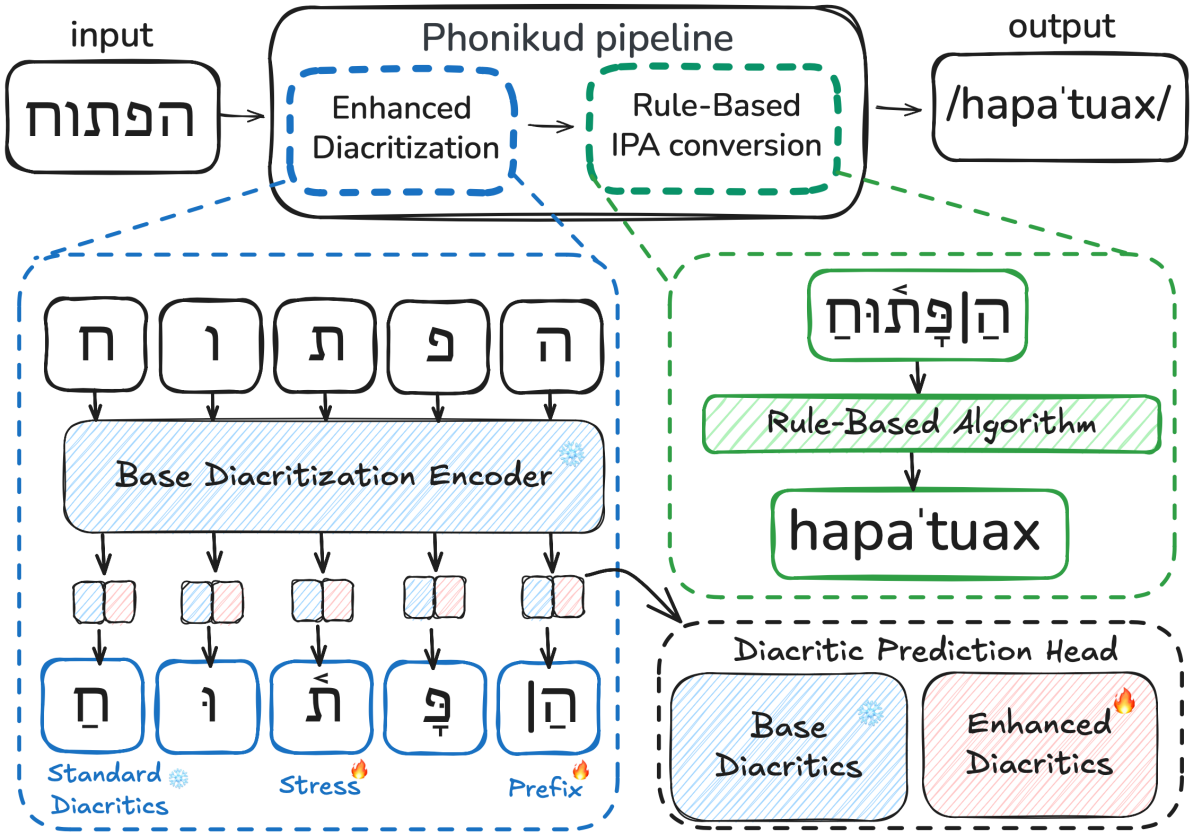

Figure 2: **The Phonikud grapheme-to-phoneme pipeline.** Phonikud converts unvocalized Hebrew text into fully-specified IPA in two steps: First, an enhanced diacritization module adds standard vowel marks and enhanced phonetic symbols to each letter of the text. This is done with a frozen (ice symbol above) base diacritization model, which is a character-level encoder model, and its per-character prediction head for standard vowel diacritics. This is augmented with an additional trainable (fire symbol above) linear adaptor serving as a head for predicting enhanced diacritics, which disambiguate the text's phonetic content. Second, a rule-based transformation module converts this text into IPA, which may be used to train small, real-time TTS models.

3 Method

Our Phonikud system is illustrated in Figure 2. As a G2P pipeline, Phonikud takes unvocalized Hebrew text as input and outputs fully-specified IPA transcriptions, which can then be used to train efficient Hebrew TTS systems. We proceed to describe the two key components of this system – an enhanced diacritization module (Section 3.1) and rule-based IPA conversion module (Section 3.2) – followed by the novel procedure used for training the system's learnable components (Section 3.3).

3.1 Enhanced Diacritization

Strong, efficient models have already been developed to add vowel diacritics to Hebrew text, achieving high accuracy on this standard vocalization task (Shmidman et al., 2023). Rather than learning to transcribe from scratch, we leverage this existing capability while extending it to predict additional phonetic features needed for disambiguation. Our key insight is to augment an existing character-level encoder-based diacritization model with lightweight prediction heads to extend the set of symbols which it may predict.

We add three additional symbols that can be predicted for each character position, with logits output by new, trainable MLP prediction heads. We freeze the base encoder model and all existing prediction heads for standard vowel diacritics, while adding and training only the added weights for each new enhanced diacritic. This approach offers several advantages: training is extremely lightweight, inference predicts both standard and enhanced diacritics in parallel with minimal runtime overhead compared to the base diacritizer, and performance on standard diacritization remains constant since the base model is frozen.

We introduce three enhanced diacritics: (1) a

superscript angle indicating a stressed syllable (e.g. לֶ֫חֶם[2]), (2) a subscript line indicating a shva vowel pronounced as /e/ (e.g. מְתִיחָה[3]), and (3) a vertical bar indicating the end of a cliticized prefix (e.g. הַ|קּוֹד[4]). The first two directly indicate missing phonetic features, while the last aids dictionary matching of irregular words, resolving the issues from Section 2. These graphemes, which include traditional Biblical cantillation marks, are chosen because they are not used in ordinary writing.

3.2 Rule-Based IPA Conversion

After generating enhanced vocalized forms (e.g. לֶ֫חֶם), the phonemic representation can be unambiguously determined. We apply a deterministic, rule-based algorithm to convert this to standard IPA (e.g. /ˈlexem/). This is primarily implemented with a finite-state transducer, with states corresponding to character n-grams and output symbols corresponding to IPA phonemes. This addresses several orthographic complexities of Hebrew:

Many-to-one mappings. Multiple Hebrew graphemes frequently map to a single phoneme. For example, ט and ת both represent /t/, and three distinct vowel symbols may map to /e/.

Non-monotonic sequences. Some Hebrew words are parsed non-monotonically (not in a linear order), such as רֵיחַ ("smell") representing /ˈreax/ (not */ˈrexa/, which would be the reading in linear order).

Dual-function letters. The letters ו and י may function as vowels or consonants depending on orthographic context. Words such as סִוּוּג /siˈvug/ ("classification") require complex logic to determine that the first ו represents /v/ while the second coalesces to the vowel /u/.

Irregular words. As discussed in Section 2, irregular words may require dictionary lookup to determine the correct pronunciation.

By addressing these complexities in the IPA conversion stage, Phonikud efficiently simplifies the representation used for TTS training. This offers several benefits: IPA provides a standardized linguistic representation that is easy to interpret and edit, and it maintains compatibility with multilingual training scenarios, while still preserving the essential phonetic information needed for TTS (shown in Section 5.3). Moreover, it allows for user choice at inference time, as users may either input IPA directly or generate speech from Hebrew text by using the Phonikud enhanced diacritization model and/or IPA conversion.

3.3 Training Procedure

A fundamental challenge in our approach is the lack of existing ground-truth (GT) annotations for Hebrew phonetic features like lexical stress. To address this limitation, we employ a human-in-the-loop procedure to distill knowledge from existing resources along with manual refinement. In particular, we semi-automatically annotate a large-scale Hebrew corpus to indicate stress placement, prefix boundaries, and shva realization. We then distill this knowledge into our model by fine-tuning it on this pseudo-GT, which we find enables Phonikud to outperform existing baselines.

To produce large-scale data with pseudo-GT annotations, we adopt the IsraParlTweet corpus consisting of 5M lines of Hebrew text (Mor-Lan et al., 2024). We leverage Dicta's[5] morpho-phonological analysis API along with a set of known linguistic rules to automatically predict stress placement, prefix boundaries, and shva realization. As this procedure is frequently inaccurate, we correct many cases of errors via manual annotation, by sorting words types by frequency and correcting the most common items.

4 ILSpeech

We introduce *ILSpeech*, a high-quality Hebrew speech dataset with expert-annotated phonetic transcriptions. This dataset serves two primary purposes: (1) establishing a benchmark for Hebrew G2P systems by providing ground-truth phonetic transcriptions for systematic evaluation, and (2) supplying high-quality training data for Hebrew TTS development. Our dataset contains approximately two hours of studio-quality speech from two native Hebrew speakers (~1.5K sentences), representing a proof-of-concept that may be extended with additional speakers and content as needed.

ILSpeech provides time-aligned transcriptions with two parallel tiers: (1) unvocalized Hebrew text, and (2) expert-annotated IPA transcriptions. The latter fully specifies phonetic features such as stress that are ambiguous in vocalized Hebrew text. To the best of our knowledge, this is the first open Hebrew audio corpus containing full IPA transcrip-

[2]Pronounced /ˈlexem/ ("bread").
[3]Pronounced /metiˈxa/ ("stretch").
[4]Pronounced /haˈkod/ ("the code"), with prefixed /ha-/
[5]https://dicta.org.il

Model	WER↓	WER$^{\sigma}$↓	CER↓	בוקר טוב
Phonikud (Ours)	0.19	0.15	0.04	ˈboker ˈtov
Diacritizers*				
DictaBERT	0.38	0.24	0.08	boˈker ˈtov
Nakdimon	0.40	0.27	0.09	boˈker ˈtov
Multilingual G2P				
eSpeak NG	1.00	0.96	0.47	vvkr tov
Goruut	1.00	0.95	0.48	boʁɐʁ tˢoːβ
CharsiuG2P	1.00	0.99	0.71	boːʔab têːb

Table 1: **G2P evaluation and example.** We test on ILSpeech, using unvocalized Hebrew as input and comparing to ground-truth IPA annotations. WER$^{\sigma}$ indicates word error rate while disregarding mismatched stress. We illustrate performance on a Hebrew phrase with GT /ˈboker ˈtov/. *Diacritizers use our IPA conversion with defaults for ambiguous features like stress.

Model	WER↓	CER↓	RTF↓	# Params
Phonikud (Ours)				
Piper	0.08	0.02	0.09	20M
StyleTTS2	0.07	0.02	0.50	90M
Open Models				
MMS	0.20	0.06	0.21	36M
SASPEECH	0.11	0.04	0.16	28M
Robo-Shaul	0.08	0.04	1.58	23M
Proprietary Models				
Google	0.04	0.02	4.08	—
OpenAI	0.05	0.02	1.60	—

Table 2: **TTS Comparison.** We compare TTS models trained using our Phonikud G2P conversion (top), existing open models for Hebrew (middle), and proprietary models (bottom). Accuracy metrics (WER, CER) and latency (RTF) are calculated are calculated as described in Section 5.2. Our method yields a superior trade-off between these two dimensions.

tions, newly enabling evaluation of Hebrew G2P systems on previously unmeasurable features such as stress placement and shva realization.

The dataset addresses a critical gap in Hebrew speech resources. While several Hebrew audio corpora exist, they lack the phonetic detail necessary for G2P evaluation. In addition, existing corpora are mostly small-scale (Izre'el et al., 2001; Azogui et al., 2016; Marmorstein and Matalon, 2022; Conneau et al., 2023; Sharoni et al., 2023), while the only open large-scale corpora contain low-quality recordings without proper segmentation (Marmor et al., 2023; Turetzky et al., 2024), unsuitable for high-quality TTS training. As such, ILSpeech provides an important contribution with high-quality recording data along with expert phonetic annotations necessary for rigorous G2P evaluation.

We release ILSpeech to support open research in Hebrew speech technology, under a non-commercial license with ethical use requirements.

5 Results

Below, we provide results for G2P conversion with Phonikud and existing baselines (Section 5.1), compare downstream TTS using Phonikud to existing Hebrew TTS systems (Section 5.2), and ablate key components of our system (Section 5.3).

5.1 G2P Evaluation

Results of our Phonikud G2P system are shown in Table 1, evaluated relative to the ground-truth IPA annotations in ILSpeech. We calculate word- and character error rates (WER, CER) and WER when disregarding stress (WER$^{\sigma}$). We compare to two baselines: Firstly, we apply the existing SOTA Hebrew diacritizers DictaBERT (Shmidman et al., 2023) and Nakdimon (Gershuni and Pinter, 2022) with our IPA conversion, using reasonable defaults for ambiguous features (e.g. final stress, common in Hebrew). Secondly, we compare to existing open-source multilingual G2P libraries which ostensibly support Hebrew: eSpeak NG[6], Goruut[7], and CharsiuG2P[8] (Zhu et al., 2022).

Our system outperforms all of these baselines: Our prediction of features such as stress improves performance significantly relative to existing diacritizers, while existing multilingual G2P systems are nearly unusable for Hebrew due to lack of dictionary support for common words (eSpeak NG) and extensive hallucinations in neural models (Goruut, CharsiuG2P). Errors in our method's outputs stem from both occasional mistakes in predicting features such as stress, as well from limitations of the base diacritization model on which our model is built (see Section 7.1). In the appendix, we provide results of all of these G2P methods applied to a standard Hebrew text for visual comparison.

5.2 Downstream TTS Evaluation

We evaluate our method's utility for downstream TTS by comparing models trained with Phonikud G2P on ILSpeech audio to baseline approaches for fast Hebrew TTS. In Table 2, we report performance of our method applied to train multiple open-source TTS architectures: a light-weight Piper[9]

[6]https://github.com/espeak-ng/espeak-ng
[7]https://github.com/neurlang/goruut
[8]https://github.com/lingjzhu/CharsiuG2P
[9]https://github.com/rhasspy/piper

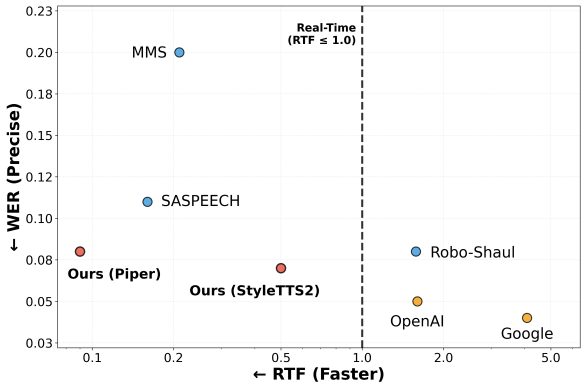

Figure 3: **Speed-accuracy trade-off.** Runtime (x-axis, log-scaled) vs. error rate (y-axis) comparison of our method (red) against open-source (blue) and proprietary (orange) TTS models. Models with RTF ≤ 1.0 (dotted line) are real-time capable, and the lower-right direction reflects better overall performance. Our method achieves a superior speed-accuracy trade-off, with additional advantages on ambiguous features such as stress that are not reflected in automatic metrics.

Model	WER↓	CER↓
Ours	0.11	0.03
-IPA conversion	0.11	0.03
-vowel diacritics	0.24	0.09

Table 3: **Ablation study.** For ablations, we use fixed, light-weight training settings described in Section 5.3. Training on vocalized Hebrew text without IPA conversion (second row) provides comparable results, evidencing that our G2P conversion does not lose important phonetic information. However, when training on unvocalized text (last row), our small model struggles to infer correct vowels and cannot reliably produce intelligible text.

(VITS) (Kim et al., 2021) model, as well as a larger StyleTTS2 (Li et al., 2023) model. We fine-tune existing pretrained English TTS checkpoints on audio from ILSpeech along with IPA transcriptions calculated with Phonikud[10]. During inference, we use Phonikud to convert input Hebrew text to IPA used for synthesis.

We compare to the open-source models MMS (Pratap et al., 2024), the SASPEECH baseline (Sharoni et al., 2023), and Robo-Shaul[11], all light-weight models using a two-stage approach (diacritizing text followed by speech synthesis). We also compare to the proprietary TTS models offered by Google[12] and OpenAI [13], which support Hebrew. In the appendix, we also analyze large models which directly process unvocalized Hebrew text; these are far from real-time performance.

Following standard practice (Roth et al., 2024), we calculate error rates (WER, CER) by applying automatic speech recognition (ASR) to generations and comparing to the original input text. We also report real-time factor (RTF) values to measure latency of each system (including diacritization time, when relevant); for all open systems this is calculated on a consistent, CPU-only hardware setup to match edge computing use cases, while for proprietary models this uses their cloud inference APIs. Following prior work (Roth et al., 2024; Zeldes et al., 2025), we evaluate on a random subset of the SASPEECH (Sharoni et al., 2023) dataset, which is out-of-distribution for our model.

Our system achieves a superior trade-off between speed and accuracy than prior methods, seen visually in Figure 3. This holds both when comparing to open models run on local hardware as well as when comparing to proprietary systems run via external API, further supporting the value of our method for real-time use cases. Importantly, these automatic metrics do not capture the effect of phonetic inaccuracies such as stress placement (as ASR may still return the original, unvocalized text). Qualitative audio comparisons at our project page demonstrate our method's improved handling of these phonetic features relative to prior methods. Our demo also illustrates that our system enables optional user control over features like stress placement.

5.3 Ablations

We ablate key parts of our system in Table 3, fixing the base TTS model (Piper) and training settings (using less training time than our main results for a light-weight comparison; see appendix for details). Removing IPA conversion (i.e. training directly on vocalized Hebrew text) yields similar objective performance to our full system. This illustrates that our IPA conversion, which has various practical advantages (Section 3.2), preserves the essential phonetic content needed to synthesize speech, while also enabling user control at inference time via either IPA or Hebrew text (with IPA conversion applied). However, training directly on undiactized Hebrew text leads to severely degraded performance. Qualitatively, the model trained without vowels struggles

[10]We use IPA output from Phonikud rather than the existing manual IPA annotations to fairly evaluate our G2P pipeline and simulate scalable training on other Hebrew datasets.

[11]https://github.com/maxmelichov/Text-To-speech

[12]Gemini 2.5 Flash TTS

[13]GPT-4o mini TTS

to infer the correct vowel sounds for uncommon words and often produces unintelligible output.

6 Related Work

Hebrew TTS. Early Hebrew TTS systems relied on rule-based formant synthesis (Laufer, 1975), while modern approaches are mainly learning-based. Some use two-step pipelines that add vowel diacritics before speech synthesis (Sharoni et al., 2023; Pratap et al., 2024), but this fails to resolve key phonetic ambiguities such as lexical stress placement. Others adopt an end-to-end approach, predicting speech directly from raw, undiacritized Hebrew text (Roth et al., 2024; Zeldes et al., 2025), but these require large models with high computational overhead, unsuitable for real-time use. We strike a middle ground by using a two-stage approach for computational efficiency while predicting IPA directly to ensure phonetic accuracy.

G2P conversion. Many languages have opaque orthographies, requiring TTS systems to resolve pronunciation ambiguities. Grapheme-to-phoneme conversion simplifies the learning process for TTS systems by offloading this disambiguation from the text synthesis model (Fong et al., 2019; Hexgrad, 2025). This may handle a variety of language-dependent issues, such as homograph disambiguation in English (e.g. *lead* as a verb vs. noun) (Ploujnikov and Ravanelli, 2022), predicting underspecified tone in Thai (Rugchatjaroen et al., 2019), and inferring vowels in Arabic (Elmallah et al., 2024; Kharsa et al., 2024) and Hebrew (Gershuni and Pinter, 2022; Shmidman et al., 2023). However, existing Hebrew vocalization systems and open-source G2P tools do not specify crucial phonetic features such as stress, while our method generates fully-specified phonetic transcriptions.

7 Conclusion

We have presented a new open-source Hebrew G2P system, Phonikud, and have shown that it newly enables the training of small, real-time Hebrew TTS models, which are needed for edge computing applications. Our experiments show our system compare favorably to existing solutions in both quality and runtime performance. We have also introduced the novel ILSpeech dataset and benchmark for Hebrew G2P evaluation and TTS training. We release our data, code, and trained models to enable applications and research. We envision future work building upon our contributions to further improve Hebrew TTS performance while retaining low latency. Additional promising directions include fine-grained prosody control, support for code-switching, extensive logic for expanding symbols such as dates and addresses, semi-automated IPA annotation to increase the scale of ILSpeech, and extensions to other languages facing related phonological and orthographic challenges.

7.1 Limitations

As our method builds on existing models, we inherit a number of their limitations. The Hebrew diacritization model may output inaccurate productions, leading to incorrect IPA transcriptions. It does not support user selection among alternative vocalizations, which may be desirable in ambiguous cases. The diacritization model adheres to the conventions of formal written Hebrew, which may diverge from spoken norms (e.g. formal /sigˈri/ vs. informal /sgeˈri/ for סגרי “close! (f.)”) Finally, when using our full pipeline, the prosodic quality of synthesized voice is constrained by the inherent trade-offs of real-time TTS models due to their limited capacity.

Ethics Statement

TTS is a dual-use technology: it enables valuable applications such as assistive tools for visually impaired users, but can also misused to generate disinformation or low-quality synthetic content. As with other generative models, responsible use is essential. We believe our work represents an important step towards making language technologies more accessible for lower-resourced languages such as Hebrew, while also acknowledging current limitations in representation. Our proposed dataset, like prior resources for Hebrew, cover a narrow set of speakers and styles, lacking adequate coverage of sociolinguistic variation such as the Mizrahi Hebrew accent. We anticipate that future work will increase this coverage to support more equitable and inclusive voice technologies.

Acknowledgements

We thank Dicta for encouraging our work and for approving our release of our model and data which incorporate their results. We also thank the speakers in ILSpeech for providing an essential resource for the development of Hebrew language technologies. Finally, we acknowledge Kush Jain, Shlomo Tannor, and Mark Kahn for their helpful feedback and suggestions.

References

International Phonetic Association. 1999. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

Jacob Azogui, Anat Lerner, and Vered Silber-Varod. 2016. The open university of israel map task corpus (matacop).

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805. IEEE.

Muhammad Morsy Elmallah, Mahmoud Reda, Kareem Darwish, Abdelrahman El-Sheikh, Ashraf Hatim Elneima, Murtadha Aljubran, Nouf Alsaeed, Reem Mohammed, and Mohamed Al-Badrashiny. 2024. Arabic diacritization using morphologically informed character-level model. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1446–1454.

Jason Fong, Jason Taylor, Korin Richmond, and Simon King. 2019. A comparison between letters and phones as input to sequence-to-sequence models for speech synthesis. In *The 10th ISCA Speech Synthesis Workshop*, pages 223–227. International Speech Communication Association.

Elazar Gershuni and Yuval Pinter. 2022. Restoring hebrew diacritics without a dictionary. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1010–1018.

Dafna Graf and Adam Ussishkin. 2003. Emergent iambs: stress in modern hebrew. *Lingua*, 113(3):239–270.

Hexgrad. 2025. G2p shrinks speech models.

Shlomo Izre'el, Benjamin Hary, and Giora Rahav. 2001. Designing cosih: the corpus of spoken israeli hebrew. *International Journal of Corpus Linguistics*, 6(2):171–197.

Ruba Kharsa, Ashraf Elnagar, and Sane Yagi. 2024. Bert-based arabic diacritization: A state-of-the-art approach for improving text accuracy and pronunciation. *Expert Systems with Applications*, 248:123416.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pages 5530–5540. PMLR.

Asher Laufer. 1975. A programme for synthesizing hebrew speech. *Phonetica*, 32(4):292–299.

M. Paul Lewis, editor. 2009. *Ethnologue: Languages of the World*, sixteenth edition. SIL International, Dallas, TX, USA.

Yinghao Aaron Li, Cong Han, Vinay Raghavan, Gavin Mischler, and Nima Mesgarani. 2023. Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models. *Advances in Neural Information Processing Systems*, 36:19594–19621.

Yanir Marmor, Kinneret Misgav, and Yair Lifshitz. 2023. ivrit. ai: A comprehensive dataset of hebrew speech for ai research and development. *arXiv preprint arXiv:2307.08720*.

Michal Marmorstein and Nadav Matalon. 2022. The huji corpus of spoken hebrew: An interaction-oriented design of a corpus.

Guy Mor-Lan, Effi Levi, Tamir Sheafer, and Shaul R Shenhav. 2024. Israparltweet: The israeli parliamentary and twitter resource. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9372–9381.

Artem Ploujnikov and Mirco Ravanelli. 2022. Soundchoice: Grapheme-to-phoneme models with semantic disambiguation.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, and 1 others. 2024. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52.

Amit Roth, Arnon Turetzky, and Yossi Adi. 2024. A language modeling approach to diacritic-free hebrew tts. In *Proc. Interspeech 2024*, pages 2775–2779.

Anocha Rugchatjaroen, Sittipong Saychum, Sarawoot Kongyoung, Patcharika Chootrakool, Sawit Kasuriya, and Chai Wutiwiwatchai. 2019. Efficient two-stage processing for joint sequence model-based thai grapheme-to-phoneme conversion. *Speech Communication*, 106:105–111.

Orian Sharoni, Roee Shenberg, and Erica Cooper. 2023. Saspeech: A hebrew single speaker dataset for text to speech and voice conversion. In *Proc. Interspeech*.

Shaltiel Shmidman, Avi Shmidman, and Moshe Koppel. 2023. Dictabert: A state-of-the-art bert suite for modern hebrew. *Preprint*, arXiv:2308.16687.

Arnon Turetzky, Or Tal, Yael Segal, Yehoshua Dissen, Ella Zeldes, Amit Roth, Eyal Cohen, Yosi Shrem, Bronya R Chernyak, Olga Seleznova, and 1 others. 2024. Hebdb: a weakly supervised dataset for hebrew speech processing. In *Proc. Interspeech 2024*, pages 1360–1364.

Werner Weinberg. 1966. Spoken israeli hebrew: Trends in the departures from classical phonology. *Journal of Semitic Studies*, 11(1):40–68.

Ella Zeldes, Or Tal, and Yossi Adi. 2025. Enhancing tts stability in hebrew using discrete semantic units. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Jian Zhu, Cong Zhang, and David Jurgens. 2022. Byt5 model for massively multilingual grapheme-to-phoneme conversion. *arXiv preprint arXiv:2204.03067*.

Appendix

A ILSpeech Dataset Details

The ILSpeech dataset consists of approximately two hours of Hebrew speech from two speakers. The speech content includes diverse topics covering science, technology, history, and everyday conversational speech. The audio was originally recorded in a studio environment at 44kHz, then enhanced using Adobe Enhance Speech v2[14] and normalized to 22.05kHz. It was then segmented by automatically splitting recordings at silence boundaries with manual refinement, resulting in disjoint segments of 4-14 seconds each. Hebrew and IPA annotations were produced manually by the authors.

We release ILSpeech under a non-commercial license with ethical use requirements. For more details, please see the dataset page[15].

B Implementation Details

B.1 G2P Conventions

In our enhanced vocalization scheme, the superscript angle marking stress is only used on non-final stressed symbols (since the most common stress pattern in Hebrew is final stress).

In our paper we provide IPA transcriptions using common linguistic conventions for clarity, while the implementation in our code and demo (and shown in Appendix C.3) differs slightly in the following conventions: Firstly, it uses the symbols /ʁ χ/ rather than /r x/ as a more narrow transcription of the sounds usually realized as uvular in Modern Hebrew. Secondly, it indicates stress immediately before the stressed vowel rather than before the entire stressed syllable. These are both illustrated in the word רוּחַ, transcribed as /ʁˈuaχ/ or /ˈruax/ depending on convention.

[14] https://podcast.adobe.com/en/enhance

[15] https://huggingface.co/datasets/thewh1teagle/ILSpeech

B.2 G2P Pseudo-GT Construction

For pseudo-GT construction, we use the IsraParlTweet dataset[16] of approximately 5M lines of Hebrew text from parliamentary proceedings and Tweets (Mor-Lan et al., 2024). We enhance the original text with diacritics using the Dicta system, including stress marking for words with second or third syllable stress, vertical bars to mark prefix letters, and programmatically apply vocal shva with expert editing.

B.2.1 G2P Model Architecture Details

The Phonikud architecture consists the following components: The base diacritization model is

As our base diacritization model, we use an open-weights DictaBERT model (Shmidman et al., 2023), using a checkpoint[17] which has been fine-tuned for Hebrew diacritization. This is an encoder-only ~300M parameter model with linear classification heads for Hebrew diacritics.

We enhance this with an additional head for predicting additional symbols (stress, vocal shva, and prefix markers). This is implemented as a two-layer MLP with hidden dimension 256, intermediate ReLU activation, outputting logits for these new symbols. These additional layers add a negligible number of parameters to the model, allowing for efficient training and inference. They are randomly initialized and trained as described in the main paper.

B.3 G2P Training Details

We train our G2P model for approximately six epochs, until early stopping is triggered. We use a batch size of 256 and a learning rate of 5e-3, with 5% of the data reserved for validation. We train the model on a single GPU.

B.4 TTS Training Details

Our Piper-based model is initialized from a pre-trained English checkpoint[18], and trained with learning rate 2e-4 and batch size 24 for ~10K epochs. Our StyleTTS2-based model is also initialized from a pre-trained English checkpoint[19]

[16] https://huggingface.co/datasets/guymorlan/IsraParlTweet

[17] dicta-il/dictabert-large-char-menaked

[18] https://huggingface.co/datasets/rhasspy/piper-checkpoints/resolve/main/en/en_US/ryan/medium/epoch=4641-step=3104302.ckpt

[19] https://huggingface.co/dangtr0408/StyleTTS2-lite

and trained with learning rate 1e-4 and batch size 5 for 123 epochs. For ablations, we use a Piper model with the previous hyperparameters and training for ~500 epochs for a light-weight comparison. All training is conducted on a single GPU.

B.5 Evaluation Setup

To compute Word Error Rate (WER) and Character Error Rate (CER) metrics, we use a Whisper-based Hebrew ASR model[20], calculated only only over Hebrew words. Real-Time Factor (RTF) was calculated as the ratio of processing time to audio duration (T/D), where RTF $<$ 1.0 indicates real-time capability. All RTF measurements include the complete processing pipeline (including diacritization when used) for consistent comparison across models. This is conducted on standard consumer hardware (macOS M1) without GPU acceleration to reflect real-world edge computing scenarios.

We evaluate on 100 samples from SASPEECH, selected randomly from samples without special characters containing at least six words.

C Additional Results

C.1 Audio Results and Comparisons

For audio results of our model and comparisons to existing Hebrew TTS models, please refer to our project page: https://phonikud.github.io

C.2 Comparison to Large, One-Stage Models

The large, open-source Hebrew TTS models LoTHM (Zeldes et al., 2025) and HebTTS (Roth et al., 2024) take unvocalized Hebrew text as input directly. They exhibit significantly slower inference speeds than our method and other models we compare to (RTF of 84.75 for LoTHM and 25.44 for HebTTS, with the same hardware setup), vs. 0.09 for our method with Piper. Since RTF > 1 indicates performance slower than real-time, we focus our main evaluation on models that are close to meeting the performance constraints of practical edge deployment scenarios.

C.3 North Wind and Sun - Hebrew Phonetic Transcriptions

Figure 4 presents the Hebrew version of the "The North Wind and the Sun" fable reproduced from the IPA Handbook (Association, 1999), with ground-truth phonemes, the output of our Phonikud G2P system and those of alternative G2P systems for comparison. Note that here we use the IPA conventions described in Appendix B.1.

[20] https://huggingface.co/ivrit-ai/whisper-large-v3-turbo-ct2

Original Unvocalized Hebrew	רוח הצפון והשמש התוכחו ביניהם מי מהם חזק יותר. גמרו, כי את הנצחון ינחל מי שיצליח לפשוט מעל עובר אורח את בגדיו. פתח רוח הצפון ונשב בחזקה. הידק האדם את בגדיו אל גופו. אז הסתער עליו הרוח ביתר עוז, אך האדם, משהוסיף הקור לענותו, לבש מעיל עליון על בגדיו. נואש ממנו הרוח ומסרו בידי השמש. תחילה זרח עליו השמש ברכות, והאדם הסיר את בגדו העליון מעליו. הגביר השמש את חומו, עד שלא יכול האדם לעמוד בפני השרב, ופשט את בגדיו ונכנס לתוך הנהר, שהיה בקרבת מקום, כדי לרחוץ במימיו.
Ground-Truth Vocalized Hebrew	רוּחַ הַצָּפוֹן וְהַשֶּׁמֶשׁ הִתְוַכְּחוּ בֵּינֵיהֶם מִי מֵהֶם חָזָק יוֹתֵר. גָּמְרוּ, כִּי אֶת הַנִּצָּחוֹן יִנְחַל מִי שֶׁיִּצְלִיחַ לִפְשׁוֹט מֵעַל עוֹבֵר אֹרַח אֶת בְּגָדָיו. פָּתַח רוּחַ הַצָּפוֹן וְנָשַׁב בְּחָזְקָה. הִדֵּק הָאָדָם אֶת בְּגָדָיו אֶל גּוּפוֹ. אָז הִסְתַּעֵר עָלָיו הָרוּחַ בְּיֶתֶר עוֹז, אַךְ הָאָדָם, מִשֶּׁהוֹסִיף הַקּוֹר לְעַנּוֹתוֹ, לָבַשׁ מְעִיל עֶלְיוֹן עַל בְּגָדָיו. נוֹאַשׁ מִמֶּנּוּ הָרוּחַ וּמְסָרוֹ בִּידֵי הַשֶּׁמֶשׁ. תְּחִילָה זָרַח עָלָיו הַשֶּׁמֶשׁ בְּרָכוֹת, וְהָאָדָם הֵסִיר אֶת בִּגְדּוֹ הָעֶלְיוֹן מֵעָלָיו. הִגְבִּיר הַשֶּׁמֶשׁ אֶת חֲמוֹ, עַד שֶׁלֹּא יָכֹל הָאָדָם לַעֲמוֹד בִּפְנֵי הַשָּׁרָב, וּפָשַׁט אֶת בְּגָדָיו וְנִכְנַס לְתוֹךְ הַנָּהָר, שֶׁהָיָה בְּקִרְבַת מָקוֹם, כְּדֵי לִרְחֹץ בְּמֵימָיו.
Ground-Truth IPA	ʁˈuaχ hatsafˈon vehaʃˈemeʃ hitvakχˈu bejnehˈem mˈi mehˈem χazˈak jotˈeʁ. gamʁˈu, kˈi ʔˈet hanitsaχˈon jinχˈal mˈi ʃejatslˈiaχ lifʃˈot meʔˈal ʔovˈeʁ ʔoʁaχ ʔˈet bgadˈav. patˈaχ ʁˈuaχ hatsafˈon venaʃˈav beχozkˈa. hidˈek haʔadˈam ʔˈet bgadˈav ʔˈel gufˈo. ʔˈaz histaʔˈeʁ ʔalˈav haʁˈuaχ bejˈeteʁ ʔˈoz, ʔˈaχ haʔadˈam, miʃehosˈif hakˈoʁ laʔanotˈo, lavˈaʃ meʔˈil ʔeljˈon ʔˈal bgadˈav. noʔˈaʃ mimˈenu haʁˈuaχ umsaʁˈo bidˈej haʃˈemeʃ. tχilˈa zaʁˈaχ ʔalˈav haʃˈemeʃ beʁakˈut, vehaʔadˈam hesˈiʁ ʔˈet bigdˈo haʔeljˈon meʔalˈav. higbˈiʁ haʃˈemeʃ ʔˈet χumˈo, ʔˈad ʃelˈo jaχˈol haʔadˈam laʔamˈod bifnˈej haʃaʁˈav, ufaʃˈat ʔˈet bgadˈav veniχnˈas letˈoχ hanahˈaʁ, ʃehajˈa bekiʁvˈat makˈom, kedˈej liʁχˈots bemejmˈav
Phonikud (Ours)	ʁˈuaχ hatsafˈon vehaʃˈemeʃ hitukχˈu bnehˈem mˈi mehˈem χazˈak jotˈeʁ. gamʁˈu, kˈi ʔˈet hanitsaχˈon jinχˈel mˈi ʃejatslˈiaχ lifʃˈot meʔˈal ʔovˈeʁ ʔoʁaaχ ʔˈet bgadˈav. patˈaχ ʁˈuaχ hatsafˈon venaʃˈav baχazkˈa. hidˈek haʔadˈam ʔˈet bgadˈav ʔˈel gufˈo. ʔˈaz histaʔˈeʁ ʔalˈav haʁˈuaχ bejˈeteʁ ʔˈoz, ʔˈaχ haʔadˈam, miʃehosˈif hakˈoʁ laʔanotˈo, lavˈaʃ meʔˈil ʔeljˈon ʔˈal bgadˈav. noʔˈaʃ mimˈenu haʁˈuaχ umasʁˈo bidˈej haʃˈemeʃ. tχilˈa zaʁˈaχ ʔalˈav haʃˈemeʃ beʁakˈot, vehaʔadˈam hesˈiʁ ʔˈet bgadˈo haʔeljˈon meʔalˈav. higbˈiʁ haʃˈemeʃ ʔˈet χamˈo, ʔˈad ʃelˈo jaχˈol haʔadˈam laʔamˈod bifnˈej haʃaʁˈav, ufaʃˈat ʔˈet hagadˈav veniχnˈas letˈoχ hanahˈaʁ, ʃehajˈa bekiʁvˈat makˈom, kedˈej liʁχˈots bememˈav.
Dicta*	ʁuˈaχ hatsafˈon vehaʃemˈeʃ hitukχˈu bnehˈem mˈi mehˈem χazˈak jotˈeʁ. gamʁˈu, kˈi ʔˈet hanitsaχˈon jinχˈel mˈi ʃejatsliˈaχ lifʃˈot meʔˈal ʔovˈeʁ ʔoʁˈaχ ʔˈet bgadˈav. patˈaχ ʁuˈaχ hatsafˈon venaʃˈav baχazkˈa. hidˈek haʔadˈam ʔˈet bgadˈav ʔˈel gufˈo. ʔˈaz histaʔˈeʁ ʔalˈav haʁuˈaχ bjetˈeʁ ʔˈoz, ʔˈaχ haʔadˈam, miʃehosˈif hakˈoʁ laʔanotˈo, lavˈaʃ mʔˈil ʔeljˈon ʔˈal bgadˈav. noʔˈaʃ mimenˈu haʁuˈaχ umasʁˈo bidˈej haʃemˈeʃ. tχilˈa zaʁˈaχ ʔalˈav haʃemˈeʃ bʁaχˈot, vehaʔadˈam hesˈiʁ ʔˈet bgadˈo haʔeljˈon meʔalˈav. higbˈiʁ haʃemˈeʃ ʔˈet χamˈo, ʔˈad ʃelˈo jaχˈol haʔadˈam laʔamˈod bifnˈej haʃaʁˈav, ufaʃˈat ʔˈet hagadˈav veniχnˈas ltˈoχ hanahˈaʁ, ʃehajˈa bkiʁvˈat makˈom, kdˈej liʁχˈots bmemˈav.
Nakdimon*	ʁuˈaχ hatsafˈon vehaʃemˈeʃ hitvokχˈu bnehˈem mˈi mehˈem χazˈak jotˈeʁ. gamʁˈu, kˈi ʔˈet hanitsaχˈon jinχˈal mˈi ʃejatsliˈaχ lifʃˈot meʔˈal ʔovˈeʁ ʔoʁˈaχ ʔˈet bgadˈav. petˈaχ ʁuˈaχ hatsafˈon veneʃˈev baχazakˈa. hidˈek haʔadˈam ʔˈet bgadˈav ʔˈel gufˈo. ʔˈaz histaʔˈeʁ ʔalˈav haʁuˈaχ bjetˈeʁ ʔˈoz, ʔˈaχ haʔadˈam, miʃehosˈif hakˈoʁ laʔanotˈo, lavˈaʃ mʔˈil ʔeljˈon ʔˈal bgadˈav. noʔˈaʃ mimenˈu haʁuˈaχ umasʁˈo bidˈej haʃemˈeʃ. tχilˈa zaʁˈaχ ʔalˈav haʃemˈeʃ bʁaχˈot, vehaʔadˈam hesˈiʁ ʔˈet bgdˈo haʔeljˈon meʔalˈav. higvˈiʁ haʃamˈeʃ ʔˈet χemˈo, ʔˈad ʃelˈo jaχˈol haʔadˈam laʔamˈod bifnˈej haʃaʁˈav, ufaʃˈat ʔˈet hagadˈav veniχnˈas ltˈoχ hanahˈaʁ, ʃehajˈa bkiʁvˈat makˈom, kdˈej liʁχˈots bmejamˈav.
Espeak	rvχ htsfvnə vhʃmʃ htvχχv vnihem mi mhem χzk joter gmrv χi ʔt hntsχvnə jnχl mi ʃjtsliχ lfʃvt mʔl ʔvvr ʔvrχ ʔt vgdiv ftχ rvχ htsfvnə vnʃv vχzka hidk hʔdm ʔt vgdiv ʔl gvfv ʔz hstʔr ʔliv hrvχ vitr ʔvz ʔχ hʔdm mʃhvsif hkor lʔnotv lvʃ mʔil ʔlivnə ʔl vgdiv nvʔʃ mmnv hrvχ vmsrv vjdi hʃmʃ tχila zrχ ʔliv hʃmʃ vrχvt vhʔdm hsir ʔt vgdu hʔlivnə mʔliv hgvjr hʃmʃ ʔt χmv ʔd ʃlʔ jaχol hʔdm lʔmvd vfni hʃrv vfʃt ʔt hgdiv vnχns ltvχ hnar ʃhih vkrvt mkom χdi lrχvts vmimiv
Goruut	riaχ hetsˈfun voʃmaʃ hθokχau benihem mi mahem χɛzek jɔr. gɛmɛruː, χi ʔat hɛntsaχˈno jnχle mi ʃitsiˈliχ lefˈʃot maʕal ʕoˈvɛr ɔrɑχ ɛt bagdiː. petɑx ʁuʔaχ hetsˈfun onʃb βχzkɑ. ajadˈˈk hadaːm ɛt bagdiː al goˈfu. ʔaz hesˈteʁ ʕaljov heruːx biˈtɛr ʔoz, ʔax hadaːm, mʃheɔsif hakuʁ leʕnuθu, lavʃ maˈʕil ʕaliːon ʕal bagdiː. nuˈaʃ mamɛno heruːx omaesro bidi heʃˈmɛʃ. θɪˈχilɑ zʁχ ʕaljov heʃˈmɛʃ brɔχɔt, vahem hesiʁ ɛt bagdo heʔalijon maʕaliː. hegˈbir heʃˈmɛʃ ɛt xmau, ʕad ʃala jakuːl hadaːm leˈʔumod beˈfi eʃrb, ɔpɛt ɛt hegdiˈɔ ʊɛnχɛns ɬoˈtɔχ henɛr, ʃaˈja bakarava maˈkum, χadiː loʁɑts ɛmiˈmio.
CharsiuG2P	rôtʰ dæʃtæːʌ otokomoɾo tataːltʰeː bɔritɛɔ mîː mætəɲ tʰæʒ̃ʊ̃ː zeːtəbɔ dæmboː tøy tʰêt tariztʰoːl jærtʰəl mîː kîfmitʰ lokoj mâːl tâmbæb æobʧ̂ tʰêt bəddijɔ tʰatʰ rôtʰ dæʃtæːʌ o̞lkʰo̞b bɛtʰʒaːa tæːdɔː taːdəm tʰêt bæddijo âl dottɛ æʒ destaaːb tælio deːboːtʰ beːtəb tæz æn taːdən mətədeːmiːr damaːjeːb laːreːta lok mæɒːyl tæmieʔ tâl bəddijɔ roaːk mæmro deːboːtʰ omməbo bidi takmakɛɪ tʰatʰəda zəbtʰ tælio takmak bəbtoːt odɒːdəm tasîb tʰêt bæddæ taːmijoʔ mæɒːmiɔ tadbiːb takmak tʰêt tʰəmaː tæd kəla zætol taːdəm laːmaːd bûri tatəba ofki tʰêt taddajo oɾɢələrs lətaːl tartab kɛdiːt boːbəbæt maːɔh tædi labtʰɔʔ bəminiɔ

Figure 4: Hebrew phonetic transcriptions of "The North Wind and the Sun" fable. We include our additional stress and vocal shva symbols in the vocalized Hebrew in the second row to illustrate the ground-truth pronunciation. The diacritization methods marked with an asterisk* are applied with our IPA conversion with defaults for ambiguous features like stress.