

HiGraph: Explicit Discrepancy Reasoning with Hierarchical Graph Propagation for Joint Pronunciation Assessment

Hyunku Kang  Eunseo Choi  Ireh Kim  Chanwoo Kim* 

Department of Artificial Intelligence, Korea University, Republic of Korea
{kakh000, ces4669, ksmkmskim, chanwcom}@korea.ac.kr

Abstract

Automated pronunciation assessment predicts pronunciation quality at the phone, word, and utterance levels. Recent joint models use both canonical and realized phone representations, but their mismatch is often encoded implicitly rather than preserved as a separate scoring path. This makes it difficult to trace how local phone errors affect word- and utterance-level scores. We propose HiGraph, which explicitly computes canonical-realized mismatch and propagates it through a hierarchical phone-to-word-to-utterance graph. This creates a structured discrepancy path from phone-level mismatch evidence to higher-level score representations. Under the same pipeline as our reproduced HMamba baseline, HiGraph improves word stress and utterance completeness on SpeechOcean762 (SO762). The learned word-level trace score also aligns with mispronunciation detection and diagnosis (MDD) ground truth. When words are ranked by this trace score, the top-ranked word contains a true mispronunciation in 81.5% of test utterances, and at least one of the top two words does so in 93.8% of test utterances. These results show that explicit discrepancy propagation can improve selected assessment dimensions while supporting feedback localization grounded in annotated mispronunciations.

1 Introduction

Automated pronunciation assessment (APA) is a core component of computer-assisted pronunciation training (CAPT) (El Kheir et al., 2023). It predicts how well a learner pronounces a given utterance (Neumeyer et al., 1996). Early CAPT systems often assessed segmental and suprasegmental aspects separately, using phone-level mea-

sures such as GOP or formant-based cues for segmental accuracy (Witt and Young, 2000; Kim and Sung, 2001) and intonation-oriented measures for suprasegmental quality (Kim and Sung, 2002). For pronunciation feedback to be actionable, a system must show where pronunciation quality changes across linguistic levels (Neri et al., 2002). It therefore needs scores at multiple levels, from phones and words to utterance-level aspects such as accuracy, completeness, fluency, and prosody.

Recently, self-supervised learning (SSL) techniques have been broadly adopted across various speech domains (Conneau et al., 2020; Hsu et al., 2021; Chen et al., 2022), achieving remarkable success in areas such as speaker verification and emotion classification (Kang et al., 2026; Choi et al., 2026). Researchers have increasingly applied SSL to the CAPT task, yielding competitive models such as HMamba (Chao and Chen, 2025) and MuFFIN (Yan et al., 2025). Another notable finding in recent work is that jointly modeling canonical and realized pronunciations is critical for performance; models combining acoustic features with canonical phone labels consistently outperform acoustic-only approaches (Gong et al., 2022). HMamba and MuFFIN also follow this paradigm by integrating acoustic and canonical information into a shared representation within a single network.

However, existing joint models can encode canonical-realized mismatch inside shared hidden layers, but they do not preserve it as a separate scoring path. As a result, mismatch-related information may contribute to score prediction, but it is hard to trace which phone error affected which word score or utterance score. This is a key problem for pronunciation feedback, because learners need to know where the score loss comes from.

This limitation also weakens structural guidance. When a learner substitutes or deletes a phone, the model does not maintain an explicit record of the local mismatch. Its effect on word- and utterance-

*Corresponding author

Code is publicly available at: <https://github.com/slp-lab-research/higraph>

level scores remains implicit inside learned representations rather than being passed through an explicit phone-to-word-to-utterance path.

We propose HiGraph to address this problem. HiGraph computes a phone-level discrepancy representation by comparing each realized phone embedding with its canonical counterpart using concatenation, difference, and product features. It then propagates these discrepancies through a phone-to-word-to-utterance graph, where learned gates control how much discrepancy information enters the main score representation. This makes scoring more traceable from phone to utterance.

Our contributions are threefold. First, we preserve canonical-realized mismatch as an explicit discrepancy path used in score formation, rather than leaving it only in shared hidden representations. Second, we propagate this discrepancy through a hierarchical phone-to-word-to-utterance graph with learned gated fusion. Third, we show that the resulting path supports structural traceability through score-sensitive masking and mispronunciation detection and diagnosis (MDD)-grounded feedback localization.

2 Background and Related Work

Early pronunciation assessment systems used goodness of pronunciation (GOP) features and shallow scoring models (Witt and Young, 2000). These methods predicted scores from local pronunciation cues, but they did not model structural links among phones, words, and utterances.

Hierarchical pronunciation assessment. Later models introduced stronger links across linguistic levels (Lin et al., 2020). GOPT (Gong et al., 2022) and 3M (Chao et al., 2022) predict phone, word, and utterance scores with multi-aspect and multi-granular representations. HiPAMA (Do et al., 2023a) models relations among scoring aspects within a hierarchical assessment framework. HierGAT (Yan and Chen, 2024) uses graph attention to pass information across linguistic units and assessment aspects. 3MH (Chao et al., 2023) adds finer units below the phone level to capture local pronunciation cues. These models show that linguistic hierarchy helps pronunciation assessment. However, they mainly refine score representations. They do not explicitly model how local pronunciation mismatch affects higher-level scores.

Joint assessment and diagnostic feedback. Recent models combine pronunciation assessment with mispronunciation detection. Joint-CAPT (Ryu et al., 2023), JAM (He et al., 2024), MuFFIN (Yan et al., 2025), and HMamba (Chao and Chen, 2025) show that canonical phone information and phone-level diagnostic supervision can improve pronunciation modeling. These models are important steps toward feedback-oriented CAPT (Shi et al., 2020). These models can encode the relation between learner speech and canonical phones in shared hidden representations. However, they do not preserve it as a separate discrepancy representation or pass it through the scoring hierarchy.

Remaining gap. Prior work shows the value of hierarchical scoring and joint diagnostic supervision, and implicit joint models can learn mismatch-related information. However, canonical-realized mismatch is not preserved as a dedicated path from local evidence to higher-level scores. HiGraph addresses this gap by making the mismatch signal explicit and propagating it through the scoring hierarchy.

3 HiGraph

Figure 1 gives an overview of HiGraph. HiGraph adds a graph neural network (GNN) reasoning layer to a hierarchical APA backbone while keeping the backbone inputs and prediction heads unchanged. It uses a dual-path graph: one path refines score representations over the phone-word-utterance hierarchy, and the other computes phone-level canonical-realized mismatch and passes it to word and utterance nodes. In Figure 1, each refinement block denotes the gated graph update in Eq. (2). Learned gates fuse the two paths at the word and utterance levels.

HMamba Backbone. HiGraph uses HMamba as its backbone, a hierarchical bidirectional Mamba-based (Gu and Dao, 2023) model that predicts phone-, word-, and utterance-level scores from shared phone-aligned representations.

Following HMamba, each phone input concatenates GOP, self-supervised learning (SSL), duration, and energy features (Kim et al., 2022). For phone position i , let x_i be the concatenated acoustic input, k_i be the canonical phone ID, t_i be the absolute position index, and b_i be the word-boundary tag for that position. Let W_{in} be a learned input projection, and let $e_{k_i}^{\text{cano}}$, $e_{t_i}^{\text{pos}}$, and $e_{b_i}^{\text{bnd}}$ be learned

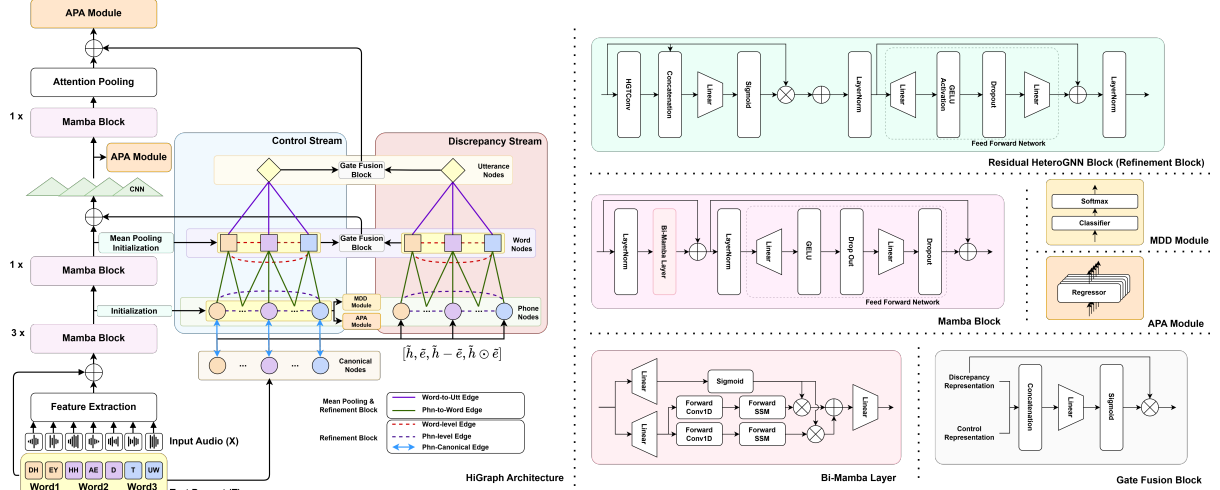


Figure 1: Overview of HiGraph. A hierarchical backbone first produces phone-aligned representations. HiGraph builds two coupled streams on top of these representations: a control stream that refines score representations over the phone-word-utterance hierarchy, and a discrepancy stream that computes the canonical-realized discrepancy input in Eq. (3) and propagates it to word and utterance nodes. Learned gates inject the discrepancy representations into the control stream before word- and utterance-level scoring.

canonical-phone, position, and boundary embeddings. We initialize the phone representation as

$$\mathbf{h}_i^{(0)} = \mathbf{W}_{\text{in}} \mathbf{x}_i + \mathbf{e}_{k_i}^{\text{cano}} + \mathbf{e}_{t_i}^{\text{pos}} + \mathbf{e}_{b_i}^{\text{bnd}}. \quad (1)$$

The sequence is then processed by bidirectional Mamba blocks to produce phone-aligned hidden states (Gu et al., 2021). After the three phone-stage blocks, these hidden states initialize phone nodes. After the fourth Mamba block, word nodes are initialized by mean pooling phone-aligned hidden states within each word. Utterance control nodes are initialized by mean pooling refined word nodes within each utterance. The original HMamba sequence path is kept and still produces the pooled utterance representation used for final scoring.

Control stream. The control stream refines backbone embeddings with HGTCConv (Hu et al., 2020) layers over a phone-word-utterance graph. Edges connect all phone pairs within the same word, phones to their parent word, all word pairs within an utterance, and words to the utterance node (Kipf and Welling, 2016). Each block computes an aggregated message (Gilmer et al., 2017) $\mathbf{m}_v$ for node v and applies a gated residual update to its node representation $\mathbf{h}_v$:

$$\begin{aligned} \mathbf{g}_v^{\text{upd}} &= \sigma(\mathbf{W}_g^{\text{upd}}[\mathbf{h}_v; \mathbf{m}_v]), \\ \mathbf{h}_v &\leftarrow \mathbf{h}_v + \mathbf{g}_v^{\text{upd}} \odot \mathbf{m}_v. \end{aligned} \quad (2)$$

We use this gated graph update as the refinement block in both the control and discrepancy streams (Li et al., 2015).

Discrepancy stream. The discrepancy stream runs in parallel to the control stream and makes realized-canonical differences explicit across the hierarchy. Here, realized phones are inferred from learner speech, while canonical phones come from the expected pronunciation. For the discrepancy stream, we maintain a learnable canonical embedding table $E^{\text{disc}} \in \mathbb{R}^{V \times d}$, where V is the number of phone types. For phone i with canonical type k_i , the discrepancy-stream canonical embedding is $e_{k_i} = E^{\text{disc}}[k_i, :]$.

Before computing the discrepancy, we refine the realized and canonical embeddings with one mutual graph refinement block. For each phone position i , let $\mathbf{h}_i$ be the current phone-level hidden state and let e_{k_i} be its discrepancy-stream canonical embedding. We build a bipartite graph, with one part containing realized nodes $\{\mathbf{h}_i\}$ and the other containing canonical nodes $\{e_{k_i}\}$. We add directed edges in both directions and apply a shared HGTCConv operator to this graph. This yields refined embeddings $\tilde{\mathbf{h}}_i$ and $\tilde{e}_{k_i}$ for the learner-speech and canonical sides, respectively. The discrepancy input for phone i is then formed by concatenating four terms: the two refined embeddings, their element-wise difference, and their element-wise product:

$$\mathbf{d}_i = [\tilde{\mathbf{h}}_i; \tilde{\mathbf{e}}_{k_i}; \tilde{\mathbf{h}}_i - \tilde{\mathbf{e}}_{k_i}; \tilde{\mathbf{h}}_i \odot \tilde{\mathbf{e}}_{k_i}] \in \mathbb{R}^{4d}. \quad (3)$$

The difference term captures how the realized phone moves away from the canonical phone, while the product term captures which feature dimensions still match or interact. These terms form a direct mismatch signal. Let $W_{\delta,1}$ and $W_{\delta,2}$ be learned projection matrices. The combined vector is projected back to dimension d :

$$\delta_i = \text{LayerNorm}(\mathbf{W}_{\delta,2} \text{GELU}(\mathbf{W}_{\delta,1} \mathbf{d}_i)). \quad (4)$$

Phone-level discrepancy representations are pooled by word using mean aggregation (Hamilton et al., 2017). For word j with phone index set P_j :

$$\delta_j^w = \frac{1}{|P_j|} \sum_{i \in P_j} \delta_i. \quad (5)$$

Inter-word graph refinement blocks then update the word-level discrepancy nodes $\{\delta_j^w\}$. For notation simplicity, δ_j^w denotes the refined word-level discrepancy representation after this inter-word graph refinement. Let $\mathcal{W}$ denote the set of word indices in an utterance. The refined word representations are then pooled to an utterance-level discrepancy node and further refined through a word-to-utterance graph layer:

$$\delta^u = \frac{1}{|\mathcal{W}|} \sum_{j \in \mathcal{W}} \delta_j^w. \quad (6)$$

Gated fusion. The discrepancy stream is fused into the control stream at the word level and at the utterance level through learned gates. Let r_i denote the phone-aligned word-stage hidden state after control-stream refinement, and let $j(i)$ be the parent word index of phone i . Let w_j^{ctrl} be the control-stream word representation for word j , and let W_g^w be a learned word-level gate projection. The word-level gate is defined as

$$g_j^w = \sigma(W_g^w[w_j^{\text{ctrl}}; \delta_j^w]). \quad (7)$$

The gated discrepancy is then injected back into the phone-aligned word-stage sequence:

$$\tilde{r}_i = r_i + g_{j(i)}^w \odot \delta_{j(i)}^w. \quad (8)$$

The sequence-based utterance representation is formed through score-attention pooling over the

phone-aligned hidden states after the utterance-stage backbone block. Let z_t denote the hidden state at position t . Let $\hat{p}_t$ denote the phone-level score prediction at position t , and let $\hat{w}_t^{\text{acc}}$, $\hat{w}_t^{\text{str}}$, and $\hat{w}_t^{\text{tot}}$ denote the word-level score predictions aligned to the same position. We then form

$$q_t = [\hat{p}_t; \hat{w}_t^{\text{acc}}; \hat{w}_t^{\text{str}}; \hat{w}_t^{\text{tot}}]. \quad (9)$$

Attention weights are computed from this local score vector:

$$a_t = \frac{\exp(f_{\text{att}}(q_t))}{\sum_s \exp(f_{\text{att}}(q_s))}, \quad (10)$$

where $f_{\text{att}}(\cdot)$ is a learned scalar scoring function, and the denominator sums over all phone-aligned positions in the utterance. The sequence-based utterance representation is then

$$u^{\text{seq}} = \sum_t a_t z_t. \quad (11)$$

We then combine the sequence-based utterance representation with the control-stream utterance node by simple addition, i.e., $u = u^{\text{seq}} + u^{\text{ctrl}}$.

At the utterance level, we inject the discrepancy representation into the combined utterance representation through a learned gate.

Let u^{ctrl} and δ^u denote the control-stream utterance representation and the utterance-level discrepancy representation, respectively, and let W_g^u be a learned utterance-level gate projection. The utterance-level gate is defined as

$$g^u = \sigma(W_g^u[u^{\text{ctrl}}; \delta^u]), \quad (12)$$

and the final utterance representation is

$$\tilde{u} = u + g^u \odot \delta^u. \quad (13)$$

The gates learn how much discrepancy information should affect each score representation.

Score heads and training. Phone heads are applied after phone-level control refinement. Word heads are applied after word-level fusion, and utterance heads are applied after sequence pooling and utterance-level fusion. We use mean squared error for phone-, word-, and utterance-level assessment scores, and add an auxiliary phone-recognition loss to preserve phone identity. Let $\mathcal{L}_{\text{phn}}$, $\mathcal{L}_{\text{wrd}}$, $\mathcal{L}_{\text{utt}}$, and $\mathcal{L}_{\text{aux}}$ denote the phone-level, word-level, utterance-level, and auxiliary losses, respectively. Let λ_{phn} , λ_{wrd} , λ_{utt} , and λ_{aux} denote their loss weights. The total loss is:

$$\mathcal{L} = \lambda_{\text{phn}}\mathcal{L}_{\text{phn}} + \lambda_{\text{wrđ}}\mathcal{L}_{\text{wrđ}} + \lambda_{\text{utt}}\mathcal{L}_{\text{utt}} + \lambda_{\text{aux}}\mathcal{L}_{\text{aux}}. \quad (14)$$

We use the same loss weights as the HMamba training recipe. In our default setting, $\mathcal{L}_{\text{aux}}$ follows HMamba and applies an auxiliary cross-entropy loss to realized phone labels.

This follows HMamba’s score-attention pooling design while allowing HiGraph to inject utterance-level discrepancy after hierarchical graph refinement.

4 Experimental Setup

We evaluate HiGraph on SpeechOcean762 (SO762) (Zhang et al., 2021), a standard benchmark for non-native English pronunciation assessment. SO762 provides phone-, word-, and utterance-level APA scores together with realized-phone labels for MDD evaluation. To our knowledge, it is the only public benchmark with the full annotation set required for this study. This enables a controlled evaluation of both the discrepancy-to-score pathway and its agreement with MDD ground truth. We follow the standard train/test split.

Input Representation. We follow the same input configuration as HMamba to isolate the effect of the proposed graph reasoning module. Each input uses GOP acoustic features, self-supervised features from wav2vec2-large-xlsr-53 (Conneau et al., 2020), HuBERT-large-ll60k (Hsu et al., 2021), and WavLM-large (Chen et al., 2022), together with low-level raw acoustic features. Canonical phone IDs and word boundary tags are also included.

Comparison Models. We compare HiGraph with prior pronunciation assessment models, including GOPT, HiPAMA, 3M, HierGAT, 3MH, Joint-CAPT, JAM, MuFFIN, and HMamba. For direct comparison, we reproduce HMamba under the same input features, prediction heads, and training recipe used by HiGraph. Since HiGraph is built on top of HMamba, we use HMamba (rep.) as the main baseline. We also report the original HMamba result in Table 1 because our reproduced scores do not match all originally reported metrics. To avoid overclaiming, all improvement claims and significance tests are made only against HMamba (rep.). Following HMamba, we select the best checkpoint by validation phone-level MSE unless stated otherwise. To test whether the proposed module is specific to HMamba, we additionally reproduce

GOPT and attach the HiGraph module while keeping its input features, prediction heads, and training recipe unchanged. Both models are evaluated over five runs, and the results are reported in Appendix A.6.

Evaluation Metrics. For APA, we report phone-level mean squared error (MSE) and Pearson correlation coefficient (PCC), and word- and utterance-level PCC (Do et al., 2023b). Following prior work, we report MSE only at the phone level, while word- and utterance-level assessment quality is reported with PCC.

For MDD, we report recall, precision, F1, false acceptance rate (FAR), false rejection rate (FRR), diagnosis error rate (DER), and phone error rate (PER) (Li et al., 2016). Unless otherwise noted, checkpoints are selected by validation phone-level MSE.

Implementation Details. HiGraph is implemented as a graph reasoning layer on top of hierarchical backbone representations, not as a replacement for the acoustic backbone. We use HMamba as the backbone and keep its input setting unchanged for fair comparison. We train for 25 epochs using Adam with the HMamba tri-stage learning-rate (LR) schedule: peak LR = 2×10^{-3} , initial LR scale = 10^{-2} , final LR = 10^{-4} , and phase ratio = (0.4, 0.4, 0.2). The train/test batch sizes are 50/250. Unless otherwise specified, we select the best checkpoint by validation phone-level MSE. All experiments were run on a single NVIDIA RTX 5090 GPU. For the computational comparison, we measure training time per epoch, peak GPU memory, and inference latency for HMamba and HiGraph under the same implementation and batch setting. HMamba (rep.) has 2.49M trainable parameters, while HiGraph has 5.26M.

5 Results

This section evaluates HiGraph in four steps: assessment and MDD performance, aspect-specific error analysis, traceability and feedback localization, and structural analyses of the proposed discrepancy path.

5.1 Assessment and Diagnosis Results

Pronunciation assessment. Table 1 evaluates whether explicit discrepancy reasoning improves the reproduced HMamba backbone and provides

Table 1: Pronunciation assessment results on SO762. We report phone-level MSE/PCC and word- and utterance-level PCC. Standard deviations over five runs are shown in parentheses. Bold and underline mark the best and second-best results. Significance is tested between HiGraph and HMamba (rep.) using approximate randomization ($p < 0.001$). HMamba is the reported result, and HMamba (rep.) is our reproduced baseline.

Model	Phoneme-level		Word-level aspects			Utterance-level aspects				
	MSE ↓	PCC ↑	Acc ↑	Stress ↑	Total ↑	Acc ↑	Comp ↑	Flu ↑	Prosody ↑	Total ↑
GOPT (Gong et al., 2022)	0.085 (0.001)	0.612 (0.003)	0.533 (0.004)	0.291 (0.030)	0.549 (0.002)	0.714 (0.004)	0.155 (0.039)	0.753 (0.008)	0.760 (0.006)	0.742 (0.005)
HiPAMA (Do et al., 2023a)	0.084 (0.001)	0.616 (0.004)	0.575 (0.004)	0.320 (0.021)	0.591 (0.004)	0.730 (0.002)	0.276 (0.177)	0.749 (0.001)	0.751 (0.002)	0.754 (0.002)
3M (Chao et al., 2022)	0.078 (0.001)	0.656 (0.005)	0.598 (0.005)	0.289 (0.033)	0.617 (0.005)	0.760 (0.004)	0.325 (0.141)	0.828 (0.006)	0.827 (0.008)	0.796 (0.004)
HierGAT (Yan and Chen, 2024)	0.073 (0.001)	0.683 (0.004)	0.648 (0.003)	0.327 (0.011)	0.663 (0.002)	0.798 (0.002)	<u>0.531</u> (0.047)	0.840 (0.002)	0.833 (0.002)	0.821 (0.002)
3MH (Chao et al., 2023)	0.071 (0.001)	0.693 (0.004)	0.682 (0.005)	0.361 (0.098)	0.694 (0.007)	0.782 (0.003)	0.374 (0.115)	0.843 (0.003)	0.836 (0.004)	0.811 (0.004)
Joint-CAPT (Ryu et al., 2023)	—	—	—	—	—	0.719	—	0.775	0.773	0.743
JAM (He et al., 2024)	0.076 (0.002)	0.664 (0.001)	0.622 (0.012)	0.241 (0.034)	0.638 (0.005)	0.773 (0.007)	0.205 (0.080)	0.831 (0.004)	0.829 (0.004)	0.805 (0.004)
MuFFIN (Yan et al., 2025)	<u>0.063</u> (0.002)	0.742 (0.006)	0.705 (0.004)	0.315 (0.033)	0.714 (0.004)	<u>0.807</u> (0.003)	0.768 (0.049)	0.841 (0.004)	0.832 (0.004)	0.830 (0.002)
HMamba (Chao and Chen, 2025)	0.062	0.739	0.708	<u>0.366</u>	<u>0.718</u>	<u>0.807</u>	0.278	<u>0.848</u>	0.843	<u>0.829</u>
HMamba (rep.) (Chao and Chen, 2025)	0.062 (0.001)	0.735 (0.004)	<u>0.709</u> (0.005)	0.327 (0.051)	<u>0.718</u> (0.006)	0.805 (0.004)	0.248 (0.054)	0.843 (0.006)	<u>0.837</u> (0.006)	0.825 (0.005)
HiGraph	0.062 (0.005)	<u>0.741</u> (0.002)	0.711 (0.005)	0.371 (0.055)	0.720 (0.005)	0.808 (0.004)	0.339 (0.072)	0.852 (0.007)	0.843 (0.007)	0.830 (0.005)

a trace score for feedback localization. The main scoring gains are on word stress and utterance completeness, which are closely related to local mismatch and missing or weakly realized phones. HiGraph improves word stress from 0.327 to 0.371 and utterance completeness from 0.248 to 0.339 over HMamba (rep.). Changes in phone PCC and word accuracy are smaller, suggesting that the main effect is concentrated on stress and completeness. Because HMamba (rep.) is the matched baseline under our pipeline, our direct improvement claims are limited to this baseline.

Aspect-specific error analysis. To examine why the main gains appear on word stress and utterance completeness, we compare prediction errors on groups related to local pronunciation mismatch. For utterance completeness, we group utterances by annotated deletion or substitution errors. For word stress, we group words by whether they receive a stress penalty. We report mean absolute error (MAE), where lower is better.

For utterance completeness, the improvement is larger when deletion or substitution errors are present than when no phone-level error is annotated. For word stress, the improvement is also

Table 2: Aspect-specific error analysis on SO762. For utterance-level groups, MAE is computed for utterance completeness. For word-level groups, MAE is computed for word stress. Deletion and substitution groups may overlap.

Group	<i>n</i>	HMamba	HiGraph	Imp.
<i>Utterance completeness</i>				
Deletion errors	184	1.028	1.002	0.026
Substitution errors	385	1.021	0.972	0.049
No phone-level error	2067	0.574	0.568	0.006
<i>Word stress</i>				
Stress penalty	127	2.518	2.265	0.253
No stress penalty	15839	0.421	0.398	0.024

larger for words with stress penalties. These results support the view that HiGraph mainly helps when local canonical-realized mismatch provides useful evidence for higher-level scores.

Mispronunciation detection and diagnosis. Table 3 shows that HiGraph maintains MDD performance, improving F1 from 63.05 to 63.56 and reducing DER from 53.24 to 52.54.

Table 3: Mispronunciation detection and diagnosis results on SO762. RE, PR, and F1 denote detection recall, precision, and F1. FAR, FRR, DER, and PER denote false acceptance rate, false rejection rate, diagnosis error rate, and phone error rate. Higher F1 and lower DER/PER are better.

Model	Detection			Diagnosis			PER ↓
	RE ↑	PR ↑	F1 ↑	FAR ↓	FRR ↓	DER ↓	
Joint-CAPT (Ryu et al., 2023)	91.60	26.90	41.50	—	—	—	9.93
JAM (He et al., 2024)	34.76	61.60	45.01	64.32	0.58	45.23	2.81
MuFFIN (Yan et al., 2025)	64.33	66.89	65.99	35.67	0.97	60.97	2.36
HMamba (rep.) (Chao and Chen, 2025)	62.28	63.83	63.05	37.72	1.01	53.24	2.83
HiGraph	62.81	64.33	63.56	37.19	0.97	52.54	2.81

5.2 Traceability Analysis

We test whether the learned discrepancy path supports feedback localization by defining a word-level trace score C_j as a gate-weighted score deficit. To compute C_j , let $\bar{g}_j^w$ be the mean activation over the feature dimension of the word-level gate:

$$\bar{g}_j^w = \frac{1}{d} \sum_{m=1}^d g_{j,m}^w. \quad (15)$$

Given the predicted word-level total score $\hat{s}_j^{\text{tot}}$, we define the score deficit as

$$\ell_j = \max(0, 10 - \hat{s}_j^{\text{tot}}). \quad (16)$$

We use 10 because the word-level total score in SO762 is defined on a 0–10 scale. Thus, ℓ_j measures how far the predicted score is from the maximum attainable score. The trace score is then

$$C_j = \bar{g}_j^w \cdot \ell_j. \quad (17)$$

Table 4: Utterance-score change under word-masking interventions. For each utterance, five words are selected uniformly, by lowest trace score C_j , or by highest C_j . Selected words are masked by zeroing δ_j^w . Mean, median, and interquartile range of the absolute score change Δ are reported.

Strategy	Mean Δ	Median Δ	IQR
Uniform	0.054	0.048	0.032–0.069
Lowest- C_j	0.046	0.035	0.016–0.063
Highest- C_j	0.067	0.059	0.032–0.096

Masking intervention. We select words based on the trace score C_j and mask them by zeroing their discrepancy representation δ_j^w . Table 4 shows the resulting absolute change in utterance score. Masking words with the highest C_j scores causes a mean change of 0.067. This is larger than 0.054 for randomly selected words and 0.046 for words with the lowest C_j scores. This consistent pattern supports the view that the trace score is linked to the score-forming path, rather than serving only as a post-hoc ranking score.

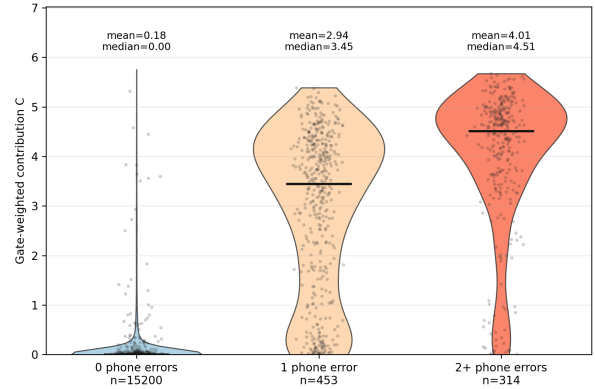


Figure 2: Trace score $C_j = \bar{g}_j^w \cdot \ell_j$ by phone error count per word on SO762. Horizontal bars show medians. Higher error counts are associated with higher trace scores.

Error alignment. Figure 2 shows the distribution of the trace score C_j grouped by phone error count per word. Words with zero phone errors have a median C_j of 0.00. Words with one phone error have a median C_j of 3.45, and words with two or more phone errors have a median C_j of 4.51. The trace score increases as the number of phone errors increases, suggesting that C_j tracks word-level error burden. Appendix A.1 further shows that this pattern holds across substitution, deletion, and mixed phone error types.

Feedback localization. Beyond internal traceability, we test whether the trace score C can rank a small set of words for learner feedback. We first compare it with its two components, the word-level gate activation $\bar{g}$ and the score deficit ℓ . We also evaluate post-hoc localization baselines on HMamba using word-score deficit, score-attention, integrated gradients, and occlusion. All ranking scores are compared with MDD ground truth, so this analysis measures agreement with annotated errors rather than validated classroom deployment.

Table 5 shows that HMamba already contains

Table 5: Feedback localization on SO762 utterances with at least one annotated mispronunciation. HMamba baselines derive word rankings from model outputs or post-hoc attribution. HiGraph rankings use the discrepancy gate, score deficit, or their product.

Ranking score	AUROC	AUPRC	Top-1	Top-2
Random	0.500	0.048	0.268	0.466
HMamba score deficit	0.949	0.652	0.806	0.930
HMamba score-attention	0.585	0.060	0.424	0.623
HMamba integrated gradients	0.752	0.199	0.743	0.856
HMamba occlusion	0.730	0.160	0.420	0.616
HiGraph gate activation $\bar{g}$	0.954	0.648	0.802	0.931
HiGraph score deficit ℓ	0.958	0.652	0.807	0.931
HiGraph trace $C = \bar{g} \cdot \ell$	0.960	0.661	0.815	0.938

useful error-location information. In particular, its word-score deficit gives strong localization performance. This supports the view that implicit models can encode mismatch-related information. However, the HiGraph trace gives the best result across AUROC, AUPRC, Top-1, and Top-2. Unlike gradient or occlusion attribution, the trace is obtained directly from the discrepancy path used during score formation.

The word-level gate activation $\bar{g}$ and score deficit ℓ are also strong ranking signals. Their Spearman correlation over all words is moderate ($\rho = 0.586$), showing that they capture related but not identical information. Their product gives the strongest overall localization result. Appendix A.5 further tests trace localization on L2-ARCTIC (Zhao et al., 2018). Because L2-ARCTIC lacks SO762-style multi-level APA scores, we use it only to test whether the trace score ranks words with annotated pronunciation errors.

Figure 3 shows one example trace. The words with phone errors have higher gate activation and larger trace scores C_j than error-free words. This visualizes the path from phone-level discrepancy to word-level contribution and utterance-level scoring.

5.3 Ablation and Design Analysis

We test whether the observed gains require the complete discrepancy path rather than a graph or local discrepancy module alone.

Structural analysis. Table 6 shows that neither graph reasoning nor local discrepancy modeling alone reproduces the full gain. The control-graph-only variant adds graph reasoning without an ex-

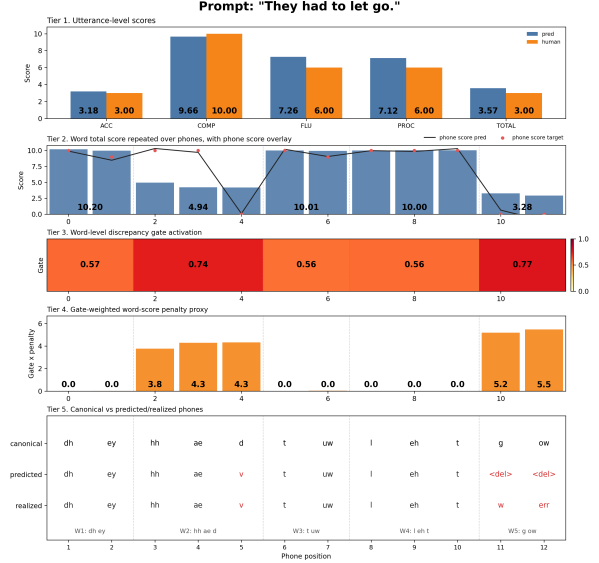


Figure 3: Trace visualization for one utterance. The figure shows utterance scores, word-score traces aligned to phone positions, mean gate activation $\bar{g}_j^w$, trace scores C_j , and phone alignments. Words with phone errors receive higher gate activation and trace scores.

Table 6: Structural analysis on SO762. Constructive variants add only part of the proposed path to HMamba. Removal variants remove one component from full HiGraph.

Model	Phn	W-Str	W-Tot	U-Comp	U-Tot
<i>Constructive variants</i>					
HMamba	0.735	0.327	0.718	0.248	0.825
+ control graph only	0.740	0.334	0.711	0.173	0.826
+ local discrepancy only	0.732	0.290	0.712	0.154	0.813
+ discrepancy + word pooling	0.735	0.283	0.718	0.165	0.818
Full HiGraph	0.741	0.371	0.720	0.339	0.830
<i>Removal variants</i>					
w/o discrepancy stream	0.740	0.354	0.711	0.203	0.826
w/o gated fusion	0.735	0.306	0.718	0.103	0.825
w/o graph refinement	0.732	0.296	0.715	0.084	0.819

PLICIT mismatch path, while the local-discrepancy variant computes mismatch without hierarchical propagation. Word-level pooling also remains weaker than full HiGraph. The removal variants show the complementary pattern: removing the discrepancy stream, gated fusion, or graph refinement reduces performance. Together, these results show that the gain comes from the complete combination of explicit discrepancy modeling, hierarchical propagation, and gated fusion.

Additional checks. Appendix A.2 shows that deeper graph stacking does not improve performance. Appendix A.3 shows that a capacity-matched HMamba-wide baseline does not repro-

duce the main gains. Appendix A.4 compares alternative discrepancy input designs. Appendix A.6 shows that the HiGraph module also improves a reproduced GOPT baseline under a five-run evaluation.

Computational overhead. HiGraph introduces additional computation relative to HMamba. Training time increases from 4.268 to 9.889 seconds per epoch, and inference latency increases from 3.21 to 7.84 ms per utterance. Peak GPU memory remains comparable in our implementation. Full results are reported in Appendix A.8.

6 Discussion

On explicit versus implicit comparison. The localization results show that implicit models can contain useful mismatch-related information. However, this information is not preserved as a separate scoring path. HiGraph keeps canonical-realized discrepancy explicit and propagates it through the scoring hierarchy. The results suggest that this structure provides a useful path for both assessment and feedback localization.

On the assessment dimension. The gains are concentrated on word stress and utterance completeness. The error-group analysis shows that completeness improves more when deletion or substitution errors are present, and word stress improves more for words with annotated stress penalties. These results are consistent with HiGraph helping most when local mismatch provides useful evidence for higher-level scores. In contrast, fluency and prosody depend more on longer-range rhythm and speaking style.

On traceability as a design goal. HiGraph provides a defined path from phone-level mismatch evidence to word- and utterance-level scores. The resulting trace score aligns with annotated errors and performs better than the tested HMamba localization baselines. Its main value is that the trace comes from the discrepancy path used during score formation. This trace should be interpreted as structural evidence for feedback targeting, not as a full causal explanation of the underlying acoustic error.

7 Conclusion

We proposed HiGraph, which preserves canonical-realized discrepancy as an explicit scoring path for joint pronunciation assessment. HiGraph propagates phone-level mismatch evidence through a

phone-to-word-to-utterance graph and controls its effect with learned gates. Under the same pipeline, it improves word stress and utterance completeness over HMamba (rep.) and provides a trace score for feedback localization. The results show that explicit discrepancy propagation is especially useful when local mismatch provides relevant evidence for higher-level assessment.

Limitations

While HiGraph advances joint pronunciation assessment through explicit discrepancy reasoning, several limitations remain. The discrepancy representation depends on canonical phone labels derived from forced alignment. For highly non-native speech, alignment errors may introduce noise into the discrepancy stream and reduce assessment quality.

HiGraph also traces score degradation mainly through phone-level canonical-realized mismatches. This design is well matched to substitution and deletion errors, but it may not fully capture prosodic or suprasegmental errors that do not appear as individual phone-level mismatches.

The present validation uses HMamba as the main backbone and SO762 as the main benchmark. This choice is mainly driven by annotation coverage. A full evaluation of HiGraph requires phone-, word-, and utterance-level APA scores, reference transcripts, and realized-phone annotations, so that both scoring behavior and MDD-grounded trace localization can be tested under the same setting. To our knowledge, SO762 is the only public benchmark that provides this combination. Appendix A.6 adds a GOPT-based backbone check with the HiGraph module, Appendix A.5 adds a partial L2-ARCTIC trace-localization test, and Appendix A.7 adds a limited automatic speech recognition (ASR)-conditioned utterance-level evaluation on MultiPA (Chen et al., 2023). The GOPT results suggest that the proposed module is not limited to the HMamba backbone, while the external datasets provide partial evidence beyond SO762. However, these checks do not establish full cross-dataset APA generalization because the external datasets lack part of the required annotation set. MultiPA is limited to utterance-level scores and relies on ASR-generated transcripts because reference transcripts are unavailable. Although MultiPA includes word-level intelligibility labels, they are not directly comparable to the SO762 word-level accuracy, stress,

and total scores. Full validation would require another annotation-complete L2 corpus, which is not currently available among common public benchmarks.

Finally, HiGraph provides structural traceability rather than a full causal explanation. The gate activation and trace score indicate which phone-level mismatch evidence is associated with score degradation. However, they do not explain which acoustic or articulatory properties caused the mispronunciation. Connecting this structural traceability to phonetic causal explanations remains future work.

Acknowledgement

This work was supported in part by: the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) under Grant No. RS-2019-II190079 for the Artificial Intelligence Graduate School Program at Korea University; the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) under Grant No. RS-2025-02304828 for the Artificial Intelligence Star Fellowship Support Program to Nurture the Best Talents.

References

- Fu-An Chao and Berlin Chen. 2025. Towards efficient and multifaceted computer-assisted pronunciation training leveraging hierarchical selective state space model and decoupled cross-entropy loss. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1947–1961.
- Fu-An Chao, Tien-Hong Lo, Tzu-I Wu, Yao-Ting Sung, and Berlin Chen. 2022. 3m: An effective multi-view, multi-granularity, and multi-aspect modeling approach to english pronunciation assessment. In *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 575–582. IEEE.
- Fu-An Chao, Tien-Hong Lo, Tzu-I Wu, Yao-Ting Sung, and Berlin Chen. 2023. A hierarchical context-aware modeling approach for multi-aspect and multi-granular pronunciation assessment. *arXiv preprint arXiv:2305.18146*.
- Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.
- Yu-Wen Chen, Zhou Yu, and Julia Hirschberg. 2023. Multipa: A multi-task speech pronunciation assessment model for open response scenarios. *arXiv preprint arXiv:2308.12490*.
- Eunseo Choi, Hyunku Kang, and Chanwoo Kim. 2026. Siser : Speaker invariant for speech emotion recognition. In *Proc. Interspeech 2026*.
- Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2020. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*.
- Heejin Do, Yunsu Kim, and Gary Geunbae Lee. 2023a. Hierarchical pronunciation assessment with multi-aspect attention. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.
- Heejin Do, Yunsu Kim, and Gary Geunbae Lee. 2023b. Score-balanced loss for multi-aspect pronunciation assessment. *arXiv preprint arXiv:2305.16664*.
- Yassine El Kheir, Ahmed Ali, and Shammur Absar Chowdhury. 2023. Automatic pronunciation assessment-a review. *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8304–8324.
- Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. Pmlr.
- Yuan Gong, Ziyi Chen, Iek-Heng Chu, Peng Chang, and James Glass. 2022. Transformer-based multi-aspect multi-granularity non-native english speaker pronunciation assessment. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7262–7266. IEEE.
- Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- Albert Gu, Karan Goel, and Christopher Ré. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.
- Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.
- Yue-Yang He, Bi-Cheng Yan, Tien-Hong Lo, Meng-Shin Lin, Yung-Chang Hsu, and Berlin Chen. 2024. Jam: A unified neural architecture for joint multi-granularity pronunciation assessment and phone-level mispronunciation detection and diagnosis towards a comprehensive capt system. In *2024 Asia*

- Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–6. IEEE.
- Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.
- Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710.
- Hyunku Kang, Minkyu Cho, and Chanwoo Kim. 2026. Beyond short segments : Expanding speaker embeddings with vector archives. In *Proc. Interspeech*.
- Chanwoo Kim and Wonyong Sung. 2001. Vowel pronunciation accuracy checking system based on phoneme segmentation and formants extraction. In *Proceedings of International Conference on Speech Processing*, pages 447–452.
- Chanwoo Kim and Wonyong Sung. 2002. Implementation of an intonational quality assessment system. In *Proc. ICSLP 2002*, pages 1225–1228.
- Eesung Kim, Jae-Jin Jeon, Hyeji Seo, and Hoon Kim. 2022. Automatic pronunciation assessment using self-supervised speech representation learning. *arXiv preprint arXiv:2204.03863*.
- Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.
- Kun Li, Xiaojun Qian, and Helen Meng. 2016. Mispronunciation detection and diagnosis in 12 english speech using multidistribution deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):193–207.
- Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.
- Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.
- Binghuai Lin, Liyuan Wang, Xiaoli Feng, and Jinsong Zhang. 2020. Automatic scoring at multi-granularity for 12 pronunciation. In *Interspeech*, pages 3022–3026.
- Ambra Neri, Catia Cucchiari, and Helmer Strik. 2002. Feedback in computer assisted pronunciation training: When technology meets pedagogy.
- Leonardo Neumeyer, Horacio Franco, Mitchel Weintraub, and Patti Price. 1996. Automatic text-independent pronunciation scoring of foreign language student speech. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP’96*, volume 3, pages 1457–1460. IEEE.
- Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.
- Hyungshin Ryu, Sunhee Kim, and Minhwa Chung. 2023. A joint model for pronunciation assessment and mispronunciation detection and diagnosis with multi-task learning. In *Interspeech*, pages 959–963.
- Jiatong Shi, Nan Huo, and Qin Jin. 2020. Context-aware goodness of pronunciation for computer-assisted pronunciation training. *arXiv preprint arXiv:2008.08647*.
- Silke M Witt and Steve J Young. 2000. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech communication*, 30(2-3):95–108.
- Bi-Cheng Yan and Berlin Chen. 2024. An effective hierarchical graph attention network modeling approach for pronunciation assessment. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3974–3985.
- Bi-Cheng Yan, Ming-Kang Tsai, and Berlin Chen. 2025. Muffin: Multifaceted pronunciation feedback model with interactive hierarchical neural modeling. *IEEE Transactions on Audio, Speech and Language Processing*.
- Junbo Zhang, Zhiwen Zhang, Yongqing Wang, Zhiyong Yan, Qiong Song, Yukai Huang, Ke Li, Daniel Povey, and Yujun Wang. 2021. speechocean762: An open-source non-native english speech corpus for pronunciation assessment. *arXiv preprint arXiv:2104.01378*.
- Guanlong Zhao, Evgeny Chukharev-Hudilainen, Sinem Sonsaat, Alif Silpachai, Ivana Lucic, Ricardo Gutierrez-Osuna, and John Levis. 2018. L2-arctic: A non-native english speech corpus.

A Appendix

A.1 Supplementary Error Type Analysis

Section 5.2 shows that the trace score C_j increases with the number of phone errors in a word. We further test whether the same pattern appears across phone error types. Each word is grouped as correct, substitution, deletion, or mixed error based on its annotated phone level errors. We then compare the mean word gate activation and the trace score C_j .

Figure 4 shows that correct words have much lower gate activation and lower trace scores than words with phone-level errors. The median word gate is 0.574 for correct words, compared with 0.740 for substitution words, 0.739 for deletion words, and 0.751 for mixed error words. The trace score shows the same pattern. Correct words have a median score of 0.018, while substitution, deletion, and mixed error words have median scores of 3.653, 3.434, and 4.388.

This analysis provides supplementary evidence for the main traceability results. HiGraph assigns higher trace scores to words that contain annotated phone-level errors across different error types. This is consistent with its discrepancy aware design.

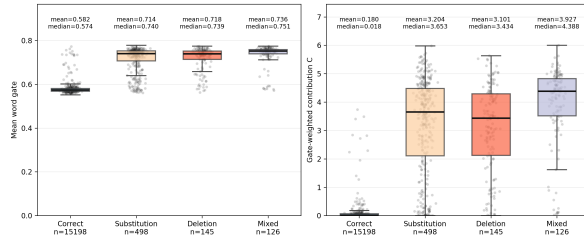


Figure 4: Supplementary error type analysis. The left panel shows the mean word gate activation by word error type. The right panel shows the trace score C_j by word error type. Words with substitution, deletion, and mixed phone errors show higher gate activation and higher trace scores than correct words.

A.2 GNN Depth Sensitivity

We further analyze the effect of GNN depth in HiGraph. The default configuration uses one layer for each graph stage. We define the three stages as phone-level refinement, phone-to-word propagation, and top-level propagation. The top-level stage includes word-to-word and word-to-utterance propagation. We group word-to-word and word-to-utterance propagation as one top-level stage because both operate after word-level discrepancy

nodes are formed and jointly control how word-level evidence reaches the utterance representation.

Table 7: Sensitivity to uniformly increasing GNN depth across all graph stages. Phone denotes phone-level graph refinement, P→W denotes phone-to-word propagation, and Top-level denotes word-to-word and word-to-utterance propagation.

Phone	P→W	Top-level	Phn	W-Str	U-Comp	U-Tot
1	1	1	0.741	0.371	0.339	0.831
2	2	2	0.734	0.289	0.011	0.825
3	3	3	0.732	0.272	0.046	0.824

Table 8: Sensitivity to increasing one graph stage at a time from the default configuration. The default setting uses one layer for each stage.

Phone	P→W	Top-level	Phn	W-Str	U-Comp	U-Tot
1	1	1	0.741	0.371	0.339	0.831
2	1	1	0.725	0.266	0.104	0.817
1	2	1	0.735	0.342	0.043	0.824
1	1	2	0.732	0.335	0.163	0.823

Table 7 shows that uniformly increasing GNN depth does not improve HiGraph. The default shallow configuration gives the best overall performance. Increasing all stages to two or three layers degrades phone-level PCC and word stress, and it causes a large drop in utterance completeness. This suggests that deeper message passing may over-smooth the discrepancy signal and hurt utterance-level assessment (Li et al., 2018).

Table 8 shows that increasing any single stage beyond one layer also fails to improve over the default setting. The drop is especially clear for utterance completeness. These results suggest that HiGraph benefits from local hierarchical refinement, while deeper graph stacking is unnecessary and less stable.

A.3 Capacity-Controlled Comparison

We further compare HiGraph with a capacity-increased HMamba variant. HMamba-wide increases the hidden dimension of HMamba while keeping the same input features, prediction heads, and training recipe. This comparison tests whether the gains of HiGraph mainly come from increased model capacity or from explicit discrepancy propagation.

Table 9: Capacity-controlled comparison on SO762. HMamba-wide increases the hidden dimension and has a comparable parameter count to HiGraph. It does not reproduce the gains of HiGraph, especially on word stress and utterance completeness.

Model	Params	Phn	W-Str	W-Tot	U-Comp	U-Tot
HMamba (rep.)	2.49M	0.735	0.327	0.718	0.248	0.825
HMamba-wide	5.28M	0.729	0.337	0.708	0.248	0.824
HiGraph	5.26M	0.741	0.371	0.720	0.339	0.830

Table 9 shows that increasing the width of HMamba does not reproduce the main improvements of HiGraph. HMamba-wide has a comparable parameter count to HiGraph, but remains below HiGraph on all reported metrics. It slightly improves word stress over HMamba (rep.), but it does not improve utterance completeness. This suggests that the gains of HiGraph are tied to explicit discrepancy propagation rather than to increased model capacity alone.

A.4 Discrepancy Input Design

We further compare different ways to form the phone-level discrepancy input.

Table 10: Comparison of discrepancy input designs on SO762. $\tilde{h}$ and $\tilde{e}$ denote refined realized and canonical phone embeddings.

Method	Phn	W-Str	W-Tot	U-Comp	U-Tot
$[\tilde{h}, \tilde{e}, \tilde{h} - \tilde{e}, \tilde{h} \odot \tilde{e}]$	0.741	0.371	0.720	0.339	0.830
$[\tilde{h}, \tilde{e}]$ only	0.738	0.356	0.708	0.189	0.827
trainable edge only	0.731	0.286	0.701	0.073	0.819
trainable edge + BCE	0.728	0.288	0.713	0.089	0.829
trainable edge + bias	0.738	0.304	0.716	0.131	0.816

Table 10 shows that using explicit difference and product features gives stronger results than using the two embeddings alone. The trainable-edge variants are also weaker. This suggests that direct mismatch features are useful for forming the discrepancy representation.

A.5 Cross-Dataset Trace Localization on L2-ARCTIC

SO762 is used as the primary benchmark because it provides phone-, word-, and utterance-level assessment labels together with realized-phone annotations. These annotations allow us to evaluate both assessment performance and MDD-grounded feedback localization in a single controlled setting.

To further test whether the learned trace score is specific to SO762, we conduct an additional cross-dataset trace-localization analysis on L2-ARCTIC.

L2-ARCTIC does not provide the same multi-level APA score annotations as SO762. Therefore, we do not use it to evaluate scoring performance. Instead, we use it only to test whether the trace score learned from SO762 can rank words that contain manually annotated pronunciation errors. We mark each word as mispronounced if it contains at least one annotated phone-level error, and evaluate the same scores used in Section 5.2: the trace score $C = \bar{g} \cdot \ell$ and its two components, the word-level gate activation $\bar{g}$ and the predicted score deficit ℓ .

We use the manually annotated subset of L2-ARCTIC, which contains 3,599 utterances. During preprocessing, we find that 32 utterances have inconsistent phone orders between the Kaldi GOP output and the ali-phone segmentation. This inconsistency breaks the strong one-to-one phone alignment assumption used in our SO762 pipeline. In these cases, the GOP phone sequence cannot be reliably matched to the ali-phone segmentation, making the phone-to-word mapping ambiguous. We therefore exclude these utterances to avoid assigning annotated phone errors to incorrect word positions. Among the remaining utterances, we use 3,552 utterances that contain at least one annotated error word for the ranking evaluation.

Table 11: Cross-dataset trace localization on L2-ARCTIC. Because L2-ARCTIC lacks SO762-style multi-level APA labels, we evaluate only whether $C = \bar{g} \cdot \ell$ ranks words with manually annotated pronunciation errors. AUROC/AUPRC measure ranking quality, and Top-1/Top-2 measure whether the top-ranked words contain annotated mispronunciations.

Ranking score	AUROC	AUPRC	Top-1	Top-2
Random	0.500	0.661	0.662	0.880
Gate activation $\bar{g}$	0.618	0.761	0.790	0.949
Score deficit ℓ	0.590	0.762	0.835	0.946
Trace score $C = \bar{g} \cdot \ell$	0.611	0.764	0.838	0.948

Table 11 shows the cross-dataset trace localization results. The random baseline is high on Top-1 and Top-2 because this evaluation is restricted to utterances that contain at least one annotated error word, and many L2-ARCTIC utterances contain multiple error words. Even under this strong random baseline, all non-random ranking scores improve over random ranking. The word-level

Table 12: Additional GOPT-based backbone check on SO762. GOPT denotes the result reported in the original paper, while GOPT (rep.) denotes our reproduced baseline. GOPT (rep.) and GOPT (rep.) + HiGraph Module report mean and standard deviation over five runs.

Model	Phoneme-level		Word-level aspects			Utterance-level aspects				
	MSE ↓	PCC ↑	Acc ↑	Stress ↑	Total ↑	Acc ↑	Comp ↑	Flu ↑	Prosody ↑	Total ↑
GOPT (Gong et al., 2022)	0.085	0.612	0.533	0.291	0.549	0.714	0.155	0.753	0.760	0.742
GOPT (rep.) (Gong et al., 2022)	0.085 (0.001)	0.611 (0.003)	0.534 (0.003)	0.295 (0.035)	0.554 (0.002)	0.712 (0.005)	0.162 (0.032)	0.754 (0.007)	0.756 (0.006)	0.744 (0.005)
GOPT (rep.) + HiGraph Module	0.081 (0.003)	0.638 (0.003)	0.589 (0.005)	0.318 (0.041)	0.606 (0.005)	0.731 (0.004)	0.227 (0.056)	0.761 (0.007)	0.763 (0.008)	0.748 (0.005)

gate $\bar{g}$ gives the highest AUROC and Top-2 score, while the predicted score deficit ℓ gives a strong Top-1 score. Their product $C = \bar{g} \cdot \ell$ gives the best AUPRC and Top-1 score. These results suggest that the discrepancy-based trace score remains useful on a separate L2 English corpus, although full cross-dataset validation of both APA scoring and feedback localization would require another annotation-complete corpus.

A.6 Additional GOPT-Based Backbone Check

To examine whether the proposed graph reasoning module is limited to the HMamba backbone, we conduct an additional backbone check with GOPT. This experiment is not intended to reproduce the full HiGraph system with the HMamba backbone. Instead, we attach the HiGraph module to the reproduced GOPT baseline while keeping the GOPT input features, prediction heads, and training recipe unchanged. Because GOPT also uses canonical phone information as input, the HiGraph module can compute canonical-realized mismatch and propagate it through the phone-to-word-to-utterance structure. This setting tests whether the proposed discrepancy-aware graph reasoning is useful beyond the HMamba backbone.

Table 12 shows the five-run results. Adding the HiGraph module improves the mean over GOPT (rep.) across all APA metrics. The gains are especially clear on phone-level PCC, word accuracy, word total, and utterance completeness. This result suggests that the proposed discrepancy-aware graph reasoning is not limited to the HMamba backbone, although broader validation across more backbones remains future work.

A.7 Limited Cross-Dataset Utterance-Level Evaluation

We further conduct a limited cross-dataset utterance-level evaluation on MultiPA, an external open-response pronunciation assessment dataset. MultiPA contains short free-speech responses collected from learners interacting with a dialogue-based chatbot. Unlike SO762, it does not provide reference transcripts or phone-level assessment labels. Although MultiPA includes word-level intelligibility labels, they are not directly comparable to the SO762 word-level accuracy, stress, and total scores used in our main evaluation. In addition, word-level evaluation would require aligning ASR-derived word sequences with time-span-based human labels, which can introduce additional noise. We therefore use MultiPA only as a limited external check of utterance-level scoring behavior, rather than as a substitute for the annotation-complete SO762 evaluation.

Since reference transcripts are unavailable, we evaluate under an automatic speech recognition (ASR)-conditioned open-response setting. We first generate transcripts using Whisper-large-v3 (Radford et al., 2023) and normalize the recognized text. We then derive canonical pronunciations from the normalized ASR transcripts using a lexicon and the CMU Pronouncing Dictionary. This setting is noisier than SO762 because ASR errors can affect the canonical side used for discrepancy computation. If an utterance contains a word that is not covered by the pronunciation dictionary, we exclude that utterance because its canonical phone sequence cannot be reliably constructed. As a result, 44 of the 50 utterances are used for evaluation, and 6 utterances are excluded. This filtering avoids assigning unreliable canonical pronunciations to out-of-vocabulary words.

The output scale also differs between our SO762-

trained models and MultiPA labels. Our model outputs are internally in a 0–2 range, while MultiPA human scores are given on a 1–5 scale. To match the target scale, we convert each prediction by

$$\hat{y}_{\text{MultiPA}} = \text{clip}(2.5 \cdot \hat{y}_{\text{model}}, 1, 5), \quad (18)$$

where $\hat{y}_{\text{model}}$ is the raw model output and $\text{clip}(\cdot, 1, 5)$ restricts the prediction to the valid MultiPA score range. We then report PCC between the converted predictions and the human utterance-level scores.

Table 13: Limited ASR-conditioned utterance-level evaluation on MultiPA. Models are trained on SO762 and evaluated using Whisper-large-v3 transcripts because reference transcripts are unavailable. Canonical pronunciations are derived from normalized ASR transcripts, and utterances with out-of-dictionary words are excluded, leaving 44 of 50 utterances. We report utterance-level PCC as a limited transfer check.

Model	U-Acc	U-Flu	U-Pros
HMamba (rep.) (Chao and Chen, 2025)	0.542	0.604	0.518
HiGraph	0.536	0.637	0.531

Table 13 shows the results. HiGraph gives a similar utterance accuracy score to HMamba, with a small decrease of 0.006. At the same time, HiGraph improves utterance fluency by 0.033 and utterance prosody by 0.013. These results suggest aspect-dependent behavior under ASR-derived transcripts. However, the small sample size and the use of ASR-generated transcripts make this setting noisy. We therefore interpret this experiment as a limited external check rather than as evidence of full cross-dataset APA generalization.

A.8 Computational Efficiency

We compare the computational cost of HiGraph and HMamba under the same implementation and batch setting on an NVIDIA RTX 5090 GPU.

Table 14: Computational comparison between HMamba and HiGraph.

Model	Params	Train/epoch	Memory	Latency
HMamba	2.49M	4.268 s	648.8 MB	3.21 ms/utt
HiGraph	5.26M	9.889 s	599.9 MB	7.84 ms/utt

HiGraph increases parameter count, training time, and inference latency relative to HMamba.

Peak GPU memory is comparable under this setting. Although inference is slower, the absolute latency remains 7.84 ms per utterance.