Optimality of FSQ Tokens for Continuous Diffusion for Categorical Data with Application to Text-to-Speech

Vadim Popov[1,2], Wenju Gu[1], Tasnima Sadekova[1,2], Georgii Aparin[1,3], Assel Yermekova[1]

[1]Huawei Noah's Ark Lab
[2]National Research University Higher School of Economics
[3]National University of Science and Technology MISIS

Forty-Third International Conference on Machine Learning Seoul, South Korea, 2026

Continuous Diffusion for Categorical Data (CDCD)

Common continuous-domain diffusion models:

- Forward diffusion adding noise to data operates in *continuous* space $\mathbb{R}^n$.
- Neural network guiding reverse diffusion also makes predictions (usually $\nabla log p_t(X_t)$, $\mathbb{E}[X_0|X_t]$, $\mathbb{E}[\varepsilon_t|X_t]$, or their linear combination) in *continuous* space $\mathbb{R}^n$ and is trained with *MSE loss.*

Common discrete-domain diffusion models:

- Forward diffusion operates in discrete space: a sequence of *discrete* tokens is perturbed (e.g. masked).
- Neural network guiding reverse diffusion predicts logits for a vocabulary with *finite* size and is trained with *CE loss.*

Continuous Diffusion for Categorical Data (CDCD)

Common continuous-domain diffusion models:

- Forward diffusion adding noise to data operates in *continuous* space $\mathbb{R}^n$.
- Neural network guiding reverse diffusion also makes predictions (usually $\nabla log p_t(X_t)$, $\mathbb{E}[X_0|X_t]$, $\mathbb{E}[\varepsilon_t|X_t]$, or their linear combination) in *continuous* space $\mathbb{R}^n$ and is trained with *MSE loss.*

Common discrete-domain diffusion models:

- Forward diffusion operates in discrete space: a sequence of *discrete* tokens is perturbed (e.g. masked).
- Neural network guiding reverse diffusion predicts logits for a vocabulary with *finite* size and is trained with *CE loss.*

Continuous Diffusion for Categorical Data (CDCD):

- Both forward and reverse diffusions operate in *continuous* space of *token embeddings* while neural network guiding reverse diffusion predicts *probabilities of tokens* from a finite vocabulary with *cross-entropy loss*:

$$\mathcal{L}_{diff}(\theta) = \mathbb{E}_{(k,X_t)\sim p_{0,t}(\cdot,\cdot|c)}[-\log P_\theta(k|X_t,t,c)]$$ where *c* is conditioning, *k* is token id and X_t is its noisy embedding

- Reverse diffusion is still a stochastic process in continuous space, and the score function can be written down in terms of the probabilities P_θ

Continuous Diffusion for Categorical Data (CDCD)

Common continuous-domain diffusion models:

- Forward diffusion adding noise to data operates in *continuous* space $\mathbb{R}^n$.
- Neural network guiding reverse diffusion also makes predictions (usually $\nabla log p_t(X_t)$, $\mathbb{E}[X_0|X_t]$, $\mathbb{E}[\varepsilon_t|X_t]$, or their linear combination) in *continuous* space $\mathbb{R}^n$ and is trained with *MSE loss.*

Common discrete-domain diffusion models:

- Forward diffusion operates in discrete space: a sequence of *discrete* tokens is perturbed (e.g. masked).
- Neural network guiding reverse diffusion predicts logits for a vocabulary with *finite* size and is trained with *CE loss.*

Continuous Diffusion for Categorical Data (CDCD):

- Both forward and reverse diffusions operate in *continuous* space of *token embeddings* while neural network guiding reverse diffusion predicts *probabilities of tokens* from a finite vocabulary with *cross-entropy loss*:

$$\mathcal{L}_{diff}(\theta) = \mathbb{E}_{(k,X_t)\sim p_{0,t}(\cdot,\cdot|c)}[-\log P_\theta(k|X_t,t,c)]$$ where *c* is conditioning, *k* is token id and X_t is its noisy embedding

- Reverse diffusion is still a stochastic process in continuous space, and the score function can be written down in terms of the probabilities P_θ

What is the optimal relative location of token embeddings for the best CDCD performance?

Optimality of FSQ latent space

FSQ is a VQ method leading to tokens located “uniformly” in some hypercube in $\mathbb{R}^n$ for relatively low dimensionality n.

Optimality of FSQ latent space

FSQ is a VQ method leading to tokens located “uniformly” in some hypercube in $\mathbb{R}^n$ for relatively low dimensionality *n*.

Optimality in terms of token prediction accuracy

Optimally trained NN predictions :

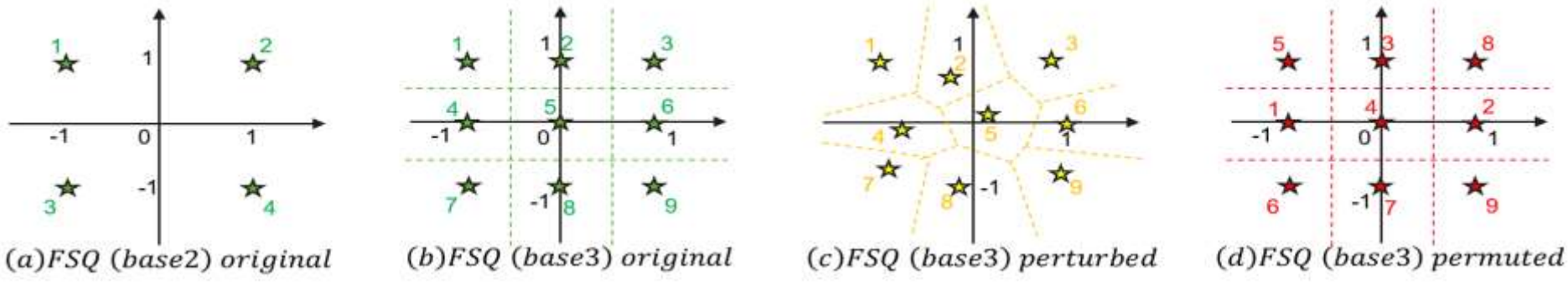
$$P_{0|t}(k|x) \propto p_k \exp\left(-\frac{1}{2\sigma_t^2}\|x - \alpha_t e_k\|_2^2\right)$$

Areas where predicted token id is k:

$$\Omega_k^X(t) = \{x \in \mathbb{R}^n : k = \arg\max_j P_{0|t}(j|x)\}$$

Average prediction accuracy:

$$A(E,t) = \sum_{k=1}^{V} P(X_0 = e_k, X_t \in \Omega_k^X(t))$$



Optimality of FSQ latent space

FSQ is a VQ method leading to tokens located "uniformly" in some hypercube in $\mathbb{R}^n$ for relatively low dimensionality *n*.

Optimality in terms of token prediction accuracy

Optimally trained NN predictions :

$$P_{0|t}(k|x) \propto p_k \exp\left(-\frac{1}{2\sigma_t^2}\|x - \alpha_t e_k\|_2^2\right)$$

Areas where predicted token id is k:

$$\Omega_k^X(t) = \{x \in \mathbb{R}^n : k = \arg\max_j P_{0|t}(j|x)\}$$

Average prediction accuracy:

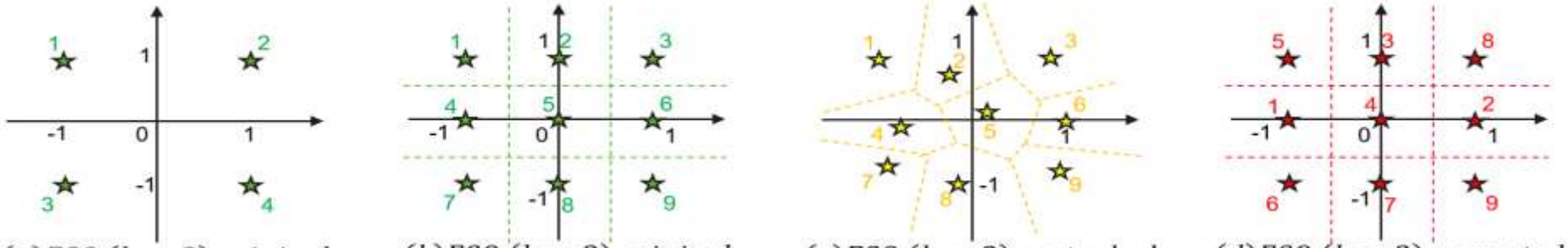
$$A(E,t) = \sum_{k=1}^{V} P(X_0 = e_k, X_t \in \Omega_k^X(t))$$

Optimality in terms of average nearest neighbour distance

ANN distance:

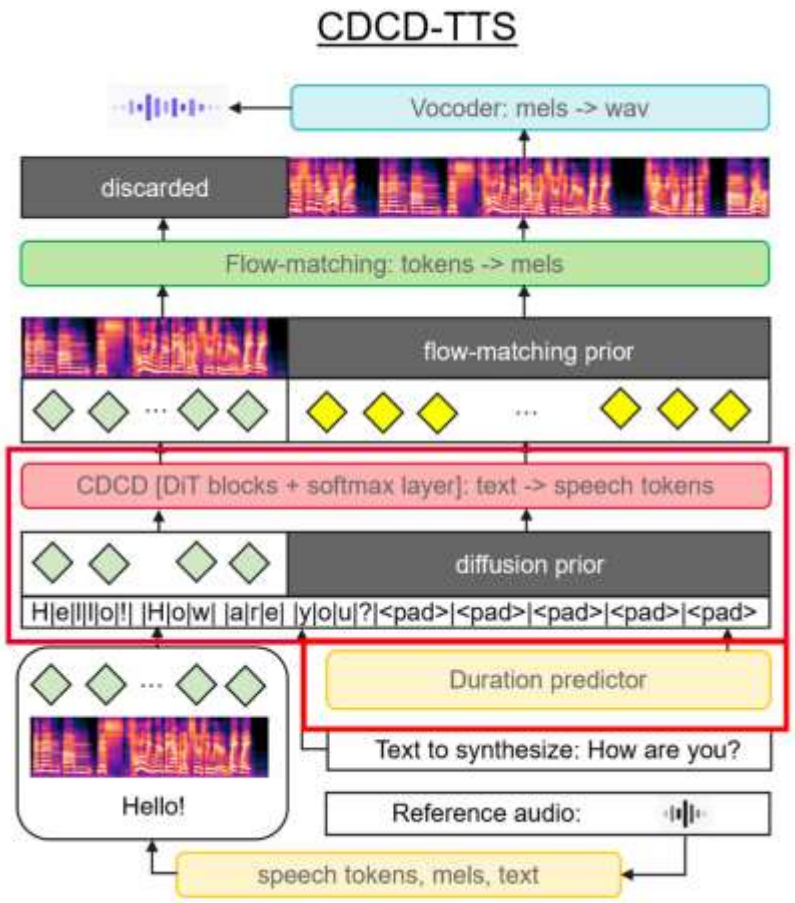
$$D(E) := \frac{1}{V}\sum_{k=1}^{V} \min_{i \neq k} \|e_i - e_k\|_2^2.$$

Theorem 4.3. *In terms of average nearest neighbour distance D, the codebooks $E_{FSQ} = \{e_k\}_{k=1}^V$ of base2 and base3 FSQ methods are locally optimal in the class of all codebooks with V entries such that $\|e_k\|_\infty \leq 1$ for all $k = 1, .., V$, i.e. any sufficiently small perturbation of a vector from the codebook E_{FSQ} leads to decreasing D.*



CDCD-TTS

CDCD-TTS – a text-to-speech model based on FSQ speech tokens.



CDCD-TTS

CDCD-TTS – a text-to-speech model based on FSQ speech tokens.

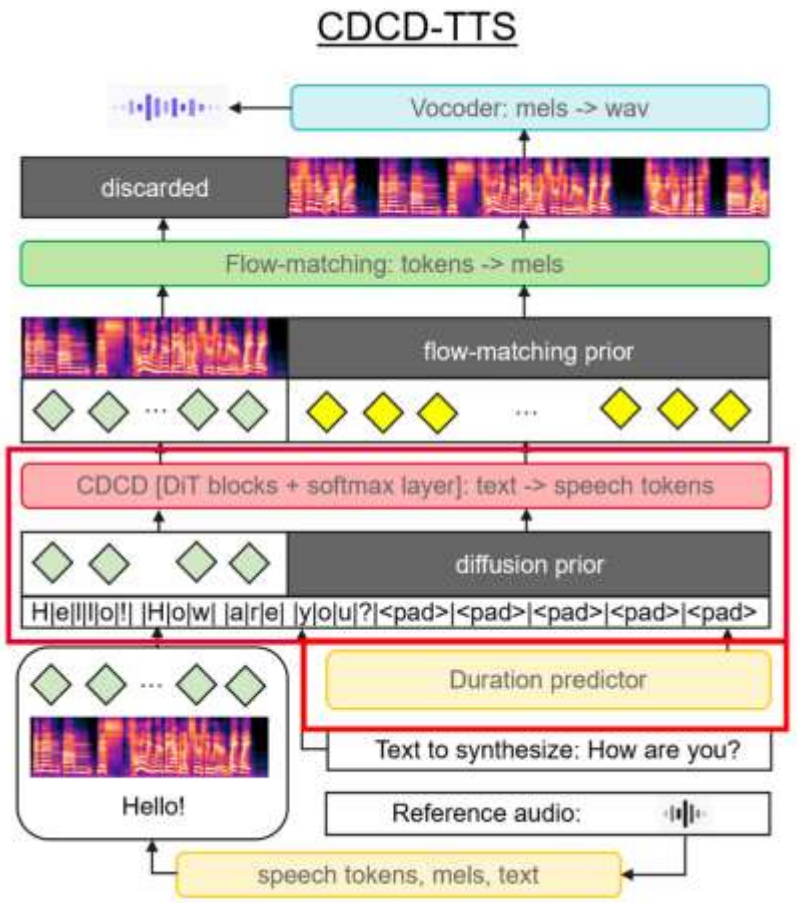


Table 1. Evaluation of TTS models on SEED *test-en* set. F5-TTS and CosyVoice3 results in italics are given as a reference.

	WER	SIM	MOS	EMO
RVQ-25	21.3%	0.382	2.932	52.0%
FSQ-permute-25	15.4%	0.588	3.631	70.1%
FSQ-original-5	2.39%	**0.654**	4.093	71.7%
FSQ-perturb-5	3.10%	0.647	3.834	70.6%
FSQ-original-8	2.10%	**0.654**	**4.119**	72.2%
FSQ-perturb-8	2.32%	0.649	4.030	71.8%
FSQ-original-12	2.05%	**0.653**	**4.120**	72.3%
FSQ-perturb-12	2.14%	0.647	4.088	72.1%
FSQ-original-25	**2.00%**	**0.653**	**4.119**	**72.7%**
FSQ-perturb-25	2.03%	0.648	**4.118**	72.3%
CosyVoice2(2024)	2.57%	**0.652**	4.077	72.2%
F5-TTS(2024)	*1.83%*	*0.665*	*3.754*	*71.4%*
CosyVoice3(2025)	*1.68%*	*0.695*	*3.937*	*72.7%*

CDCD-TTS performs comparable or better than LLM-based counterpart CosyVoice2 while being significantly smaller and faster.

Thank you!