# Exploring Conditional Variational Mechanism to Pinyin Input Method for Addressing One-to-Many Mappings in Low-Resource Scenarios

Bin Sun<sup>1\*</sup>, Jianfeng Li<sup>2</sup>, Hao Zhou<sup>2</sup>, Fandong Meng<sup>2</sup>, Kan Li<sup>1†</sup>, Jie Zhou<sup>2</sup>

<sup>1</sup>School of Computer Science & Technology, Beijing Institute of Technology

<sup>2</sup>WeChat AI, Tencent Inc., China

{binsun,likan}@bit.edu.cn

{lijfli,tuxzhou,fandongmeng,withtomzhou}@tencent.com

#### Abstract

Pinyin input method engine (IME) refers to the transformation tool from pinyin sequence to Chinese characters, which is widely used on mobile phone applications. Due to the homophones, Pinyin IME suffers from the oneto-many mapping problem in the process of pinyin sequences to Chinese characters. To solve the above issue, this paper makes the first exploration to leverage an effective conditional variational mechanism (CVM) for pinyin IME. However, to ensure the stable and smooth operation of Pinyin IME under low-resource conditions (e.g., on offline mobile devices), we should balance diversity, accuracy, and efficiency with CVM, which is still challenging. To this end, we employ a novel strategy that simplifies the complexity of semantic encoding by facilitating the interaction between pinyin and the Chinese character information during the construction of continuous latent variables. Concurrently, the accuracy of the outcomes is enhanced by capitalizing on the discrete latent variables. Experimental results demonstrate the superior performance of our method.

# 1 Introduction

Input method engines  $(IMEs)^1$  are important tools to connect users with mobile applications, drawing dramatic attentions (Chen and Lee, 2000; Li et al., 2004; Zheng et al., 2011; Han and Chang, 2013; Chen et al., 2013; Jia and Zhao, 2014; Huang et al., 2015, 2018; Zhang et al., 2019; Liu et al., 2021; Tan et al., 2022; Ding et al., 2023). In China, there are two common Pinyin IMEs<sup>2</sup> for cellphones: the 9-key IMEs and the 26-key IMEs, which are used by more than **97%** of Chinese people (Hu et al., 2022). As shown in Figure 1, the 26-key keyboard

<sup>1</sup> @/.		DEF	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$
GHI	JKL	MNO	Ã Ś Ď F Ġ Ĥ J K L
PQRS	TUV	WXYZ	★ Ž X Č V B Ň M
(a) 9	-key IN	1E	(b) 26-key IME

Figure 1: The 9-key and 26-key IME.

uses the 26 English letters as Chinese pinyin syllables, while the 9-key keyboard maps the 26 pinyin syllables onto 8 keys.

Due to the Chinese homophones, the process of converting pinyin sequences to Chinese character sequences inevitably presents a one-to-many mapping challenge for Pinyin IME. In the perfect pinyin mode of a 26-key IME, 500 pinyin combinations need to correspond to nearly 10,000 Chinese characters (Jia and Zhao, 2014; Zhang et al., 2019). For instance, inputting the pinyin sequence "bei zi" can map to various Chinese characters with completely different meanings, such as "被子" (blanket), "杯 子" (cup), and "辈子" (lifetime). In the case of the abbreviated pinyin mode, entering the initial letters "b z" for "bei zi" can result in not only the aforementioned characters but also others like "不 止" (more than), "不在" (not present), and "步骤" (steps). As for 9-key IMEs, each key can represent 3 to 4 pinyin syllables, which means that inputting "23494" offers 323 possible pinyin combinations except "beizi". While some pinyin combinations that do not adhere to standard rules can be pruned, this undoubtedly expands the solution space.

One effective method to alleviate the one-tomany problem is to generate more candidates for users to autonomously choose the one they need. Existing methods typically employ beam search to generate additional candidates. However, Holtzman et al. (2020) found that unlike beam search, which selects the token with the highest probability, humans tend to choose more surprising and

<sup>\*</sup>Work done at WeChat AI, Tencent Inc.

<sup>&</sup>lt;sup>†</sup>Corresponding Author

<sup>&</sup>lt;sup>1</sup>https://en.wikipedia.org/wiki/Input\_method

<sup>&</sup>lt;sup>2</sup>https://en.wikipedia.org/wiki/Pinyin\_input\_method

diverse tokens. Furthermore, beam search requires sorting multiple candidates during the generation process, and in some low-resource scenarios (such as on offline mobile devices), it is challenging to ensure stable and rapid generation due to the lack of sufficient memory and computational resources.

To alleviate the aforementioned problems, we take inspirations from conditional variational mechanism, which models the one-to-many cases through the latent variable space and generate various results by sampling different latent variables (Shen et al., 2017; Zhao et al., 2017; Bao et al., 2020; Lin et al., 2020; Fang et al., 2021; Sun et al., 2021). Therefore, instead of prioritizing the arrangement of the highest-scoring candidate result, our primary study of interest is to recall more eligible candidates within the same inference time. To this end, we propose a conditional variational IME model (CV-IME) with a novel hybrid latent variables strategy. Please refer to § 2.2 for details.

Our contributions are as follow: To the best of our knowledge, this is the first exploration and investigation of the impact of CVM on the performance of Pinyin IME in low-resource scenarios, specifically on offline mobile platforms. Furthermore, we propose a novel hybrid latent variable that designed to balance the performance and efficiency of our CV-IME model.

## 2 Methodology

## 2.1 Base Model

With the advancement of technology, the latest Pinyin IMEs, e.g., PinyinGPT (Tan et al., 2022) and GeneInput (Ding et al., 2023), primarily adopt models based on the transformer architecture (Vaswani et al., 2017). Therefore, we have adopted the transformer structure and conducted a series of experiments to identify the most suitable configuration.

#### 2.2 Conditional Variational IME

Following the previous work of CVM, CV-IME primarily consists of four components: a encoderdecoder model, a prior network  $p_{\theta}(z|c)$ , a recognition network  $q_{\phi}(z|r,c)$  and a discrete latent variable matrix M. c, r and z represent the user input (i.e., context and pinyin sequence), the character result and the continues latent variable.

**Hybrid Latent Variable.** Previous researches indicate that continuous latent variables can enhance diversity but may reduce relevance, whereas discrete latent variables strengthen relevance but lack diversity (Gao et al., 2019; Bao et al., 2020; Sun et al., 2021, 2023). Therefore, a promising direction is to hybrid the continuous and discrete latent variables, leveraging their respective strengths to complement and offset their weaknesses. To build the hybrid latent variables H, we follow Sun et al. (2023), adding sentence-level continuous latent variable  $z'_s$  to the discrete latent variables M:  $H = (z'_s + M[1], \dots, z'_s + M[k])$ , where K represents the number of discrete latent variables.

**Continuous Latent Variables.** We initially employ the model encoder to transform c and c+r into prior memory **h** and posterior memory **h**'. Given that there is a degree of alignment between the pinyin and character sequences in the task of pinyinto-character conversion, relying solely on the encoder's self-attention mechanism for interaction may not yield effective information. Therefore, we have introduced an interaction between the prior memory and the posterior memory:

$$\mathbf{h}' = \text{SoftMax}(\mathbf{h} \cdot \mathbf{h}'^T) \cdot \mathbf{h}'$$
(1)

To enhance the recognition process, we use **h** and **h**' together to estimate the isotropic Gaussian distribution  $q_{\phi}(z|c,r) \sim \mathcal{N}(\mu', \sigma'^2 \mathbf{I})$ :

$$\begin{pmatrix} \mu'_1, ..., \mu'_n \\ \log(\sigma'^2_1), ..., \log(\sigma'^2_n) \end{pmatrix} = \begin{pmatrix} [h_1; h'_1] \\ \cdots \\ [h_n; h'_n] \end{pmatrix} W'_u,$$

where  $W'_u$  is trainable parameters of  $q_{\phi}(z|r, c)$ . After that, we follow additive Gaussian mixing (Wang et al., 2017) to obtain the sentence-level continuous latent variables. (see more details in Appendix B)

## **3** Experimental Settings

**Benchmarks.** We used two public benchmarks, namely the People's Daily (PD) corpus (Yang et al., 2012) and WD dataset (Tan et al., 2022), in the experiments. PD is extracted from the People's Daily from 1992 to 1998, while WD is extracted from the WuDaoCorpora (Yuan et al., 2021). Different from PD, WD contains test cases from 16 different domains of test cases.

**Evaluation Metrics.** We use the precision of top-N, indicating whether the desired result is included in the generated top-N results. We also use the inference time for one instance as a metric. **Training Dataset.** To train our CV-IME and base model, we built a training dataset for the pinyin-to-character task based on the news2016 corpus<sup>3</sup>. We randomly extract sentence from news2016 corpus and incorporate "pypinyin" tool to convert Chinese Characters into Pinyin syllables. Table 1 shows the statistics of this dataset. (Please refer to appendix C for more details)

Number of	Samples	Average Sequ	uence Length
# Perfect	9692887	Context	18.28
# Abbreviated	9637283	Pinyin	14.68
# Total	19330170	Character	7.31

Table 1: Key statistics of our training dataset.

**Baseline Models.** We introduced some IMEs, i.e., Google IME<sup>4</sup>, On-OMWA (Zhang et al., 2017) and On-P2C (Zhang et al., 2019) as baselines.

- GoogleIME is a commercial Chinese IME that offers an API with debugging capabilities.
- On-OMWA system, introduced by Zhang et al. (2017), is an adaptive online model designed for the acquisition of new words, specifically tailored for Chinese IMEs.
- On-P2C model, as described in Zhang et al. (2019) on open vocabulary learning, is a neural network-based Pinyin-to-Chinese conversion system that improves its performance by dynamically updating its word database to facilitate learning of an open vocabulary.

**Training Detail.** The hidden size of all models is set to 512. Our CV-IME employs a Transformer model with 2 encoder layers and 1 decoder layer, and additionally incorporates two fully-connected layers as a prior network. We set the batch sizes to 1024 and 256 for base model and CV-IME, respectively. Adam is used for optimization. The initial learning rate is set to 0.0001. We also introduce KL annealing trick to leverage the KL divergence during the training. The KL weight increases linearly from 0 to 1 in the first 3000000 batches. We train all models in 100 epochs on four A100 GPU cards with Pytorch, and save the model parameters when the validation loss reaching minimum.

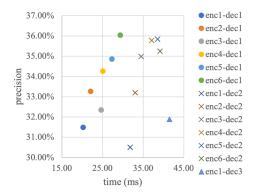


Figure 2: Results of different encoder-decoder layer configurations over PD using 9-key IME.

Model	# Enc	# Dec	# Parameters
	1	1	21.89M
	2	1	24.89M
Base Model	3	1	27.90M
	4	1	30.91M
	5	1	33.91M
CV-IME	2	1	28.90M

Table 2: The number of parameters contained in different configurations of base model and CV-IME.

#### 4 Result and Analysis

#### 4.1 Model Structure Selection.

Figure 2 and Table 2 show the generation latency, accuracy and parameters of base models with different configurations, which illustrates that: (1) maintaining a fixed number of encoder while solely increasing decoder layers significantly raises latency ( $\approx$ 10ms) without notably improving accuracy. (2) while the increase in the number of encoder layers leads to a gradual rise in latency ( $\approx$ 2ms), accompanied by an upward trend in accuracy. (3) the number of parameters in an encoder is to some extent positively correlated with accuracy.

In selecting the final encoder-decoder configuration, we primarily considered constraints on memory storage and latency. In this work, we posit that under low-resource constraints, with a storage ceiling of no more than 32MB and a generation latency not exceeding 30ms for a single candidate, the system can operate reliably.

Regarding storage, we aimed to emulate a realistic mobile environment, mindful of the fact that an IME system houses multiple models, such as PinyinIME, speech recognition, handwriting recognition, etc. Given the overall storage consumption of the system must remain low, we endeavored to limit the size of the PinyinIME model to within

<sup>&</sup>lt;sup>3</sup>https://github.com/brightmart/nlp\_chinese\_corpus <sup>4</sup>https://www.google.com/inputtools/services/features/input-

method.html

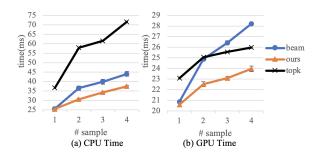


Figure 3: Comparative results of different methods to generate delay.

32MB. To ensure stable performance, the model was restricted to using at most INT8 quantization, which means the parameters had to be kept under 32M. As for latency, we set a strict benchmark: the CPU latency for generating a single case under lengthy text conditions must not exceed 30ms. Since the model needs to regenerate results immediately with each pinyin character input by the user, to prevent perceptible delays, we aimed to set even more stringent latency requirements.

Therefore, a configuration comprising a 4-layer encoder and a 1-layer decoder represents the most cost-effective choice.

# 4.2 Generation Latency.

Figure 3 shows a comparative analysis of the latency incurred by CV-IME (ours), Base+Beam search (beam) and Base+TopK sample (topk) (Fan et al., 2018) in generating a varying number of candidates. As can be observed from the figure, the latency of CV-IME when generating four candidates is nearly identical to that of the base model when producing two candidates, which demonstrates the superiority of the CV-IME approach in recalling more candidate results under low-resource conditions. We also observed that the latency of the topk significantly increases when generating on CPU devices, which may be attributed to the higher computational complexity of the multinomial function in PyTorch on CPU.

## 4.3 PD Benchmark.

Table 3, 4 show the results of PD. CV-IME-i means the results of CV-IME using i-th hybrid latent variable. Beam represents the beam search. From these results, we can observe that: (1) Our models outperform the baselines on the PD benchmark; (2) our models show more significant results in the task of converting from a abbreviated pinyin to characters; (3) Under the condition of equivalent

Model	Top-N	26-1	26-key IME		
		Perfect	Abbreviated		
Google IME	P@1	70.90%	-	_	
On-OMWA	P@1	64.40%	-	-	
On-P2C	P@1	71.30%	-	-	
Base-Beam1	P@1	71.53%	21.65%	20	
CV-IME-1	P@1	71.43%	23.04%	20	
CV-IME-2	P@1	67.14%	21.20%	20	
CV-IME-3	P@1	68.82%	22.09%	20	
CV-IME-4	P@1	68.64%	20.18%	20	
Google IME	P@10	82.30%	_	_	
On-OMWA	P@10	77.90%	-	-	
On-P2C	P@10	81.30%	-	-	
Base-Beam2	P@2	81.08%	27.32%	27	
CV-IME	P@4	82.97%	29.90%	27	

Table 3: Results of different methods over PD. Each score is averaged over all context-target configurations.

Model	Top-N	9-k	9-key IME		
		Perfect	Abbreviated		
Base-Beam1	P@1	54.49%	10.14%	20	
CV-IME-1	P@1	45.54%	12.04%	20	
CV-IME-2	P@1	52.05%	9.90%	20	
CV-IME-3	P@1	57.85%	10.80%	20	
CV-IME-4	P@1	49.22%	10.22%	19	
Base-Beam2	P@2	65.62%	13.84%	27	
CV-IME	P@4	66.06%	15.94%	27	

Table 4: Results of different 9-key IMEs over PD.

time expenditure, our model is capable of generating more candidates and achieve better accuracy compared to the baselines; (4) The four hybrid latent variables exhibited a clustering effect in the 9-key IME, where CV-IME-1 excelled in abbreviated pinyin and CV-IME-3 in perfect pinyin. However, this phenomenon was not replicated in the 26key IMEs. These results suggest that the current training methodology for hybrid latent variables has certain limitations, as it struggles to encourage different latent variables to focus on distinct data categories during training. This will be a direction for our future research.

## 4.4 WD Benchmark.

Table 5 reports the results of different domains over WD. We have selected the results from four domains where the differences between CV-IME and the Base model are the smallest and the largest under various pinyin input patterns. This result demonstrates that the CV-IME achieves a superior performance than base model in terms of all domains in WD. We also conduct experiments with different configurations on WD, which are detailed

26-key Perfect	Entertainment (%)	Education (%)	Journey (%)	Agriculture (%)
Base+Beam2	77.67±0.00	80.99±0.00	74.71±0.00	73.68%±0.00
CV-IME	<b>79.48±0.29</b> (∆ 1.80)	<b>83.26±0.10</b> (△ 2.27)	<b>78.73±0.02</b> (△ 4.02)	<b>78.65±0.19</b> (△ 4.97)
9-key Perfect	Entertainment (%)	Sports (%)           59.93±0.00           62.12±0.06 (∆ 2.19)	Real Estate (%)	Agriculture (%)
Base+Beam2	60.55±0.00		60.94±0.00	57.02 $\pm$ 0.00
CV-IME	<b>62.48±0.20</b> (∆ 1.94)		<b>65.76±0.07</b> (∆ 4.82)	<b>61.94<math>\pm</math>0.14 (<math>\triangle</math> 4.92)</b>
26-key Abbreviated	Journey (%)	Sports (%)	Real Estate (%)	Economy (%)
Base+Beam2	19.75±0.00	20.65±0.00	20.40±0.00	20.85±0.00
CV-IME	<b>20.87±0.09</b> (∆ 1.12)	<b>22.18±0.20</b> (∆ 1.53)	<b>24.38±0.02</b> (△ 3.98)	<b>25.32±0.22</b> (△ 4.47)
9-key Abbreviated	Agriculture (%)	Automobile (%)	Real Estate (%)	International (%)
Base+Beam2	8.60±0.00	9.25±0.00	8.60±0.00	7.85 $\pm$ 0.00
CV-IME	<b>9.35±0.07</b> (△ 0.75)	<b>10.33±0.15</b> (∆ 1.08)	<b>12.15±0.15</b> (△ 3.55)	<b>11.40<math>\pm</math>0.08</b> ( $\triangle$ 3.55)

Table 5: Results of different domains over WD.

9-key IMEs	Perfect		Abbreviated		
	PD (%)	WD (%)	PD (%)	WD (%)	
CLS	65.690	62.901	15.878	10.190	
CHVT	65.781	62.992	15.924	10.165	
CV-IME	66.056	<b>63.269</b>	15.935	<b>10.310</b>	
w/o. CLV	<b>66.343</b>	62.656	<b>16.899</b>	10.259	
w/o. DLV	57.649	53.446	7.885	4.420	

Table 6: The results of ablation study.

in the appendix D.3.

#### 4.5 Ablation Study.

Table 6 presents the results of ablation experiments, where "CLS" and "CHVT" are two alternative strategies for constructing hybrid latent variables that differ from our approach:

- "CLS" means using the [CLS] token to determine the prior distribution of continuous latent variables.
- "CHVT" stands for Conditional Hybrid Variational Transformer (Sun et al., 2023), which also utilizes hybrid latent variables, but it is primarily used in dialogue tasks.

Compared to CLS and CHVT, CV-IME achieves better performance on PD and WD benchmarks, indicating the effectiveness of the proposed strategy in the pinyin-to-character task. Moreover, "w/o. CLV" and "w/o. DLV" denote the CV-IME model variants with the continuous latent variables (CLV) and discrete latent variables (DLV) removed, respectively. The findings indicate that DLV may excel in accuracy but fall short in generalizing across diverse scenarios. Therefore, the "w/o. CLV" performs well on news data (PD) similar to the training set but not as well on OOD data (WD). Similarly, CLV may excel in diversity but compromise on precision, which exhibit a marked decline in performance when the DLV is removed. Furthermore, from perfect to abbreviated pinyin, the degradation in performance becomes more pronounced.

#### 4.6 Discussion on Top-1 Results.

For the Pinyin-to-character task, there is a clear correlation between the top-1 accuracy and the distribution differences between training and testing data. This is due to the presence of one-to-many samples in the data, where identical Pinyin corresponds to completely different outcomes. If the most proportionate samples in the test data also happen to be the highest probability samples in the training data, the model's top-1 results are likely to be high. The CV-IME is proposed to internalize one-to-many data through latent variables, mitigating the excessive influence of training data distribution on the test data distribution. Experimental results reveal differentiated outcomes presented by various hybrid latent variables, indicating that latent variables can indeed diversify data distributions. However, the current training is unsupervised, and the overall differentiation effect is not pronounced, necessitating further research.

# 5 Conclusion

This paper introduces the conditional variational mechanism into the IME model, presenting the CV-IME model. By incorporating hybrid latent variables, CV-IME enhances diversity while maintaining the quality of generated results. In comparison to existing IME models, the experimental results demonstrate that CV-IME can recall more diverse and accurate results within similar time constraints, exhibiting significant advantages in low-resource scenarios, such as offline mobile devices.

# Limitations

**Application Scenarios.** The CV-IME model is proposed to effectively mitigate the severe one-tomany problem in the task of pinyin-to-character conversion under low-resource conditions (e.g., on the mobile phone devices). Therefore, under conditions of abundant computational and storage resources, the introduction of larger pre-trained language models with more parameters may yield better results. After all, the practical application of latent variable-based pre-training techniques remains to be tested, which also constitutes one of our future research directions.

**Data Distribution.** The training data and evaluation benchmarks are extracted from different Chinese corpora, which are not not consistent with the data generated by real users of the Pinyin IME, and there are certain differences in their distributions. Consequently, in constructing our training dataset, we selected news data closely aligned with the PD benchmark to approximate independent and identically distributed scenarios (comparison with PD), and out of domain scenarios (comparison with WD). Through the aforementioned configurations, we have rudimentarily simulated real-world scenarios of general distribution and user-specific personalization, which to some extent, demonstrates the efficacy of our approach in practical applications.

Flexibility and Differentiation. The hyperparameters (e.g., the number of discrete latent variables, the annealing steps of KL and so on) need to be determined through multiple experiments, which cannot be set adaptively. Additionally, the experimental results indicate that the different mixed latent variables are not sufficiently independent, as the corresponding generated texts are not entirely distinct. This may be attributed to the current training methodology being guided by unsupervised gradient backpropagation. It might be necessary to introduce regularization terms or to devise a novel training approach to enhance the discriminability between the mixed latent variables. These initial promising results for distinguishing different hybrid latent variables for recalling diverse candidate results will hopefully lead to future work in this interesting direction.

### **Ethics Statement**

We acknowledge and ensure that our study is compatible with the provided Code of Ethics. Pinyin input method engine (IME) is crucial for building connection between Chinese people and mobile applications, which is an import topic in Chinese natural language process field. All our experiments are conducted on public available datasets to avoid ethical concerns. All terms for using these datasets are strictly followed in our study. There are no direct ethical concerns in our research.

## Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. This work is supported by the Beijing Natural Science Foundation, China (Nos. 4222037, L181010).

#### References

- David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cogn. Sci.*, 9(1):147–169.
- Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: pre-trained dialogue generation model with discrete latent variable. In *ACL*, pages 85–96. ACL.
- Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*, volume 4. Springer.
- Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *CoNLL*, pages 10–21.
- Hongshen Chen, Zhaochun Ren, Jiliang Tang, Yihong Eric Zhao, and Dawei Yin. 2018. Hierarchical variational memory network for dialogue generation. In *WWW*, pages 1653–1662. ACM.
- Kuan-Yu Chen, Hung-Shin Lee, Chung-Han Lee, Hsin-Min Wang, and Hsin-Hsi Chen. 2013. A study of language modeling for chinese spelling check. In *SIGHAN*, pages 79–83. Asian Federation of NLP.
- Wei Chen, Yeyun Gong, Song Wang, Bolun Yao, Weizhen Qi, Zhongyu Wei, Xiaowu Hu, Bartuer Zhou, Yi Mao, Weizhu Chen, Biao Cheng, and Nan Duan. 2022. Dialogved: A pre-trained latent variable encoder-decoder model for dialog response generation. In ACL, pages 4852–4864. ACL.
- Zheng Chen and Kai-Fu Lee. 2000. A new statistical approach to chinese pinyin input. In *ACL*, pages 241–247. ACL.
- Hsun-wen Chiu, Jian-Cheng Wu, and Jason S. Chang. 2013. Chinese spelling checker based on statistical machine translation. In *SIGHAN*, pages 49–53. Asian Federation of NLP.

- Keyu Ding, Yongcan Wang, Zihang Xu, Zhenzhen Jia, Shijin Wang, Cong Liu, and Enhong Chen. 2023. Generative input: Towards next-generation input methods paradigm. *CoRR*, abs/2311.01166.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *ACL*, pages 889–898, Melbourne, Australia. ACL.
- Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. 2021. Transformerbased conditional variational autoencoder for controllable story generation. *CoRR*, abs/2101.00828.
- Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019. Jointly optimizing diversity and relevance in neural response generation. In *NAACL-HLT (1)*, pages 1229–1238.
- Dongxu Han and Baobao Chang. 2013. A maximum entropy approach to chinese spelling check. In *SIGHAN*, pages 74–78. Asian Federation of NLP.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*. OpenReview.net.
- Yong Hu, Fandong Meng, and Jie Zhou. 2022. CSCD-IME: correcting spelling errors generated by pinyin IME. *CoRR*, abs/2211.08788.
- Guoping Huang, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2015. A new input method for human translators: Integrating machine translation effectively and imperceptibly. In *IJCAI*, pages 1163–1169. AAAI.
- Yafang Huang, Zuchao Li, Zhuosheng Zhang, and Hai Zhao. 2018. Moon IME: neural-based chinese pinyin aided input method with customizable association. In *ACL*, pages 140–145. ACL.
- Yafang Huang and Hai Zhao. 2018. Chinese pinyin aided ime, input what you have not keystroked yet. In *EMNLP*, pages 2923–2929. ACL.
- Zhongye Jia and Hai Zhao. 2014. A joint graph model for pinyin-to-chinese conversion with typo correction. In *ACL*, pages 1512–1523. ACL.
- Diederik P. Kingma and Max Welling. 2014. Autoencoding variational bayes. In *ICLR*.
- Haizhou Li, Min Zhang, and Jian Su. 2004. A joint source-channel model for machine transliteration. In *ACL*, pages 159–166. ACL.
- Zhaojiang Lin, Genta Indra Winata, Peng Xu, Zihan Liu, and Pascale Fung. 2020. Variational transformers for diverse response generation. *CoRR*, abs/2003.12738.
- Shulin Liu, Tao Yang, Tianchi Yue, Feng Zhang, and Di Wang. 2021. PLOME: pre-training with misspelled knowledge for chinese spelling correction. In *ACL/IJCNLP*, pages 2991–3000. ACL.

- Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2022. Locally typical sampling. *Transactions of the ACL*.
- Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. 2017. A conditional variational framework for dialog generation. In *ACL* (2), pages 504–509.
- Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *NeurIPS*, pages 3483–3491.
- Bin Sun, Shaoxiong Feng, Yiwei Li, Jiamou Liu, and Kan Li. 2021. Generating relevant and coherent dialogue responses using self-separated conditional variational autoencoders. In ACL/IJCNLP, pages 5624– 5637. ACL.
- Bin Sun, Yitong Li, Fei Mi, Weichao Wang, Yiwei Li, and Kan Li. 2023. Towards diverse, relevant and coherent open-domain dialogue generation via hybrid latent variables. In *AAAI*, pages 13600–13608. AAAI Press.
- Minghuan Tan, Yong Dai, Duyu Tang, Zhangyin Feng, Guoping Huang, Jing Jiang, Jiwei Li, and Shuming Shi. 2022. Exploring and adapting chinese GPT to pinyin input method. In *ACL*, pages 1899–1909. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.
- Liwei Wang, Alexander G. Schwing, and Svetlana Lazebnik. 2017. Diverse and accurate image description using a variational auto-encoder with an additive gaussian encoding space. In NIPS, pages 5756–5766.
- Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. 2016. Attribute2image: Conditional image generation from visual attributes. In *ECCV*, volume 9908, pages 776–791.
- Shaohua Yang, Hai Zhao, and Bao-liang Lu. 2012. A machine translation approach for chinese wholesentence pinyin-to-character conversion. In *PACLIC*, pages 333–342.
- Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, Xiao Liu, Yukuo Cen, Xu Zou, Zhilin Yang, and Jie Tang. 2021. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. *AI Open*, 2:65–68.
- Xihu Zhang, Chu Wei, and Hai Zhao. 2017. Tracing a loose wordhood for chinese input method engine. *CoRR*, abs/1712.04158.
- Zhuosheng Zhang, Yafang Huang, and Hai Zhao. 2019. Open vocabulary learning for neural chinese pinyin IME. In *ACL*, pages 1584–1594. ACL.

- Tiancheng Zhao, Ran Zhao, and Maxine Eskénazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In ACL (1), pages 654–664.
- Yabin Zheng, Chen Li, and Maosong Sun. 2011. CHIME: an efficient error-tolerant chinese pinyin input method. In *IJCAI*, pages 2551–2556. IJ-CAI/AAAI.

### A Related Work

#### A.1 Input Method Engine

Input method engines (IMEs) are important tools to connect users with mobile applications, By providing an efficient and user-friendly interface, it enables users to input text with ease, thereby enhancing their overall experience with the apps. Different from alphabetic languages, the input of some Asian language (i.e. Chinese) characters must rely on the IMEs. In China, there are two common Pinyin IMEs for cellphones: the 9-key IMEs and the 26-key IMEs (Hu et al., 2022). Previous research on Chinese IMEs primarily focused on three tasks associated with the 26-key keyboard:

(1) The perfect (abbreviated) pinyin to Chinese characters (PTC) task (Chen and Lee, 2000; Li et al., 2004; Zhang et al., 2017; Huang et al., 2018; Zhang et al., 2019; Tan et al., 2022). This task represents the most fundamental aspect of the Pinyin IME, revealing the core performance capabilities of the IME model.

(2) The input noise correction tasks, such as input typo correction (Zheng et al., 2011; Jia and Zhao, 2014; Liu et al., 2021) and Chinese spelling check (Chiu et al., 2013; Han and Chang, 2013; Chen et al., 2013). Due to the limited screen size of mobile phones, users may accidentally press the wrong keys on a 26-key keyboard, leading to incorrect pinyin syllables being entered. For instance, while typing 'songgei' (送给, give), the 'i' might be mistakenly hit as 'u', resulting in 'songgeu'. Identifying the noise caused by these accidental touches and correcting them to output what the user intended is a significant challenge.

(3) The intelligent association task (Huang et al., 2015; Huang and Zhao, 2018; Ding et al., 2023). Usually Pinyin IMEs simply predict a list of character sequences for user choice only according to the pinyin input. However, Chinese inputting is a multi-turn procedure, which can be supposed to be exploited for further user experience promoting. This task is a commonly used input assistance function, which predicts possible next sentences

based on the content already entered by the user for selection, to improve input efficiency.

#### A.2 Conditional Variational Mechanism

Conditional variational mechanisms (Kingma and Welling, 2014; Sohn et al., 2015; Yan et al., 2016; Bowman et al., 2016) are powerful tools in text generation task, and they are usually used in dialogue generation models. By using continuous latent variables, previous conditional variational mechanisms are introduced into dialogue generation models to tackle short, dull and general responses problem (Shen et al., 2017; Zhao et al., 2017; Chen et al., 2018; Lin et al., 2020; Fang et al., 2021; Sun et al., 2022; Sun et al., 2023).

The conditional variational mechanism estimates the posterior probability distributions p(z|c, r) and the prior probability distribution p(z|c) of latent variable z based on the dialogue corpora, where c denotes the context, r denotes the response, and a context and a response together constitute a singleturn dialogue pair. During training, these models sample the continuous latent variable z from p(z|c, r) and maximize the conditional probability p(r|c, z) to encode context and response into latent space. Meanwhile, they also minimize the KL-divergence  $D_{KL}(p(z|c, r)||p(z|c))$  to bring the two distributions closer together, thus constraining the continuous latent variables z sampled from the prior distribution p(z|c) for inference.

In practically, the continuous latent variables effectively help dialogue models to generate diverse responses. Nevertheless, owing to the one-to-many and many-to-one phenomena, the continuous latent variables frequently struggle to encapsulate the precise contextual semantics, leading to responses that are irrelevant and lack coherence (Sun et al., 2021). Different from the continuous latent variables, discrete latent variables are better at producing relevant and coherent responses. For example, Bao et al. (2020) uses Latent Act Recognition to model the relationship between discrete latent variables and multiple responses, and proposes Response Selection to chose the generated responses of most coherent with the context. However, owing to their limited scale, discrete latent variables might encapsulate a narrower range of features compared to their continuous counterparts.

Therefore, combining continuous and discrete latent variables presents a promising direction. By doing so, the strengths of each can be harnessed and their weaknesses mitigated, allowing for a more balanced approach that capitalizes on the diversity provided by continuous variables and the specificity afforded by discrete variables. This hybrid approach could potentially lead to more robust and nuanced models that better capture the complexities of the data they are designed to represent. Based on this, (Sun et al., 2023) propose a hybrid latent variable strategy and a Conditional Hybrid Variational Transformer (CHVT) for dialogue generation task. Different from the CHVT, our CV-IME focus on the pinyin-to-characters task. Owing to the pronounced alignment between the input pinyin sequences and the target character sequences within IME data, the conventional approach to information interchange employed during the construction of continuous latent variables in the CHVT framework can inadvertently overlook salient character sequence details. This oversight has the potential to compromise model performance. In response to this challenge, we introduce an innovative strategy for the formulation of continuous latent variables. This strategy is designed to intensify the interaction of information between pinyin and character sequences, consequently bolstering the efficacy of the training phase.

#### A.3 Generation Methods

Beam Search (BS), a popular breadth-first decoding method, is widely used in text generation task. Unfortunately, they inherently exhibit a deficiency in diversity, which frequently results in performance degradation within human-like contexts (Holtzman et al., 2020). Additionally, BS necessitates the computation of cumulative scores for each candidate during the decoding process, and concurrently requires the sorting and recombination of samples, thereby augmenting the computational burden. Under conditions of constrained computational resources, this may impede the realization of its advantages.

To enrich the diversity of BS, stochastic decoding strategies are introduced in the generation phrase. Ancestral sampling (AS) (Bishop and Nasrabadi, 2006) is the most straightforward but less effective sampling method. Temperature sampling (Ackley et al., 1985) is an improvement of AS, which introduces temperature to shape the probability distribution. However, due to the randomness, both of them will damage the quality of generated results. To mitigate this problem, top-k (Fan et al., 2018), nucleus sampling (Holtzman et al., 2020), and locally typical (Meister et al., 2022) sampling are proposed to truncate the distributions, which aim at improving quality while preserving diversity. However, the truncation and re-scaling of probabilities also demand additional computational effort, similarly presenting challenges with respect to latency.

Diverging from the aforementioned approaches, we introduce a conditional variational mechanism into the IME model, optimizing the sampling process through adjustments to the model structure, while exclusively employing greedy search to circumvent additional computational overhead. Leveraging the latent variables in sampling, CV-IME is capable of enhancing the diversity of generated results under conditions of limited latency.

#### **B** Method

**Construction of Continuous Latent Variables.** The prior and recognition network are responsible for estimating the prior and the posterior distribution of continues latent variables. We first use the Transformer encoder to encode the input sequence  $(\mathbf{x} = x_1, x_2, \dots, x_n)$  to obtain its final hidden state  $(\mathbf{h} = h_1, h_2, \dots, h_n)$  as prior memory, where ndenotes the length of c. Then, we use the same encoder to encode the input and target sequence  $(\mathbf{x}' = x_1, \dots, x_n, \dots, x_{n+m})$  to obtain the posterior memory  $(\mathbf{h}' = h'_1, \dots, h'_n, \dots, h'_{n+m})$ , where m means the length of r. Next, we use prior memory to recompute the posterior memory:

$$\mathbf{h}' = \text{SoftMax}(\mathbf{h} \cdot \mathbf{h}'^T) \cdot \mathbf{h}'$$
(2)

Finally, similar with previous works (Bowman et al., 2016; Zhao et al., 2017; Shen et al., 2017) that assume z follows isotropic Gaussian distribution, we use fully-connected networks as  $p_{\theta}(z|c) \sim \mathcal{N}(\mu, \sigma^2 \mathbf{I})$  and  $q_{\phi}(z|c, r) \sim \mathcal{N}(\mu', \sigma'^2 \mathbf{I})$ :

$$\begin{pmatrix} \mu_1, \dots, \mu_n \\ \log(\sigma_1^2), \dots, \log(\sigma_n^2) \end{pmatrix} = \tanh(\begin{pmatrix} h_1 \\ \cdots \\ h_n \end{pmatrix} W_d) W_u$$
$$\begin{pmatrix} \mu'_1, \dots, \mu'_n \\ \log(\sigma'^{2}_1), \dots, \log(\sigma'^{2}_n) \end{pmatrix} = \begin{pmatrix} [h_1; h'_1] \\ \cdots \\ [h_n; h'_n] \end{pmatrix} W'_u ,$$

where  $W_{\{d,u\}}$ ,  $W'_u$  are trainable parameters of prior network and recognition network. At this point we have *n* token-level probability distributions for *n* tokens in *c*. To take full use of these distributions, we follow the *additive Gaussian mixing* (Wang et al., 2017) to compute the sentence-level distribution:

$$p_{\theta}(z_s|c) \sim \mathcal{N}(\sum_{i=1}^n w_i \mu_i, \prod_{i=1}^n \sigma_i^{2w_i})$$
$$q_{\phi}(z'_s|c, r) \sim \mathcal{N}(\sum_{i=1}^n w_i \mu'_i, \prod_{i=1}^n \sigma_i'^{2w_i})$$

where  $z_s$  represents the sentence-level latent variable,  $w_i$  denotes the weight of the *i*-th distribution.

Finally, we use the reparameterization trick (Kingma and Welling, 2014; Zhao et al., 2017) to obtain samples of  $z_s$  either from  $p(z_s|c, r)$  (training) or  $p(z_s|c)$  (inference). The sentence-level latent variable  $z_s$  will be used for constructing the hybrid latent variable afterwards.

**Construction of Hybrid Latent Variables.** To build the hybrid latent variables H, during training, we first sample the  $z'_s$  from the  $p(z'_s|c, r)$  and then expanded K times that make it added to the discrete latent variables M:

$$H = \begin{pmatrix} z'_s + M[1] \\ \cdots \\ z'_s + M[K] \end{pmatrix}$$

where K represents the number of discrete latent variables.

**Loss Function.** During training, CV-IME introduce the *self-separation training* and aims to maximizing the variational lower bound of the conditional log likelihood (Kingma and Welling, 2014; Sohn et al., 2015; Yan et al., 2016):

$$\begin{aligned} \mathcal{L}(\theta, \phi, \Omega, M; r, c) \\ &= \sum_{i=1}^{K} \alpha_i \mathbb{E}_{q_{\phi}(z'_s | r, c)} \left[ \log p(r | [H_i; c]) \right] \\ &- \lambda \operatorname{D}_{\mathrm{KL}}(q_{\phi}(z'_s | r, c) | | p_{\theta}(z_s | c)) \\ \alpha_i &= \begin{cases} 1 & if \ \mathbb{E}_i = \max(\mathbb{E}_1, \cdots, \mathbb{E}_K) \\ 0 & otherwise \end{cases} \\ \mathbb{E}_i &= \mathbb{E}_{q_{\phi}(z'_s | r, c)} \left[ \log p(r | [H_i; c]) \right], \end{aligned}$$

where  $\theta, \phi, \psi, \Omega, M$  are parameters of CV-IME, and  $\lambda$  is the scale factor of KL divergence.

**Inference Phase.** During inference, CV-IME use the prior distribution  $p(z_s|c)$  to sample the sentence-level continuous latent variable  $z_s$  and mix  $z_s$  with discrete latent variables to construct hybrid latent variables. Based on the K discrete latent variables, CV-IME can directly generate K results for the same input.

## **C** Experimental Settings

Benchmarks. We used two benchmarks:

(1) PD benchmark, a commonly used benchmark dataset for the Chinese IME task, is extracted from the People's Daily from 1992 to 1998 that has word segmentation annotations by Peking University. It contains 2,000 segments of consecutive Chinese characters for testing. For each test case, the input pinyin are all perfect pinyin and the context is null.

(2) WD benchmarks is extracted from the Wu-DaoCorpora (Yuan et al., 2021) that contains 3TB Chinese corpus collected from 822 million Web pages. Tan et al. (2022) randomly select 16 domains from WuDaoCorpora, and segment those documents into sentences. For each sentence, they randomly selected a context ranging from 0-3, 4-9, and 10+ words, while continuously selecting a target of 1-3, 4-9, or 10+ words. Each context-target length tuple like (0-3, 1-3) serves as an evaluation configuration and contains 2,000 test instances.

Training Data. To train our CV-IME and the base model, we built a new pinyin-to-character dataset based on the news2016 corpus<sup>5</sup>. The news2016 corpus comprises 2.5 million news articles, each containing keywords and descriptions. Initially, we extract paragraphs from the data and segment them into sentences using periods, exclamation points, and question marks as delimiters. Subsequently, we divide each sentence from the end into two parts: context and target, omitting segmentation points where the target part includes numbers or special characters. Following this, we retain the case with a probability of 50%. Subsequent to the initial processing, we incorporate the 'pypinyin' package to construct Pinyin sequences for the target portion of the retained cases. For each case, we determine with a 50% probability whether the data will represent a 'perfect' Pinyin or an 'abbreviated' Pinyin. Ultimately, we retained a total of 19,330,170 training samples. Table 1 shows the statistics of our training set.

**Training Detail.** The hidden size of all models is set to 512. The base model consists of 4 layers of encoder and 1 layer of decoder. The CV-IME employs a Transformer model with 2 encoder layers and 1 decoder layer, and additionally incorporates two fully-connected layers as a prior network. The maximum length of input sequence (pinyin + context) and result are set to 42 and 20, respectively.

<sup>&</sup>lt;sup>5</sup>https://github.com/brightmart/nlp\_chinese\_corpus

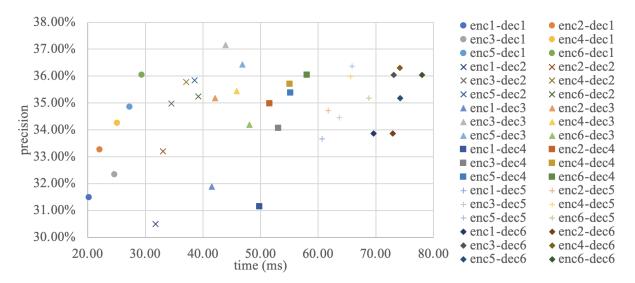


Figure 4: Results of different encoder-decoder layers configurations over PD using 9-key IME

Perfect	Sample-1	Sample-2	Sample-3	Sample	Aver.	Time
Base+Beam1	84.434%±0.00%	76.002%±0.00%	69.624%±0.00%	56.040%±0.00%	71.525%	19
CV-IME-1	$83.050\% \pm 0.29\%$	$76.536\% {\pm} 0.10\%$	$70.075\% \pm 0.12\%$	$56.040\% \pm 0.52\%$	71.425%	20
CV-IME-2	80.097%±0.21%	71.543%±0.29%	$65.681\% {\pm} 0.21\%$	$51.221\% {\pm} 0.43\%$	67.135%	19
CV-IME-3	$81.465\% {\pm} 0.21\%$	$73.480\% {\pm} 0.05\%$	66.917%±0.22%	53.421%±0.31%	68.821%	20
CV-IME-4	79.696%±0.17%	$73.163\%{\pm}0.14\%$	$67.034\%{\pm}0.40\%$	$54.675\%{\pm}0.11\%$	68.642%	19
Base+Beam2	90.841%±0.00%	85.471%±0.00%	$79.950\% \pm 0.00\%$	68.053%±0.00%	81.079%	27
CV-IME	$93.093\%\pm0.15\%$	$87.141\% \pm 0.09\%$	$82.038\%\pm0.13\%$	69.593%±0.33%	82.966%	27
Abbreviated	Sample-1	Sample-2	Sample-3	Sample	Aver.	Time
Abbreviated Base+Beam1	Sample-1 44.645%±0.00%	Sample-2 24.649%±0.00%	Sample-3 13.677%±0.00%	Sample 3.614%±0.00%	Aver. 21.646%	Time 20
	•	•	•	•		
Base+Beam1	44.645%±0.00%	24.649%±0.00%	13.677%±0.00%	3.614%±0.00%	21.646%	20
Base+Beam1 CV-IME-1	44.645%±0.00% 45.362%±0.23%	24.649%±0.00% 26.703%±0.18%	13.677%±0.00% 15.715%±0.17%	3.614%±0.00% 4.367%±0.21%	21.646% 23.037%	20 20
Base+Beam1 CV-IME-1 CV-IME-2	44.645%±0.00% 45.362%±0.23% 41.508%±0.17%	24.649%±0.00% 26.703%±0.18% 24.933%±0.17%	13.677%±0.00% 15.715%±0.17% 14.412%±0.09%	3.614%±0.00% 4.367%±0.21% 3.932%±0.09%	21.646% 23.037% 21.196%	20 20 20
Base+Beam1 CV-IME-1 CV-IME-2 CV-IME-3	44.645%±0.00% 45.362%±0.23% 41.508%±0.17% 42.809%±0.05%	24.649%±0.00% 26.703%±0.18% 24.933%±0.17% 26.486%±0.05%	13.677%±0.00% 15.715%±0.17% 14.412%±0.09% 15.230%±0.04%	$3.614\% \pm 0.00\%$ $4.367\% \pm 0.21\%$ $3.932\% \pm 0.09\%$ $3.849\% \pm 0.06\%$	21.646% 23.037% 21.196% 22.094%	20 20 20 20 20

Table 7: Results of different context-target length configurations over PD using 26-key IME.

We set the batch sizes to 1024 and 256 for base model and CV-IME, respectively. Adam is used for optimization. The initial learning rate is set to 0.0001. We also introduce KL annealing trick to leverage the KL divergence during the training. The KL weight increases linearly from 0 to 1 in the first 3000000 batches. We train all models in 100 epochs on four A100 GPU cards with Pytorch, and save the model parameters when the validation loss reaching minimum.

### **D** Experimental Results

## D.1 Model Structure Selection

Figure 4 elucidates the following points: (1) Merely augmenting the number of layers in the decoder significantly increases the generation latency without improving the accuracy of the generated results; (2) A model with a single-layer encoder and a sixlayer decoder exhibits a base latency of 69.62 on CPU devices, yet its accuracy is inferior to that of a model with a four-layer encoder and a single-layer decoder; (3) Increasing the number of encoder layers effectively enhances the accuracy of the generated results. Consequently, retaining a single-layer decoder offers the best cost-effectiveness in lowresource scenarios, and, where possible, augmenting the number of encoder layers under constrained conditions contributes to improved accuracy.

#### **D.2 PD Benchmark**

Table 7 and Table 8 show the results of different context-target length configurations over PD benchmakrs. Sample-i means that the target in the set

Perfect	Sample-1	Sample-2	Sample-3	Sample	Aver.	Time
Base+Beam1	72.322%±0.00%	60.521%±0.00%	50.877%±0.00%	34.257%±0.00%	54.494%	19
CV-IME-1	48.415%±0.25%	$54.142\%{\pm}0.38\%$	$48.488\% {\pm} 0.20\%$	$31.122\% {\pm} 0.61\%$	45.542%	20
CV-IME-2	$66.867\%{\pm}0.36\%$	$58.333\% {\pm 0.10\%}$	50.192%±0.33%	$32.827\% {\pm} 0.10\%$	52.055%	19
CV-IME-3	$75.592\% \pm 0.10\%$	63.945%±0.40%	$54.737\% \pm 0.25\%$	$37.118\% \pm 0.21\%$	57.848%	19
CV-IME-4	$56.390\% {\pm} 0.31\%$	$56.446\% {\pm} 0.16\%$	$49.925\%{\pm}0.05\%$	$34.103\%{\pm}0.45\%$	49.216%	19
Base+Beam2	81.431%±0.00%	72.295%±0.00%	$63.058\% {\pm} 0.00\%$	45.677%±0.00%	65.615%	27
CV-IME	$83.600\%\pm 0.23\%$	$72.495\%{\pm}0.40\%$	$63.442\% {\pm} 0.33\%$	44.686%±0.20%	66.056%	27
Abbreviated	Sample-1	Sample-2	Sample-3	Sample	Aver.	Time
Abbreviated Base+Beam1	Sample-1 27.327%±0.00%	Sample-2 9.469%±0.00%	Sample-3 3.507%±0.00%	Sample 0.251%±0.00%	Aver. 10.139%	Time 20
	1	•	•	•		
Base+Beam1	27.327%±0.00%	9.469%±0.00%	3.507%±0.00%	0.251%±0.00%	10.139%	20
Base+Beam1 CV-IME-1	27.327%±0.00% 29.696%±0.43%	9.469%±0.00% 12.391%±0.13%	3.507%±0.00% 4.843%±0.23%	0.251%±0.00% 1.222%±0.03%	10.139% <b>12.038%</b>	20 20
Base+Beam1 CV-IME-1 CV-IME-2	27.327%±0.00% 29.696%±0.43% 24.975%±0.49%	9.469%±0.00% <b>12.391%±0.13%</b> 9.786%±0.30%	3.507%±0.00% 4.843%±0.23% 4.225%±0.10%	0.251%±0.00% 1.222%±0.03% 0.602%±0.13%	10.139% <b>12.038%</b> 9.897%	20 20 20
Base+Beam1 CV-IME-1 CV-IME-2 CV-IME-3	27.327%±0.00% 29.696%±0.43% 24.975%±0.49% 27.127%±0.28%	9.469%±0.00% 12.391%±0.13% 9.786%±0.30% 11.055%±0.32%	$3.507\% \pm 0.00\%$ <b>4.843% ± 0.23%</b> 4.225% ± 0.10% 4.409% ± 0.05%	$\begin{array}{c} 0.251\% \pm 0.00\% \\ \textbf{1.222\%} \pm \textbf{0.03\%} \\ 0.602\% \pm 0.13\% \\ 0.602\% \pm 0.13\% \end{array}$	10.139% <b>12.038%</b> 9.897% 10.798%	20 20 20 20

Table 8: Results of different context-target length configurations over PD using 9-key IME.

<i>Model</i>	Sports	Journey	Games	Culture
Base+Beam2	77.190%±0.000%	74.714%±0.000%	75.684%±0.000%	69.808%±0.000%
CV-IME	80.439%±0.107%	<b>78.730%±0.024%</b>	<b>78.833%±0.235%</b>	72.760%±0.073%
<i>Model</i>	Military	Real Estate	Technology	Finance
Base+Beam2	73.007%±0.000%	77.650%±0.000%	79.677%±0.000%	79.300%±0.000%
CV-IME	<b>75.977%±0.195%</b>	<b>81.027%±0.107%</b>	82.560%±0.149%	<b>82.959%±0.154%</b>
<i>Model</i>	Education	Economy	Entertainment	International
Base+Beam2	80.995%±0.000%	78.486%±0.000%	77.675%±0.000%	77.207%±0.000%
CV-IME	<b>83.264%±0.099%</b>	<b>80.802%±0.191%</b>	<b>79.475%±0.294%</b>	<b>79.803%±0.297%</b>
<i>Model</i>	Medical	Automobile	Agriculture	Society
Base+Beam2	81.584%±0.000%	78.486%±0.000%	73.684%±0.000%	78.127%±0.000%
CV-IME	<b>84.260%±0.227%</b>	<b>81.393%±0.242%</b>	<b>78.655%±0.190%</b>	80.666%±0.310%

Table 9: Results of different domains over WD using 26-key IME perfect pinyin mode.

<i>Model</i>	Sports	Journey	Games	Culture
Base+Beam2	20.650%±0.000%	19.750%±0.000%	16.650%±0.000%	17.150%±0.000%
CV-IME	22.183%±0.201%	20.867%±0.094%	20.117%±0.062%	18.967%±0.085%
<i>Model</i>	Military	Real Estate	Technology	Finance
Base+Beam2	16.150%±0.000%	20.400%±0.000%	19.450%±0.000%	21.750%±0.000%
CV-IME	<b>19.033%±0.094%</b>	<b>24.383%±0.024%</b>	<b>23.183%±0.103%</b>	<b>25.650%±0.212%</b>
<i>Model</i>	Education	Economy	Entertainment	International
Base+Beam2	22.000%±0.000%	20.850%±0.000%	19.850%±0.000%	19.150%±0.000%
CV-IME	<b>24.967%±0.306%</b>	<b>25.317%±0.225%</b>	<b>23.450%±0.283%</b>	<b>22.300%±0.204%</b>
<i>Model</i>	Medical	Automobile	Agriculture	Society
Base+Beam2	26.800%±0.000%	20.850%±0.000%	19.250%±0.000%	21.100%±0.000%
CV-IME	<b>30.183%±0.295%</b>	<b>24.017%±0.272%</b>	<b>23.200%±0.082%</b>	<b>23.517%±0.287%</b>

Table 10: Results of different domains over WD using 26-key IME abbreviated pinyin mode.

contains *i* tokens, and Sample means that the number of tokens in the context in this data set is 0, and all tokens are in the target. CV-IME-*i* means the results of CV-IME using *i*-th hybrid latent variable. Base+Beam represents the results of base model with beam search. In the tables presented, we observe that our model outperforms across nearly all configurations of context-target lengths.

## D.3 WD Benchmark

We conducted experiments on the WD dataset across different domains and with various contexttarget length configurations.

Table 9, Table 10, Table 11 and Table 12 report the results of different domains over WD. From these tables, it can be observed that: (1) Our CV-

<i>Model</i>	Sports	Journey	Games	Culture
Base+Beam2	59.927%±0.000%	56.542%±0.000%	57.976%±0.000%	52.356%±0.000%
CV-IME	62.122%±0.065%	61.025%±0.065%	<b>60.317%±0.175%</b>	<b>57.034%±0.088%</b>
<i>Model</i>	Military	Real Estate	Technology	Finance
Base+Beam2	54.299%±0.000%	60.940%±0.000%	61.907%±0.000%	62.990%±0.000%
CV-IME	<b>56.609%±0.130%</b>	65.762%±0.065%	<b>64.947%±0.089%</b>	<b>65.656%±0.043%</b>
<i>Model</i>	Education	Economy	Entertainment	International
Base+Beam2	63.874%±0.000%	63.764%±0.000%	60.545%±0.000%	60.177%±0.000%
CV-IME	67.138%±0.193%	<b>66.269%±0.129%</b>	62.483%±0.198%	<b>63.517%±0.107%</b>
<i>Model</i>	Medical	Automobile	Agriculture	Society
Base+Beam2	67.471%±0.000%	59.634%±0.000%	57.018%±0.000%	59.032%±0.000%
CV-IME	<b>70.497%±0.089%</b>	63.708%±0.259%	61.937%±0.135%	<b>63.287%±0.064%</b>

Table 11: Results of different domains over WD using 9-key IME perfect pinyin mode.

<i>Model</i>	Sports	Journey	Games	Culture
Base+Beam2	8.600%±0.000%	7.300%±0.000%	6.600%±0.000%	6.750%±0.000%
CV-IME	9.633%±0.047%	<b>9.633%±0.103%</b>	<b>8.633%±0.094%</b>	<b>8.400%±0.147%</b>
<i>Model</i>	Military	Real Estate	Technology	Finance
Base+Beam2	7.000%±0.000%	8.600%±0.000%	8.200%±0.000%	9.650%±0.000%
CV-IME	<b>8.183%±0.085%</b>	<b>12.150%±0.147%</b>	<b>10.533%±0.024%</b>	<b>11.417%±0.125%</b>
<i>Model</i>	Education	Economy	Entertainment	International
Base+Beam2	8.650%±0.000%	9.050%±0.000%	7.550%±0.000%	7.850%±0.000%
CV-IME	<b>10.517%±0.062%</b>	<b>11.700%±0.122%</b>	<b>9.267%±0.165%</b>	<b>11.400%±0.082%</b>
<i>Model</i>	Medical	Automobile	Agriculture	Society
Base+Beam2	11.700%±0.000%	9.250%±0.000%	8.600%±0.000%	8.650%±0.000%
CV-IME	<b>13.933%±0.170%</b>	<b>10.333%±0.155%</b>	<b>9.350%±0.071%</b>	<b>9.883%±0.094%</b>

Table 12: Results of different domains over WD using 9-key IME abbreviated pinyin mode.

IME model consistently outperforms the baseline model across various domains. (2) The improvement ratio varies across different domains, ranging from a minimum of 0.75 points to a maximum of 4.97 points. We hypothesize that this variability may be attributed to the fact that the training data is extracted from news data, which differs in domain information from the various domains in WD.

Table 13 presents the results of different contexttarget length configurations over WD. The data from the tables indicate the following observations: (1) CV-IME model achieves superior performance over the baseline in most configurations; (2) As the length of the target increases, the difficulty of achieving an exact match between the generated results and the ground truth progressively rises; (3) Extending the length of the context portion effectively enhances the accuracy of the generated outcomes; (4) CV-IME model exhibits improved performance in the target length phases of 0-3 and 4-9, yet its performance diminishes in scenarios where the target length exceeds 10. This may be attributed to the diversity introduced by latent variables, which leads to a discrepancy between the generated content and the ground-truth.

#### D.4 Case Study

Table 14 presents examples of perfect Pinyin mode in 9-key IME. From the table, it can be observed that our CV-IME can recall more diverse and accurate results within similar generation time constraints. However, since CV-IME utilizes unsupervised training of latent variables, the 4 generated results from CV-IME only represent the corresponding mixed latent variables and do not imply the priority of the results. Thus, identifying a timeefficient sorting method is our future research.

26-key Perfect	model	0-3	4-9	10+
0-3	Base+Beam2	77.082%±0.00%	61.814%±0.00%	<b>35.983%±0.00%</b>
	CV-IME	<b>79.354%±0.19%</b>	62.709%±0.20%	35.556%±0.61%
4-9	Base+Beam2	80.541%±0.00%	65.011%±0.00%	35.520%±0.00%
	CV-IME	84.136%±0.16%	67.356%±0.17%	37.841%±0.63%
10+	Base+Beam2	82.148%±0.00%	68.041%±0.00%	38.764%±0.00%
	CV-IME	85.800%±0.11%	69.952%±0.16%	<b>39.700%±0.73%</b>
26-key Abbreviated	model	0-3	4-9	10+
0-3	Base+Beam2	19.744%±0.00%	4.497%±0.00%	<b>0.363%±0.00%</b>
	CV-IME	21.653%±0.20%	<b>4.507%±0.11%</b>	0.308%±0.03%
4-9	Base+Beam2	26.144% ±0.00%	6.188%±0.00%	<b>0.457%±0.00%</b>
	CV-IME	30.340% ±0.22%	7.105%±0.12%	0.451%±0.02%
10+	Base+Beam2	28.278%±0.00%	6.897%±0.00%	0.463%±0.00%
	CV-IME	32.977%±0.17%	<b>8.064%±0.10%</b>	0.487%±0.02%
9-key Perfect	model	0-3	4-9	10+
9-key Perfect 0-3	model Base+Beam2 CV-IME	0-3 60.258%±0.00% <b>62.053</b> %± <b>0.19</b> %	4-9 <b>39.708%±0.00%</b> 39.197%±0.23%	10+ <b>15.755%±0.00%</b> 14.759%±0.39%
	Base+Beam2	60.258%±0.00%	<b>39.708</b> %±0.00%	15.755%±0.00%
0-3	Base+Beam2	60.258%±0.00%	<b>39.708%±0.00%</b>	<b>15.755%±0.00%</b>
	CV-IME	62.053%±0.19%	39.197%±0.23%	14.759%±0.39%
	Base+Beam2	64.176%±0.00%	42.432%±0.00%	15.764%±0.00%
0-3	Base+Beam2	60.258%±0.00%	<b>39.708%±0.00%</b>	<b>15.755% ±0.00%</b>
	CV-IME	62.053%±0.19%	39.197%±0.23%	14.759% ±0.39%
	Base+Beam2	64.176%±0.00%	42.432%±0.00%	15.764% ±0.00%
	CV-IME	69.329%±0.14%	<b>44.543%±0.13%</b>	<b>16.488% ±0.45%</b>
	Base+Beam2	67.647%±0.00%	45.197%±0.00%	16.923% ±0.00%
0-3 4-9 10+	Base+Beam2 CV-IME Base+Beam2 CV-IME Base+Beam2 CV-IME	60.258%±0.00% 62.053%±0.19% 64.176%±0.00% 69.329%±0.14% 67.647%±0.00% 71.851%±0.18%	<b>39.708%±0.00%</b> 39.197%±0.23% 42.432%±0.00% <b>44.543%±0.13%</b> 45.197%±0.00% <b>47.567%±0.16%</b>	<b>15.755% ±0.00%</b> 14.759% ±0.39% 15.764% ±0.00% <b>16.488% ±0.45%</b> 16.923% ±0.00% <b>17.588% ±0.49%</b>
0-3 4-9 10+ 9-key Abbreviated	Base+Beam2 CV-IME Base+Beam2 CV-IME Base+Beam2 CV-IME model Base+Beam2	60.258%±0.00% 62.053%±0.19% 64.176%±0.00% 69.329%±0.14% 67.647%±0.00% 71.851%±0.18% 0-3 7.463%±0.00%	39.708%±0.00% 39.197%±0.23% 42.432%±0.00% 44.543%±0.13% 45.197%±0.00% 47.567%±0.16% 4-9 0.606%±0.00%	15.755%±0.00%         14.759%±0.39%         15.764%±0.00%         16.488%±0.45%         16.923%±0.00%         17.588%±0.49%         10+         0.000%±0.00%

Table 13: Results of different context-target length configuration over WD. Each score is averaged over all domains.

id	Case		Predictions		
1	Context Pinyin Abbreviated Target Translation	经常有这样的 8432 No 提法 There is often such a statement	Base+Beam 1. 体罚 (physical punishment) 2. 提法 (statement)	<ul> <li>CV-IME</li> <li>1. 同行对比(peer comparison)</li> <li>2. 提法(statement)</li> <li>3. 体罚(physical punishment)</li> <li>4. 体会答案</li> <li>(experience solution)</li> </ul>	
2	Context Pinyin Abbreviated Target Translation	- 94 No 以 with	Base+Beam 1. 恣(wantonly) 2. 一(one)	CV-IME 1. 中国(China) 2. 优惠(discount) 3. 以(with) 4. 香菇(mushroom)	
3	Context Pinyin Abbreviated Target Translation	- 9824364 No 组成 composition	Base+Beam 1. 组成(composition) 2. 无成(no success)	CV-IME 1. 五成(fifty percent) 2. 组成(composition) 3. 禹城(Yucheng) 4. 吴城(Wucheng)	

Table 14: Case study.