Advancing Sequential Numerical Prediction in Autoregressive Models

Xiang Fei[*1] **Jinghui Lu**[*1] **Qi Sun**[*2] **Hao Feng**[†1] **Yanjie Wang**[1]
Wei Shi[1] **An-lan Wang**[1] **Jingqun Tang**[1] **Can Huang**[†1]
[1]ByteDance Inc. [2]City University of Hong Kong
`{feixiang.77, lujinghui, fenghao.2019}@bytedance.com`
`{wangyanjie.prince, shiwei.11, wanganlan}@bytedance.com`
`{tangjingqun, can.huang}@bytedance.com`
`qisun.new@gmail.com`

Abstract

Autoregressive models have become the de facto choice for sequence generation tasks, but standard approaches treat digits as independent tokens and apply cross-entropy loss, overlooking the coherent structure of numerical sequences. This paper introduces ***N**umerical **T**oken **I**ntegrity **L**oss (NTIL)* to address this gap. NTIL operates at two levels: (1) token-level, where it extends the Earth Mover's Distance (EMD) to preserve ordinal relationships between numerical values, and (2) sequence-level, where it penalizes the overall discrepancy between the predicted and actual sequences. This dual approach improves numerical prediction and integrates effectively with LLMs/MLLMs. Extensive experiments show significant performance improvements with NTIL. All resources are available at `https://github.com/xfey/NTIL`.

1 Introduction

In recent years, sequence generation has become a crucial approach for implementing a broad range of AI applications, including visual question answering (Wang et al., 2024d; Reich and Schultz, 2024; Fan et al., 2024; Liu et al., 2024b; He et al., 2025; Wang et al., 2025; Lu et al., 2024b), key information extraction (Kim et al., 2024; Yu et al., 2024; Kang et al., 2024; Wang et al., 2024a; Yi et al., 2025; Lu et al., 2023a, 2024a), object detection (Wen et al., 2024), math reasoning (Zhao et al., 2024), text spotting (Li et al., 2024; Yu et al., 2025), and automatic audio recognition (Zhou et al., 2024).

Autoregressive models, especially large language models (LLMs) such as GPT (Achiam et al., 2023), LLaMA (Touvron et al., 2023; Dubey et al., 2024), Qwen (Yang et al., 2024; Wang et al., 2024c) series, with multi-modal large language models (MLLMs) based on them, now dominate the sequence generation tasks. During training, these models generate sequences token-by-token, typically using cross-entropy (CE) loss, to minimize the negative log-likelihood of the ground truth token at each time step. However, CE loss has several inherent limitations when predicting numerical values. Specifically, CE suffers from **Limitation 1** that *it ignores the inherent closeness between numerical tokens, where each digit in a numerical prediction is not independent but related to its neighboring digits.* For example, in Figure 2(a) and 2(b), for the ground truth token "3", the CE loss yields same values of $-log(0.5)$ for different prediction distributions. However, the distribution in Figure 2(b) is more accurate, as it assigns higher probability to the neighboring token "2".

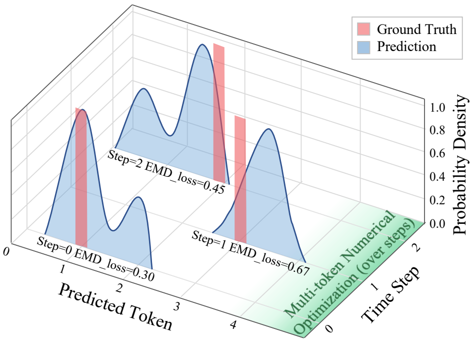


Figure 1: Sequence-level digit token loss illustration.

CE also suffers from **Limitation 2** that *it fails to capture the holistic numerical error when sequential tokens are involved, as it focuses on the precision of each token rather than the overall value.* In an autoregressive generation manner, producing a numerical value typically requires consecutive time steps. For example, the target value *"0.98"* requires the prediction of four sequential tokens — *"0"*, *"."*, *"9"*, *"8"*. Thus, a prediction such as *1.01* (*"1"*,*"."*,*"0"*,*"1"*) incurs a high CE loss as the first, third and fourth tokens are significantly differ-

[*] Equal Contribution
[†] Corresponding Author

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 562–574
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

ent from the target tokens. Conversely, a prediction like *1.98* (*"1"*, *"."*, *"9"*, *"8"*) could yield a lower CE loss due to a closer match at the token level, despite the overall numerical difference being larger (1.00 vs. 0.03). This discrepancy shows the limitation of CE in evaluating predictions holistically.

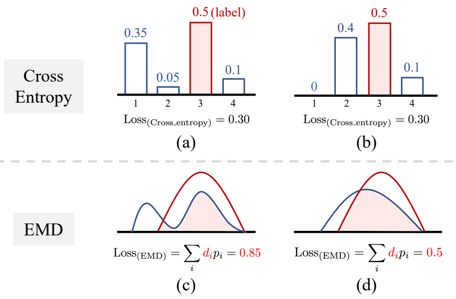

Cross Entropy
0.35
0.05
0.5 (label)
0.1
1
2
3
4
$\mathrm{Loss}_{(\mathrm{Cross\_entropy})} = 0.30$
(a)
0
0.4
0.5
0.1
1
2
3
4
$\mathrm{Loss}_{(\mathrm{Cross\_entropy})} = 0.30$
(b)
EMD
$\mathrm{Loss}_{(\mathrm{EMD})} = \sum_i d_i p_i = 0.85$
(c)
$\mathrm{Loss}_{(\mathrm{EMD})} = \sum_i d_i p_i = 0.5$
(d)

Figure 2: Cross-entropy fails to distinguish predictions, whereas EMD correlates smaller loss for better predicted distributions.

To overcome the above issues, we introduce a novel sequence-level numerical prediction loss: ***N**umerical **T**oken **I**ntegrity **L**oss (NTIL)*. At the token-level, NTIL replaces the traditional CE loss with Earth Mover's Distance (EMD) (Rubner et al., 1998). Additionally, we enhance the EMD with an *Exponential Position-based Weighting* scheme (Section 3.1), which leverages place-value number systems to better capture the nuanced differences between numerical distributions at each time step. At the sequence-level, NTIL evaluates the overall numerical difference between predicted and actual sequences through *Multi-token Numerical Optimization* (Section 3.2), considering all time steps holistically, as illustrated in Figure 1. It enables NTIL to effectively model the actual value of digit sequences, and capture discrepancies across the consecutive numerical range, moving beyond simple token-by-token comparison.

To the best of the authors' knowledge, it is the first time that EMD is used as an optimization method for autoregressive models. Moreover, our holistic approach is the first of its kind to improve sequential numerical prediction by considering numerical tokens across multiple time steps. Our method can be seamlessly integrated into both LLMs and MLLMs. Experimental results show that NTIL boosts performance in tasks requiring precise numerical outputs, such as object detection, text spotting, and math reasoning (Section 4).

2 Related Work

The Earth Mover's Distance (EMD) measures the minimal cost of transforming one distribution into another, and has become a valuable metric in deep learning applications. Notably, Wasserstein GAN (Arjovsky et al., 2017) uses EMD as its loss function to stabilize training in GANs. Cuturi (2013) and Courty et al. (2016) also adopted EMD for smoothing the training procedure. Despite the success of EMD, it has not been applied to autoregressive models. Most recently, autoregressive models, especially LLMs, have advanced NLP (Radford et al., 2019; Touvron et al., 2023; Lu et al., 2023b; Cui et al., 2025), and multi-modal tasks (Alayrac et al., 2022; Wang et al., 2024c; Feng et al., 2025; Lu et al., 2025). While the tasks mentioned above require high precision in numerical value prediction, none of the previous works have specifically optimized for this criterion. Our work addresses this gap by focusing on advancing the sequential numerical prediction for autoregressive models.

3 Method

This section details the components of the proposed method. Section 3.1 proposes exponential weighted EMD to single digits; Section 3.2 describes how we go through multiple digital tokens to derive a simple yet effective numerical measure.

3.1 Exponential Position-Based Weighting

For token-level prediction, to address **Limitation 1** in Section 1, we replace the conventional CE loss with EMD to account for the ordinal relationship during optimization. The preliminaries for both CE and EMD objectives, and the simplification via numerical prediction, are outlined in Appendix D.

Furthermore, we extend EMD to account for the place-value number systems, where leading digits have greater numerical significance. We implement an exponential weighting scheme to progressively assign weights based on digit positions, to scale their contributions to the loss accordingly:

$$\mathbf{W}_{\mathbf{exp}} = \left[(1+\sigma)^{n-i-1}\right]_{i=0}^{n-1}, \qquad (1)$$

where σ is the exponential increment rate, and n is the length of consecutive digits. This implementation helps the model understand the order relationship between consecutive numbers.

3.2 Multi-Token Numerical Optimization

To overcome **Limitation 2** outlined in Section 1, we propose the following procedure and losses.

Differentiable Numerical Value Construction. In this step, we construct the complete numerical value from consecutive discrete digital tokens. Figure 3 illustrates how we obtain the digit index from the predicted distribution using argmax to derive the integer representation. To maintain differentiability, we employ the Gumbel-softmax approximation with reduced temperature and noise parameters to ensure consistent results. The resulting tensor is element-wise multiplied with positional indices, scaled by the appropriate powers of 10, and aggregated to obtain the final value. For further implementation details, see Appendix C.

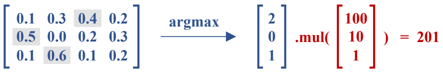


Figure 3: Constructing a numerical value from tokens.

Relative Deviation Metric. For numerical comparison, while absolute difference provides a straightforward measure equivalent to L1 loss, we propose a normalized metric defined as:

$$\mathcal{L}_{relative} = \frac{|X - Y|}{\max(X, Y) + \epsilon}, \tag{2}$$

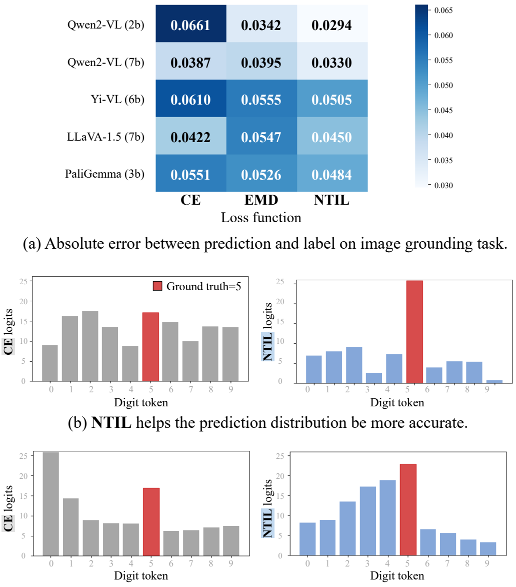
where X is the sequence-level numerical prediction (*e.g.*, “234”) and Y is the ground truth, and ϵ is a small quantity to avoid division by zero. This normalization ensures consistency across different magnitude ranges.

Magnitude Deviation Metric We also apply a normalized metric on the order of magnitude as:

$$\mathcal{L}_{magnitude} = \log\left(\frac{\max(X, Y)}{\min(X, Y)}\right). \tag{3}$$

The objective penalizes the difference in the order of magnitude between two values. For example, given the pairs $(1, 10)$ and $(1, 100)$, which have similar $\mathcal{L}_{relatvie}$ values 0.90 and 0.99, but differ in $\mathcal{L}_{magnitude}$ value: $\log\left(\frac{10}{1}\right) \approx 2.30$ for the first pair and $\log\left(\frac{100}{1}\right) \approx 4.61$ for the second. This results in a larger penalty for greater differences in magnitude. The final formulation of NTIL combines the above loss functions, with tunable hyperparameters to weight their individual contributions.

$$\mathcal{L} = \mathbf{W_{exp}}\,\text{EMD} + \alpha \cdot \mathcal{L}_{relative} + \beta \cdot \mathcal{L}_{magnitude} \tag{4}$$



(a) Absolute error between prediction and label on image grounding task.

(b) **NTIL** helps the prediction distribution be more accurate.

(c) **NTIL** helps the prediction distribution concentrate around ground truth.

Figure 4: Results for quantitative analysis.

4 Experiments and Results

This section presents a comprehensive empirical evaluation of the proposed NTIL across various LLMs/MLLMs (Section 4.1). **CE** (Shannon, 1948) and **EMD** (Rubner et al., 1998) are chosen as baselines due to their widespread adoption. The evaluation encompasses multiple task domains that focus on numerical prediction including *Image Grounding*, *Scene Text Detection*, *Clock Time Recognition*, *Mathematical Reasoning* and *Arithmetic Calculations*. Appendix B provides details on tasks, datasets, and evaluation metrics. We also conduct systematic ablation studies to evaluate the critical components of our approach. Implementation details are available in Appendix A.

4.1 Main Results

4.2 Results of MLLMs

Image Grounding As shown in Table 1, our method outperforms both CE and EMD across nearly all datasets and VLM backbones, as evidenced by the overall performance improvements.

Scene Text Detection Table 2 shows that our method improves accuracy across multiple datasets, demonstrating its effectiveness in predicting multiple object coordinates.

Clock Time Recognition Table 4 demonstrates that NTIL surpasses CE and EMD significantly in performance across all model architectures.

Mathematical Reasoning As shown in Table 5, our method outperforms CE and EMD across all

Model	Method	RefCOCO			RefCOCO+			RefCOCOg		Avg
		Val	TestA	TestB	Val	TestA	TestB	Val	Test	
PaliGemma (3b) (Beyer et al., 2024)	CE	0.839	0.865	0.784	0.740	0.797	0.664	0.792	0.797	0.785
	EMD	0.841	0.864	**0.796**	0.749	0.805	0.669	0.789	0.799	0.789
	Ours	**0.844**	**0.873**	0.791	**0.750**	**0.812**	**0.678**	**0.804**	**0.802**	**0.795**
LLaVA-1.5 (7b) (Liu et al., 2024a)	CE	0.855	0.880	0.813	**0.801**	0.843	0.741	0.799	0.816	0.818
	EMD	0.856	0.879	**0.822**	0.798	0.845	0.743	0.798	0.816	0.820
	Ours	**0.858**	**0.885**	0.815	0.800	**0.853**	**0.747**	**0.802**	**0.817**	**0.822**
Yi-VL (6b) (Young et al., 2024)	CE	0.767	0.796	0.734	0.706	0.757	0.651	0.722	0.731	0.733
	EMD	**0.779**	0.805	0.738	**0.719**	0.762	0.657	0.721	0.737	0.740
	Ours	0.777	**0.808**	**0.741**	0.717	**0.770**	**0.665**	**0.727**	**0.743**	**0.744**
Qwen2-VL (2b) (Wang et al., 2024c)	CE	0.897	0.928	**0.850**	0.841	**0.896**	0.776	0.851	**0.867**	0.863
	EMD	0.889	0.931	0.843	0.838	0.889	0.772	0.853	0.858	0.859
	Ours	**0.898**	**0.932**	0.849	**0.844**	0.891	**0.788**	**0.858**	0.863	**0.866**
Qwen2-VL (7b) (Wang et al., 2024c)	CE	**0.892**	0.929	0.841	0.842	0.902	0.784	0.843	0.848	0.860
	EMD	0.886	0.926	0.834	0.843	0.901	0.768	0.836	0.843	0.855
	Ours	0.889	**0.931**	**0.840**	**0.844**	**0.904**	**0.786**	**0.848**	**0.853**	**0.862**

Table 1: Performance comparison (Acc@0.5) of models on image grounding tasks.

Model	Method	Dataset				Avg
		CTW1500	ICDAR1500	TD500	Total-Text	
PaliGemma (3b)	CE	0.220	0.129	0.183	0.259	0.193
	EMD	0.314	0.124	0.252	0.307	0.241
	Ours	**0.369**	**0.155**	**0.257**	**0.318**	**0.263**
Yi-VL (6b)	CE	**0.682**	0.370	0.753	0.673	0.586
	EMD	0.668	0.398	**0.778**	0.678	0.594
	Ours	0.680	**0.403**	0.752	**0.678**	**0.597**
Qwen2-VL (2b)	CE	0.786	0.538	0.851	0.827	0.720
	EMD	**0.786**	0.535	**0.867**	0.808	0.718
	Ours	0.776	**0.577**	0.854	**0.835**	**0.732**
Qwen2-VL (7b)	CE	**0.771**	0.648	**0.889**	0.864	0.764
	EMD	0.762	0.625	0.874	0.860	0.751
	Ours	0.770	**0.669**	0.869	**0.872**	**0.770**
LLaVA-1.5 (7b)	CE	0.735	0.490	0.821	0.786	0.675
	EMD	0.724	0.545	**0.840**	0.776	0.690
	Ours	**0.739**	**0.547**	0.839	**0.791**	**0.698**

Table 2: Performance (Acc@0.5) on scene text detection tasks.

Model	Accuracy (%)		
	CE	EMD	**Ours**
Baichuan2 (7b) (Yang et al., 2023)	44.3	46.6	**46.9**
Qwen2.5 (1.5b) (Team, 2024)	40.3	40.7	**42.4**
LLaMA3 (8b) (Dubey et al., 2024)	61.9	61.8	**61.9**
Yi (6b) (Young et al., 2024)	53.0	**54.6**	54.4
MiniCPM3 (4b) (Hu et al., 2024)	66.8	68.2	**68.6**

Table 3: Performance comparison of accuracies on the arithmetic calculation task.

Metric	Method	LLaVA-1.5 (7b)	Qwen2-VL (7b)	Qwen2-VL (2b)	Yi-VL (6b)
Accuracy (%) ↑	CE	95.1	75.0	81.3	76.2
	EMD	95.3	78.7	81.7	75.1
	Ours	**98.3**	**80.5**	**85.3**	**87.4**
Time gap (minute) ↓	CE	8.52	30.84	32.34	56.58
	EMD	7.98	30.78	31.98	54.78
	Ours	**4.14**	**27.72**	**24.66**	**26.58**

Table 4: Performance of the clock time recognition task.

			PaliGemma		**LLaVA-1.5**		**Qwen2-VL**	**Yi-VL**
Exp	Rel	Mag	Mathvision	Mathvista	Mathvision	Mathvista	Clock_Time	Clock_Time
×	✓	✓	0.096	0.137	0.151	0.166	0.798	0.834
✓	×	✓	0.095	0.137	0.145	0.154	0.790	0.856
✓	✓	×	0.094	0.142	**0.160**	0.143	0.816	**0.876**
✓	✓	✓	**0.098**	**0.157**	0.146	**0.170**	**0.853**	0.874

Table 6: Ablations on NTIL. Exp: Exponential Position-Based Weighting. REL: Relative Deviation Metric. Mag: Magnitude Deviation Metric.

Dataset	Method	Qwen2-vl (2b)	Qwen2-vl (7b)	LLaVA-1.5 (7b)	Yi-VL (6b)	PaliGemma (3b)
Mathvision	CE	0.139	0.184	0.146	0.143	0.097
	EMD	0.130	0.188	**0.148**	0.142	0.088
	Ours	**0.145**	**0.191**	0.146	**0.153**	**0.098**
Mathvista	CE	0.248	**0.315**	0.140	0.187	0.143
	EMD	**0.262**	0.303	0.157	0.192	0.149
	Ours	0.251	0.300	**0.170**	**0.222**	**0.157**

Table 5: Performance of the math reasoning task.

datasets, with the most significant improvements seen in the Mathvision dataset using the Qwen2-VL (2b) and in Mathvista with the Yi-VL (6b).

4.3 Results of LLMs

Arithmetic Calculation As shown in Table 3, our method improves accuracy across multiple LLMs, though LLaMA3 shows minimal gains, possibly due to its extensive pre-training. Overall, the majority of cases show that for numerical predictions, while EMD performs comparably or marginally better than CE loss, NTIL consistently delivers superior results in most scenarios. This underscores the effectiveness and generalizability of NTIL.

4.4 Ablation Analysis

Table 6 indicates that incorporating all components of NTIL generally leads to better performance, as evidenced by the highest scores in most metrics when all components are enabled. As an exception, the inclusion of Magnitude leads to worse results in Mathvision for LLaVA-1.5, which indicates the fluctuation of applying Magnitude in some cases.

4.5 Quantitative Analysis

As shown in Figure 4(a), NTIL achieves the lowest absolute errors among all models, indicating more consistent performance compared to CE and EMD. Figure 4(b) and 4(c) illustrate that NTIL produces more accurate predictions with distributions more concentrated around the ground truth. Overall, NITL offers more stability and lower variability. Qualitative examples can be seen in Appendix F.

5 Conclusion

We propose NTIL, which improves numerical prediction accuracy in LLMs at both the token and sequence levels. Experiments show improvement across multiple datasets and models, highlighting effectiveness of NTIL.

Limitations

The limitations of the NTIL include the exponential position-based weighting scheme, while effective in many cases, had limited or negative impact in certain configurations, such as with the Mathvision dataset and LLaVA-1.5 model. Future exploration could focus on refining the exponential position-based weighting scheme with adaptive strategies to address its inconsistent impact.

Furthermore, NTIL introduces additional computational overhead compared to CE loss, resulting in reduced training efficiency. This trade-off between performance improvement and computational cost needs to be considered in practical applications.

Another limitation arises from the tokenization strategies of certain models like LLaMA-3, which encode common multi-digit numbers (e.g., "123") as a single token. Such cases require special handling in NTIL's implementation, adding complexity to the framework.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Martin Arjovsky, S Chintala, and Léon Bottou. 2017. Wasserstein gan. arxiv preprint arxiv: 170107875. *arXiv preprint arXiv:1701.07875*.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. 2024. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*.

Chee Kheng Ch'ng and Chee Seng Chan. 2017. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 935–942. IEEE.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. 2016. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865.

Xiao Cui, Mo Zhu, Yulei Qin, Liang Xie, Wengang Zhou, and Houqiang Li. 2025. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23724–23732.

Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yue Fan, Jing Gu, Kaiwen Zhou, Qianqi Yan, Shan Jiang, Ching-Chen Kuo, Yang Zhao, Xinze Guan, and Xin Wang. 2024. Muffin or Chihuahua? challenging multimodal large language models with multipanel VQA. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6845–6863, Bangkok, Thailand. Association for Computational Linguistics.

Hao Feng, Shu Wei, Xiang Fei, Wei Shi, Yingdong Han, Lei Liao, Jinghui Lu, Binghong Wu, Qi Liu, Chunhui Lin, et al. 2025. Dolphin: Document image parsing via heterogeneous anchor prompting. *arXiv preprint arXiv:2505.14059*.

gpiosenka. 2022. Time-image dataset-classification.

Yangfan He, Jianhui Wang, Kun Li, Yijin Wang, Li Sun, Jun Yin, Miao Zhang, and Xueqian Wang. 2025. Enhancing intent understanding for ambiguous prompts through human-machine co-adaptation. *arXiv preprint arXiv:2501.15167*.

Le Hou, Chen-Ping Yu, and Dimitris Samaras. 2016. Squared earth mover's distance-based loss for training deep neural networks. *arXiv preprint arXiv:1611.05916*.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Iris AM Huijben, Wouter Kool, Max B Paulus, and Ruud JG Van Sloun. 2022. A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(2):1353–1371.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Hyeonseok Kang, Hyein Seo, Jeesu Jung, Sangkeun Jung, Du-Seong Chang, and Riwoo Chung. 2024. Guidance-based prompt data augmentation in specialized domains for named entity recognition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 665–672, Bangkok, Thailand. Association for Computational Linguistics.

Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. 2015. Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE.

Seoyeon Kim, Kwangwook Seo, Hyungjoo Chae, Jinyoung Yeo, and Dongha Lee. 2024. VerifiNER: Verification-augmented NER via knowledge-grounded reasoning with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2441–2461, Bangkok, Thailand. Association for Computational Linguistics.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Wenhao Liu, Xiaohua Wang, Muling Wu, Tianlong Li, Changze Lv, Zixuan Ling, Zhu JianHao, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2024b. Aligning large language models with human preferences through representation engineering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10619–10638, Bangkok, Thailand. Association for Computational Linguistics.

Jinghui Lu, Yanjie Wang, Ziwei Yang, Xuejing Liu, Brian Mac Namee, and Can Huang. 2024a. Padellmner: parallel decoding in large language models for named entity recognition. *Advances in Neural Information Processing Systems*, 37:117853–117880.

Jinghui Lu, Haiyang Yu, Yanjie Wang, Yongjie Ye, Jingqun Tang, Ziwei Yang, Binghong Wu, Qi Liu, Hao Feng, Han Wang, et al. 2024b. A bounding box is worth one token: Interleaving layout and text in a large language model for document understanding. *arXiv preprint arXiv:2407.01976*.

Jinghui Lu, Haiyang Yu, Siliang Xu, Shiwei Ran, Guozhi Tang, Siqi Wang, Bin Shan, Teng Fu, Hao Feng, Jingqun Tang, et al. 2025. Prolonged reasoning is not all you need: Certainty-based adaptive routing for efficient llm/mllm reasoning. *arXiv preprint arXiv:2505.15154*.

Jinghui Lu, Rui Zhao, Brian Mac Namee, and Fei Tan. 2023a. Punifiedner: A prompting-based unified ner system for diverse datasets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 13327–13335.

Jinghui Lu, Dongsheng Zhu, Weidong Han, Rui Zhao, Brian Mac Namee, and Fei Tan. 2023b. What makes pre-trained language models better zero-shot learners? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2288–2303, Toronto, Canada. Association for Computational Linguistics.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023c. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Daniel Reich and Tanja Schultz. 2024. Uncovering the full potential of visual grounding methods in VQA. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4406–4419, Bangkok, Thailand. Association for Computational Linguistics.

Y. Rubner, C. Tomasi, and L.J. Guibas. 1998. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 59–66.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuandong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, et al. 2020. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12965–12974.

Han Wang, Yongjie Ye, Bingru Li, Yuxiang Nie, Jinghui Lu, Jingqun Tang, Yanjie Wang, and Can Huang. 2025. Vision as lora. *arXiv preprint arXiv:2503.20680*.

Huiming Wang, Liying Cheng, Wenxuan Zhang, De Wen Soh, and Lidong Bing. 2024a. Order-agnostic data augmentation for few-shot named entity recognition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7792–7807, Bangkok, Thailand. Association for Computational Linguistics.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. 2024b. Measuring multimodal mathematical reasoning with math-vision dataset. *arXiv preprint arXiv:2402.14804*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024c. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Qunbo Wang, Ruyi Ji, Tianhao Peng, Wenjun Wu, Zechao Li, and Jing Liu. 2024d. Soft knowledge prompt: Help external knowledge become a better teacher to instruct LLM in knowledge-based VQA. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6132–6143, Bangkok, Thailand. Association for Computational Linguistics.

Haoyang Wen, Eduard Hovy, and Alexander Hauptmann. 2024. Transitive consistency constrained learning for entity-to-entity stance detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1467–1480, Bangkok, Thailand. Association for Computational Linguistics.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. 2012. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1083–1090. IEEE.

Qiang Yi, Yangfan He, Jianhui Wang, Xinyuan Song, Shiyao Qian, Xinhang Yuan, Miao Zhang, Li Sun, Keqin Li, Kuan Lu, et al. 2025. Score: Story coherence and retrieval enhancement for ai narratives. *arXiv preprint arXiv:2503.23512*.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

Haiyang Yu, Jinghui Lu, Yanjie Wang, Yang Li, Han Wang, Can Huang, and Bin Li. 2025. Eve: Towards end-to-end video subtitle extraction with vision-language models. *arXiv preprint arXiv:2503.04058*.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. 2016. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 69–85. Springer.

Zhuohao Yu, Chang Gao, Wenjin Yao, Yidong Wang, Wei Ye, Jindong Wang, Xing Xie, Yue Zhang, and Shikun Zhang. 2024. KIEval: A knowledge-grounded interactive evaluation framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5967–5985, Bangkok, Thailand. Association for Computational Linguistics.

Liu Yuliang, Jin Lianwen, Zhang Shuaitao, and Zhang Sheng. 2017. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*.

Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. 2024.

DocMath-eval: Evaluating math reasoning capabilities of LLMs in understanding long and specialized documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16103–16120, Bangkok, Thailand. Association for Computational Linguistics.

Shilin Zhou, Zhenghua Li, Yu Hong, Min Zhang, Zhefeng Wang, and Baoxing Huai. 2024. CopyNE: Better contextual ASR by copying named entities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2675–2686, Bangkok, Thailand. Association for Computational Linguistics.

A Implementation

The proposed loss function is incorporated into the model's training objective through linear combination with a weighting coefficient $\lambda = 0.3$. The hyperparameters governing the loss computation are maintained at $\alpha = \beta = \sigma = 0.2$ throughout all experiments, unless otherwise specified. All tasks are trained with a learning rate of 10^{-5} for fine-tuning. Our experiments are conducted based on a widely used open source training repository[1].

B Dataset

This section provides a detailed description of each task along with the corresponding evaluation metrics. The illustrations for each task are presented in Figure 5.

Image Grounding. Grounding task aims to output the bounding box of the corresponding object given a description. We compare on the referring expression comprehension (REC) task on RefCOCO (Lin et al., 2014), RefCOCO+ (Yu et al., 2016) and RefCOCOg (Mao et al., 2016) datasets. The Average Accuracy at IoU ≥ 0.5 (Acc@0.5) is used as the evaluation metric.

Scene Text Detection. The scene text detection task focuses on detecting text in natural images. We selected several commonly used datasets: TD500 (Yao et al., 2012), ICDAR2015 (Karatzas et al., 2015), CTW1500 (Yuliang et al., 2017) and Total-Text (Ch'ng and Chan, 2017) for scene text detection tasks. We utilize the identical metric employed in the image grounding task.

Clock Time Recognition. The perception of clock aims to recognize the specific time by images of clocks. We compare the performance of accuracy and time gap on a widely-used TIME (gpiosenka, 2022) dataset. The output are formatted as the label of "2_55", as shown in Figure 6. We use overall accuracy as an metric, and additionally count the time gap between the prediction and the ground truth for further evaluation. For example, the time gap between prediction "4_35" and ground truth "6_20" is 1.75 hours.

Mathematical Reasoning. Completing the mathematical reasoning tasks requires models to understand the context and the image of the mathematical field. We select the MathVista (Lu et al., 2023c) and MathVision (Wang et al., 2024b) datasets to evaluate models. We utilize exact matching accuracy to evaluate math reasoning task.

Arithmetic Calculations. Calculation task involves training LLMs to perform numerical operations accurately. In this task, the "arithmetic_mix" subset from the widely-used mathematics dataset (Saxton et al., 2019) is used for training and evaluation, which contains 2M training and 10k test items. In this task, exact matching accuracy is applied as the evaluation metric.

C Gumbel Softmax

The Gumbel softmax, also known as Concrete Distribution, is a continuous differentiable approximation to categorical sampling. It replaces the non-differentiable argmax operation with a softmax function and Gumbel noise. Given logits π_i, the Gumbel softmax sample y_i is computed as:

$$y_i = \mathrm{softmax}\left((log(\pi_i) + g_i)/\tau\right),$$

where g_i is the Gumbel noise, which is i.i.d. samples drawn from the $\mathrm{Gumbel}(0, 1)$ distribution, and τ is the temperature parameter.

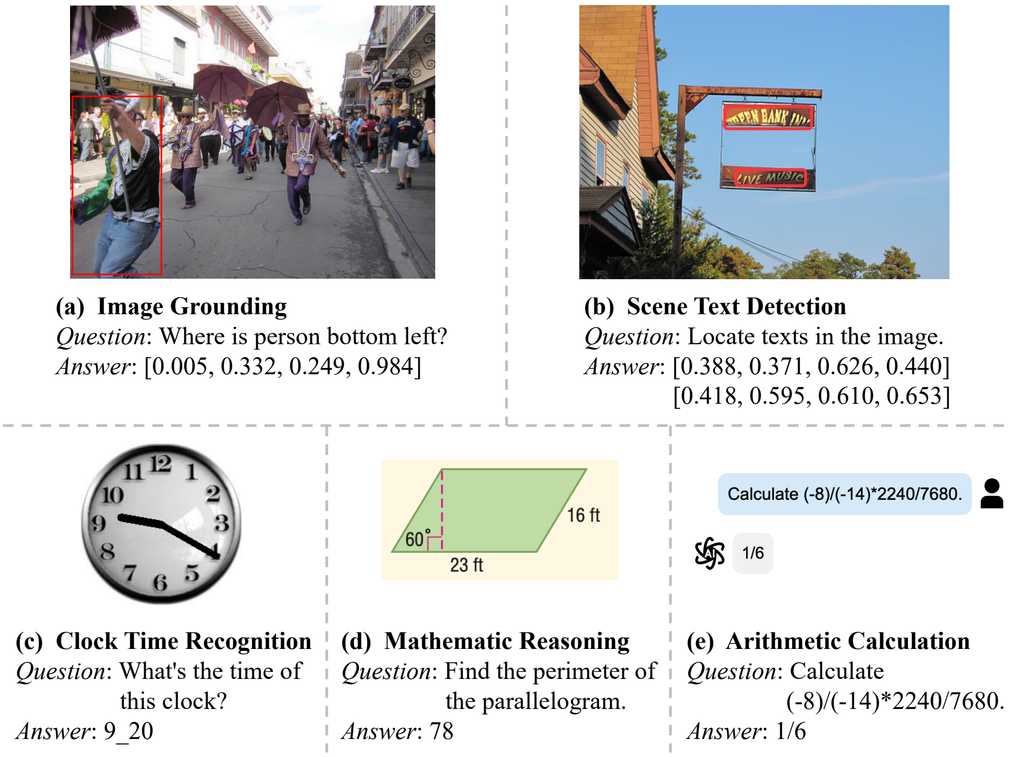
The Gumbel noise term g_i introduces stochasticity into the sampling process, enabling exploration of the probability space while maintaining differentiability. Moreover, using Gumbel noise also works like regularization, which helps provide gradient information near the decision boundary, to improve generalization ability. The temperature parameter τ controls the sharpness of the distribution: as τ approaches 0, the samples become more discrete and closer to one-hot vectors, while higher temperatures make the distribution more uniform. In our implementation, we use $\tau = 0.1$ to ensure that the results are consistent with the original argmax results.

Gumbel softmax is differentiable as it replaces the discrete argmax with a continuous softmax function, allowing gradients to flow through the sampling process during backpropagation. Thus, Gumbel softmax is widely used in scenarios requiring discrete latent variables in neural networks, such as in VAEs(Jang et al., 2016) or reinforcement learning(Huijben et al., 2022; Wan et al., 2020).

D Preliminaries

This section first briefly introduces the autoregressive decoding process based on cross-entropy in Section D.1, and then compares and analyzes Earth Mover's Distance (EMD) in Section D.2.

[1] https://github.com/hiyouga/LLaMA-Factory



(a) Image Grounding
Question: Where is person bottom left?
Answer: [0.005, 0.332, 0.249, 0.984]

(b) Scene Text Detection
Question: Locate texts in the image.
Answer: [0.388, 0.371, 0.626, 0.440]
[0.418, 0.595, 0.610, 0.653]

(c) Clock Time Recognition
Question: What's the time of this clock?
Answer: 9_20

(d) Mathematic Reasoning
Question: Find the perimeter of the parallelogram.
Answer: 78

Calculate (-8)/(-14)*2240/7680.

1/6

(e) Arithmetic Calculation
Question: Calculate (-8)/(-14)*2240/7680.
Answer: 1/6

Figure 5: The illustrations for each task.

D.1 Autoregressive Prediction with Cross Entropy

Autoregressive models operate through sequential decoding, generating tokens one at a time conditioned on previously generated tokens. For each position, the model outputs a probability distribution across the vocabulary, employing the Softmax function to select the most probable token during training.

In the context of language modeling tasks, cross-entropy loss serves as the fundamental training objective for autoregressive models. This loss function quantifies the divergence between the predicted probability distribution and the ground truth distribution:

$$\mathcal{L} = -\sum_i p_i \log(q_i), \tag{5}$$

Åwhere p_i represents the one-hot encoded ground truth distribution, and q_i denotes the model's predicted probability.

While cross-entropy loss effectively minimizes distributional differences between predictions and labels during training, it exhibits a fundamental limitation in autoregressive decoding: the function treats each class independently, disregarding the inherent relationships between different classes. This limitation becomes particularly problematic when modeling numerical sequences where ordinal relationships between values carry semantic significance(Hou et al., 2016), as shown in Figure 2.

D.2 Earth Mover's Distance

To introduce a distance term when calculating the above-mentioned distribution differences, one method is Earth Mover's Distance (EMD), also known as Wasserstein distance. It is an evaluation based on optimal transport theory, measuring the minimal cost of transforming one distribution into the other:

$$\mathrm{EMD}(P, Q) = \min_{\gamma \in \Gamma(P,Q)} \sum_{i=1}^{n} \sum_{j=1}^{m} \gamma_{ij} \cdot d(x_i, y_j), \tag{6}$$

where $P = \{(p_i, x_i)\}$ and $Q = \{(q_i, y_i)\}$ are two discrete distributions, with p_i and q_j are the masses at the points x_i and y_j, respectively. The transport plans, represented as $\Gamma(P, Q)$, are all possible ways to move the mass, and γ_{ij} represents the amount of mass that is transported from p_i to q_j. The distance matrix $d(x_i, y_j)$ indicates the cost of transporting masses between points x_i and y_j. A widely-used distance matrix d is Euclidean distance.

Since the distance between labels is explicitly considered, predicted values closer to the label are associated with smaller distance terms. Thus, the Earth Mover's Distance effectively incorporates distance-based weighting. As illustrated in Figure 2, when the distribution is more concentrated around the label, the EMD loss becomes smaller, thereby reflecting the differences between distributions.

D.3 Predicting Digits with EMD

This section presents our approach to refining distance metrics for numerical representation at the digit level. In traditional autoregressive models, cross-entropy loss is typically employed to predict the probability distributions of individual tokens. However, this method treats each numerical digit as an independent entity, disregarding the continuous relationships between numbers. For example, when the target digit is 4, a model prediction of 3 should ideally be considered closer to accurate than a prediction of 9, as it represents a smaller numerical deviation. To address this limitation, we propose incorporating a distance metric that captures these intrinsic numerical relationships more accurately.

Computational Complexity. As established in D.2, Earth Mover's Distance (EMD) provides a robust measure for distributional distances, making it particularly well-suited for numerical prediction tasks. Prior research has applied EMD to align hidden representations within neural networks, often requiring the transport plan (γ_{ij} in Equ. (6)) to be approximated or recalculated dynamically during training. However, the computational demands of EMD present practical challenges, especially in large-scale deep learning applications. Solving the underlying optimization problem in Equ. (6) has a computational complexity of $O\left((n \times m)^3\right)$, which can be prohibitive. Regularized EMD (Cuturi, 2013) addresses this by employing the Sinkhorn-Knopp algorithm to iteratively refine the transport plan γ_{ij} in Equ. (6), reducing complexity to $O\left(k \times n \times m\right)$, where each iteration involves an $O(n \times m)$ matrix operation.

$\beta \setminus \alpha$	0	0.1	0.2	0.3	0.4
0	0.780	0.783	0.789	0.789	0.771
0.1	0.777	**0.790**	**0.799**	0.789	0.778
0.2	0.782	**0.793**	**0.795**	0.785	0.784
0.3	0.788	0.783	0.786	0.785	0.781
0.4	0.774	0.782	0.778	0.781	0.784

Table 7: Ablation studies of hyper-parameters α and β

	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7
σ	0.784	0.785	**0.789**	**0.788**	0.783	0.780	0.770	0.763
λ	0.741	0.777	**0.798**	**0.795**	0.788	0.786	0.787	0.777

Table 8: Ablation studies of the hyper-parameters σ (exp-weighting) and λ (coefficient)

Numerical Prediction Optimization with EMD. When estimating the transport plan, the algorithm's complexity is generally quadratic. However, when restricted to one-dimensional numerical distributions, where the prediction and target values are aligned in position ($i = j$), the transport plan can be simplified to an identity matrix. Thus, Earth Mover's Distance emerges as a highly suitable metric for capturing digit-level numerical distance, formulated as:

$$\mathrm{EMD}(P, Q) = \sum_{i} |x_i - y_i| \cdot |i - \mathrm{argmax}\,(Q)|\,, \tag{7}$$

where the distance matrix $d\left(x_i, y_j\right) = |i - \mathrm{argmax}\,(Q)|$ refers to the index distance of each digit to the label. Given that the predicted probability distribution P is obtained through the softmax transformation, and the ground truth label Q is represented as a one-hot vector, the gradient of EMD with respect to component x_i can be expressed as:

$$\frac{\partial\,\mathrm{EMD}}{\partial x_i} = \left\{|k-1|\,, |k-2|\,, ..., |k-n|\right\}. \tag{8}$$

where $k = \mathrm{argmax}\,(Q)$ denotes the index of the label element in the one-hot vector. This gradient exhibits an inverse relationship with the proximity between the predicted distribution and the ground truth: as the prediction approaches the true label, the magnitude of the gradient diminishes. This characteristic is particularly advantageous for numerical prediction tasks, as it inherently accounts for the ordinal relationships between numerical classes, and addresses the fundamental limitation of the conventional cross-entropy.

E Ablations

We have supplemented our work with comprehensive experiments on the clock time recognition task using Qwen2-VL-7B, as shown in Table 7 and Table 8. We have bolded some of the outperforming results. The results demonstrate that NTIL adapts well to different hyperparameters.

We have also supplemented NTIL with the GSM8k math reasoning dataset. NTIL brings consistent performance improvement on mixed text-numeric tasks, which shows the effectiveness of our method.

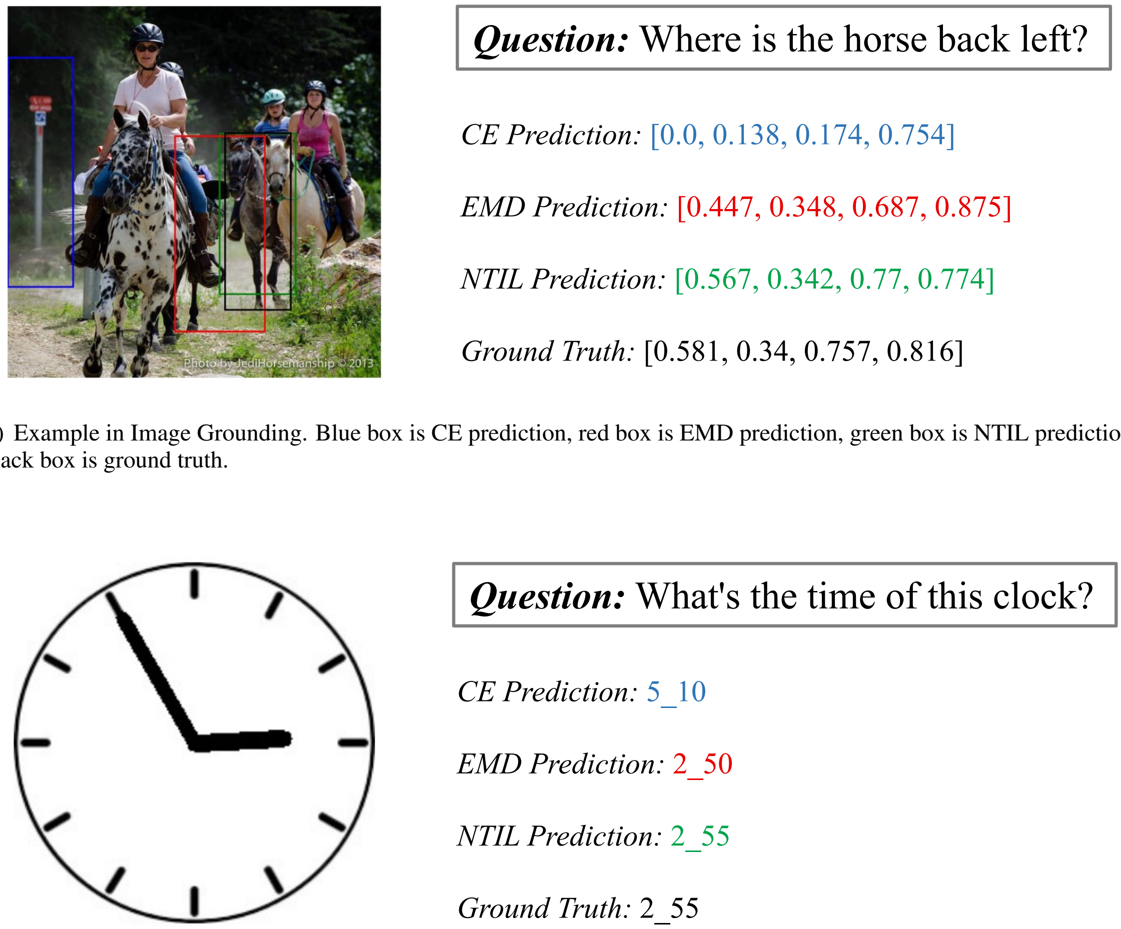
Model	Params	CE	EMD	**NTIL**
Qwen2.5	2b	0.509	0.517	**0.523**
MiniCPM3	4b	0.672	0.676	**0.704**
Yi	6b	0.375	0.383	**0.400**
LLaMA3	8b	0.638	0.643	**0.646**

Table 9: Results of the GSM8k mathematical reasoning task.

F Qualitative Examples

Visualizations of the outputs of different losses are shown in Figure 6, and the examples are taken from experimental results using LLaVA-1.5. For image grounding task (Figure 6(a)), the task was to predict the location of "horse back left" in an image. The CE loss (blue box) performed poorly, with predictions far from the ground truth. EMD (red box) showed an improvement, capturing spatial features better, while NTIL (green box) provided the most accurate predictions, closely matching the ground truth (black box). Overall, NTIL outperformed both CE and EMD, demonstrating its effectiveness in this task.

Figure 6(b) presents a qualitative comparision for clock time recognition task. In this case, NTIL provides the most accurate prediction of the clock time, correctly identifying 2:55, which matches the ground truth. EMD performs better than CE, predicting 2:50, but it is still slightly off. CE, however, predicts 5:10, a significant deviation. Overall, NTIL outperforms both EMD and CE in predicting the clock time accurately.


Question: Where is the horse back left?
CE Prediction: [0.0, 0.138, 0.174, 0.754]
EMD Prediction: [0.447, 0.348, 0.687, 0.875]
NTIL Prediction: [0.567, 0.342, 0.77, 0.774]
Ground Truth: [0.581, 0.34, 0.757, 0.816]
Photo by JediHorsemanship © 2013
Question: What's the time of this clock?
CE Prediction: 5_10
EMD Prediction: 2_50
NTIL Prediction: 2_55
Ground Truth: 2_55

(a) Example in Image Grounding. Blue box is CE prediction, red box is EMD prediction, green box is NTIL prediction. Black box is ground truth.

(b) Example in clock time recognition.

Figure 6: Comparisons between CE, EMD and NTIL.