61 ACL 2023

INK: Injecting kNN Knowledge in Nearest Neighbor Machine Translation

Wenhao Zhu[1], Jingjing Xu[2], Shujian Huang[1], Lingpeng Kong[3], Jiajun Chen[1]

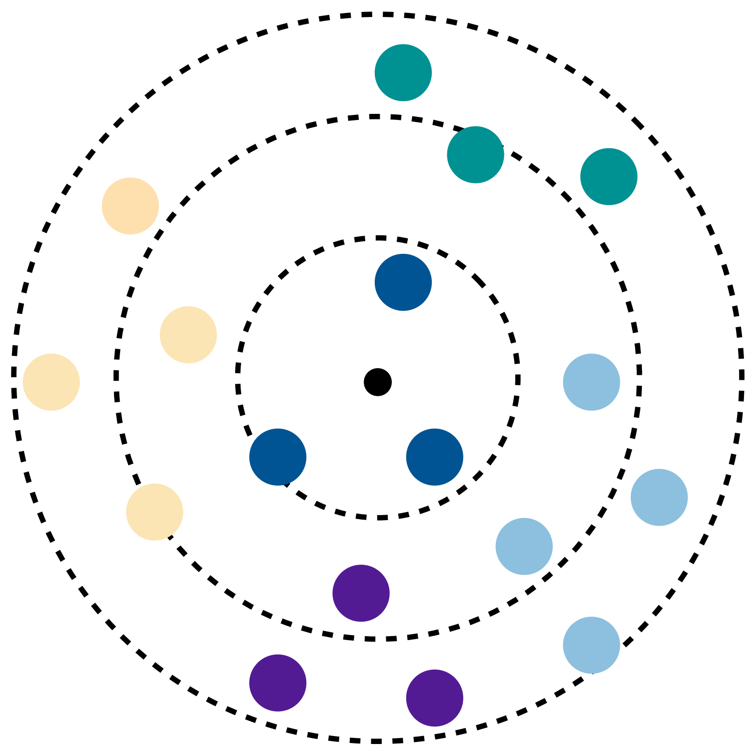
National Key Laboratory for Novel Software Technology, Nanjing University

Shanghai AI Laboratory The University of Hong Kong

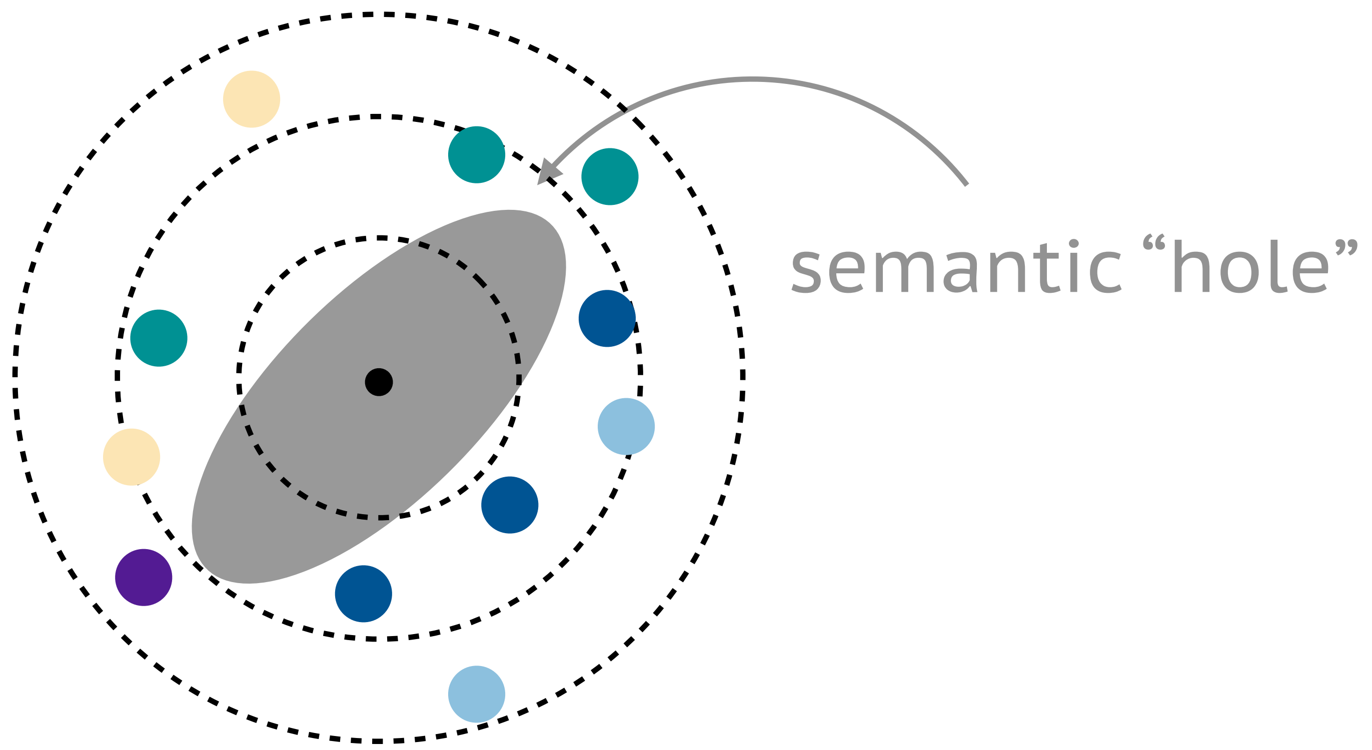
Neural Machine Translation

- The target of NMT is to learn a generalized representation space to adapt to diverse scenarios.
- However, neural networks often induce a non-smooth representation space, limiting its generalization ability.

Ideally, all of the representations in a neighborhood should share the same target token.



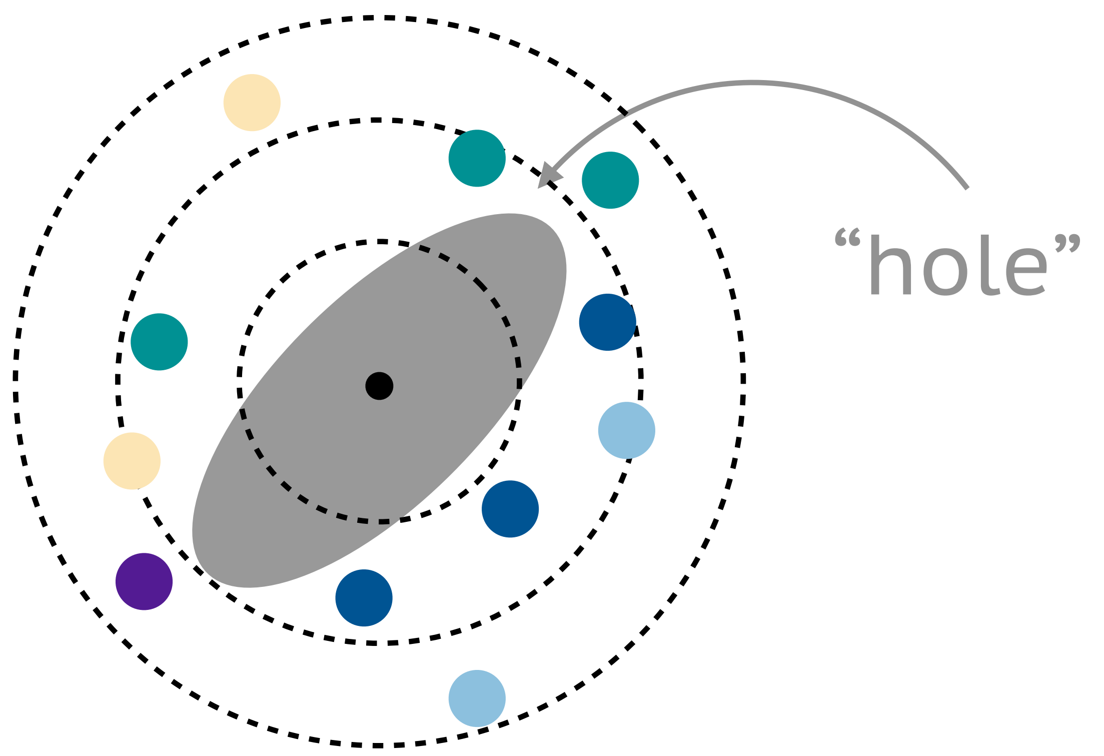
✅smooth space



❌non-smooth space

Non-smooth Representation Space of NMT Model

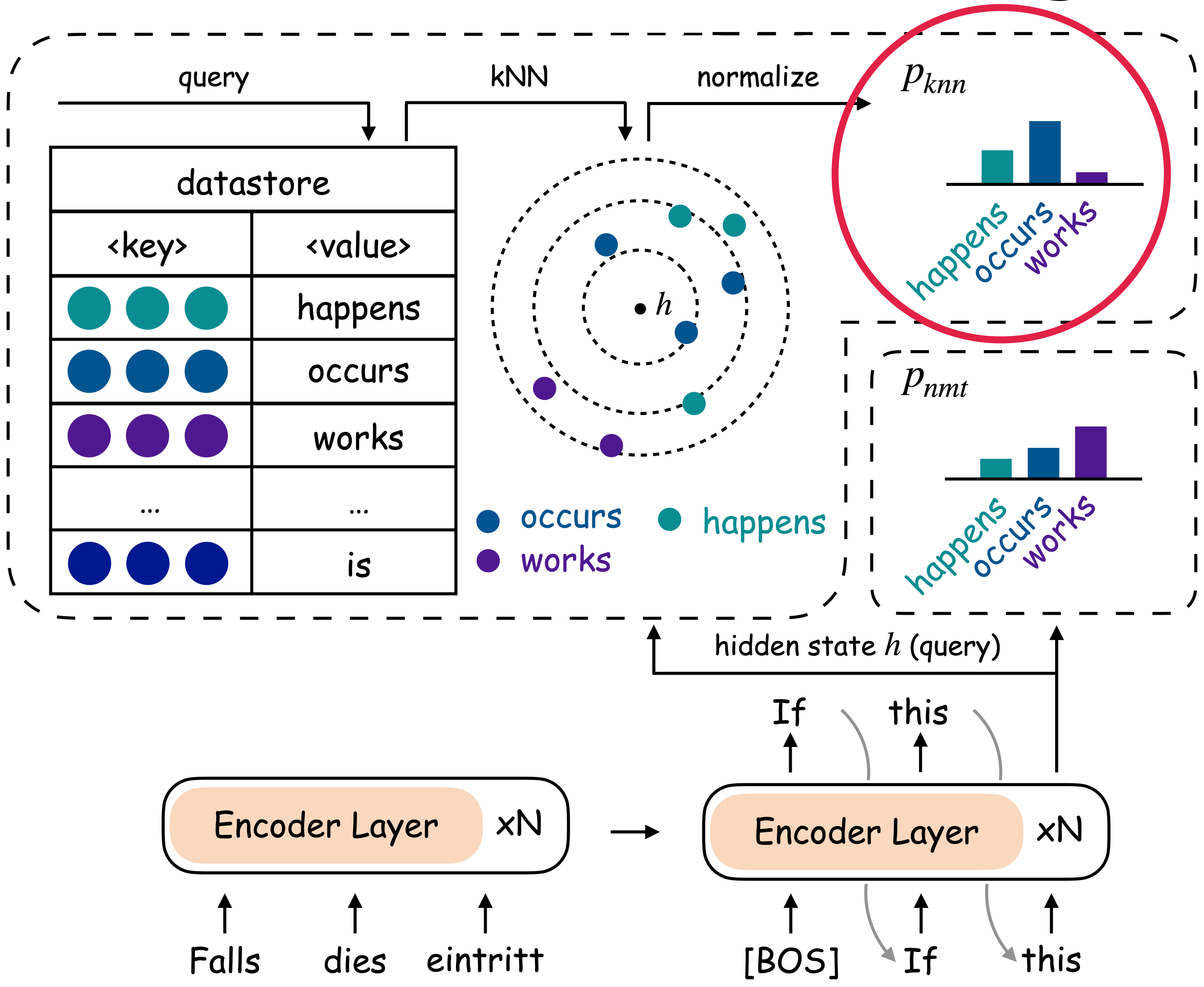
- non-smooth representation space
  - low-frequency tokens disperse sparsely.
  - many “holes” could be formed, where the semantic meaning can be poorly defined.
- As a result, when NMT is used to translate examples from an unseen domain, the performance drops sharply.



❌non-smooth space

Previous Solution: kNN-MT

- kNN-MT (k-nearest neighbor machine translation)
  - saving representations and target tokens into a datastore
  - smoothing predictions with nearest neighbors



Drawbacks of kNN-MT

- Retrieving neighbors from a large datastore at each decoding step is time consuming
- Once the datastore is constructed, representations can not be easily updated

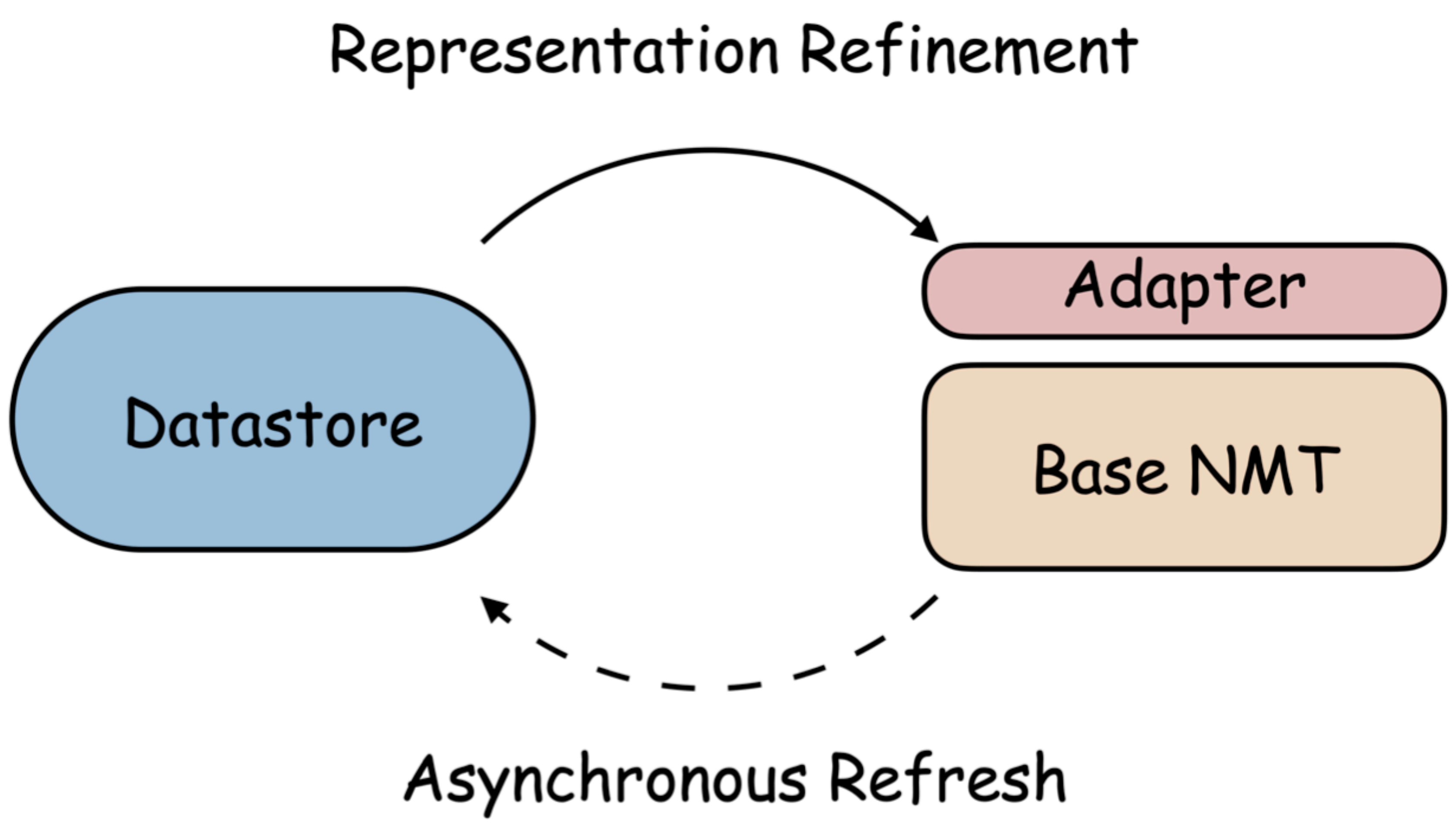
To overcome these drawbacks, we propose
INK to INject kNN Knowledge into NMT.

Smoothing Representation Space with INK

- overview of INK training loop
  - representation refinement

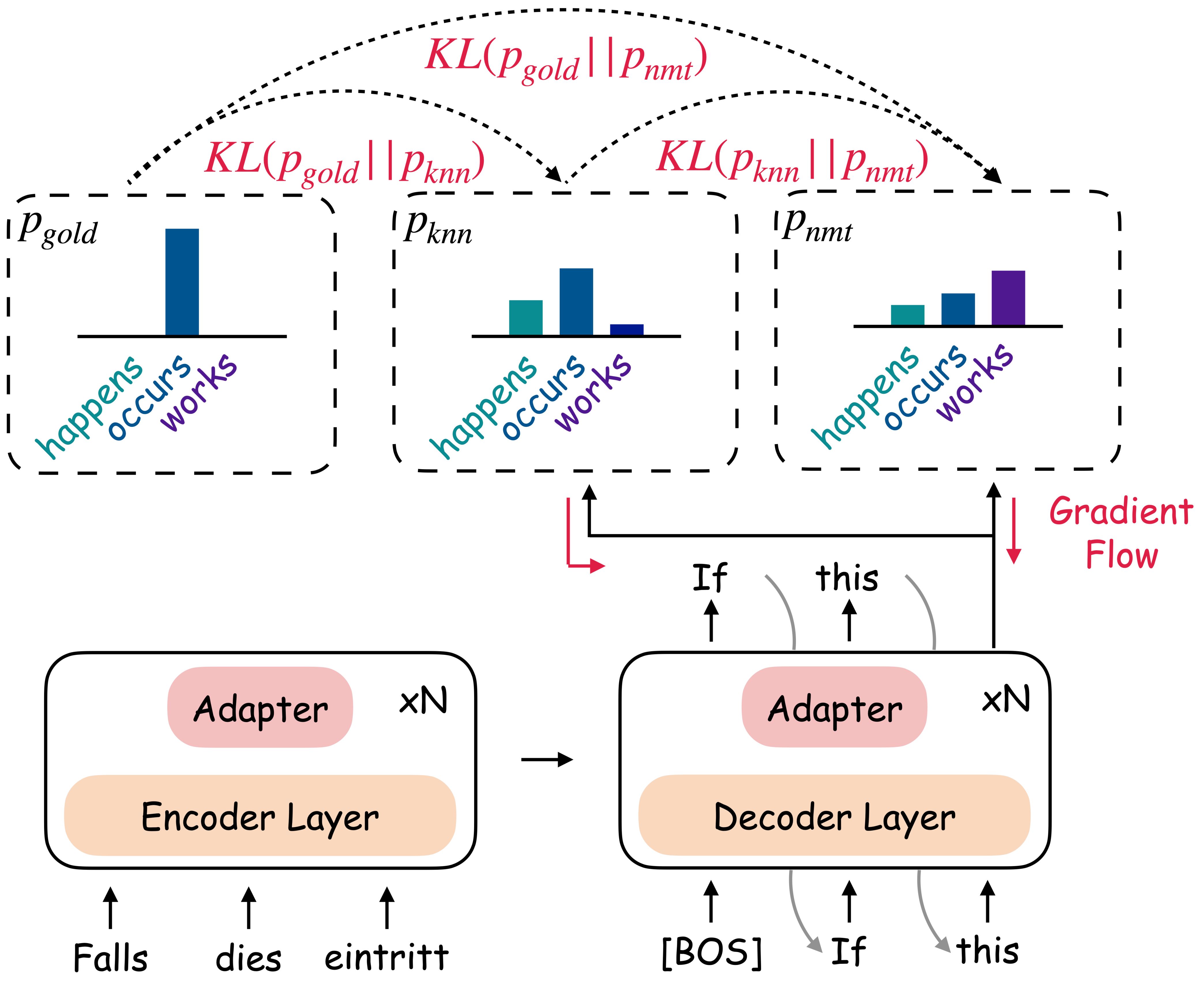
    extracting kNN knowledge to adjust representation
  - asynchronous refresh

    using updated representation to refresh the datastore



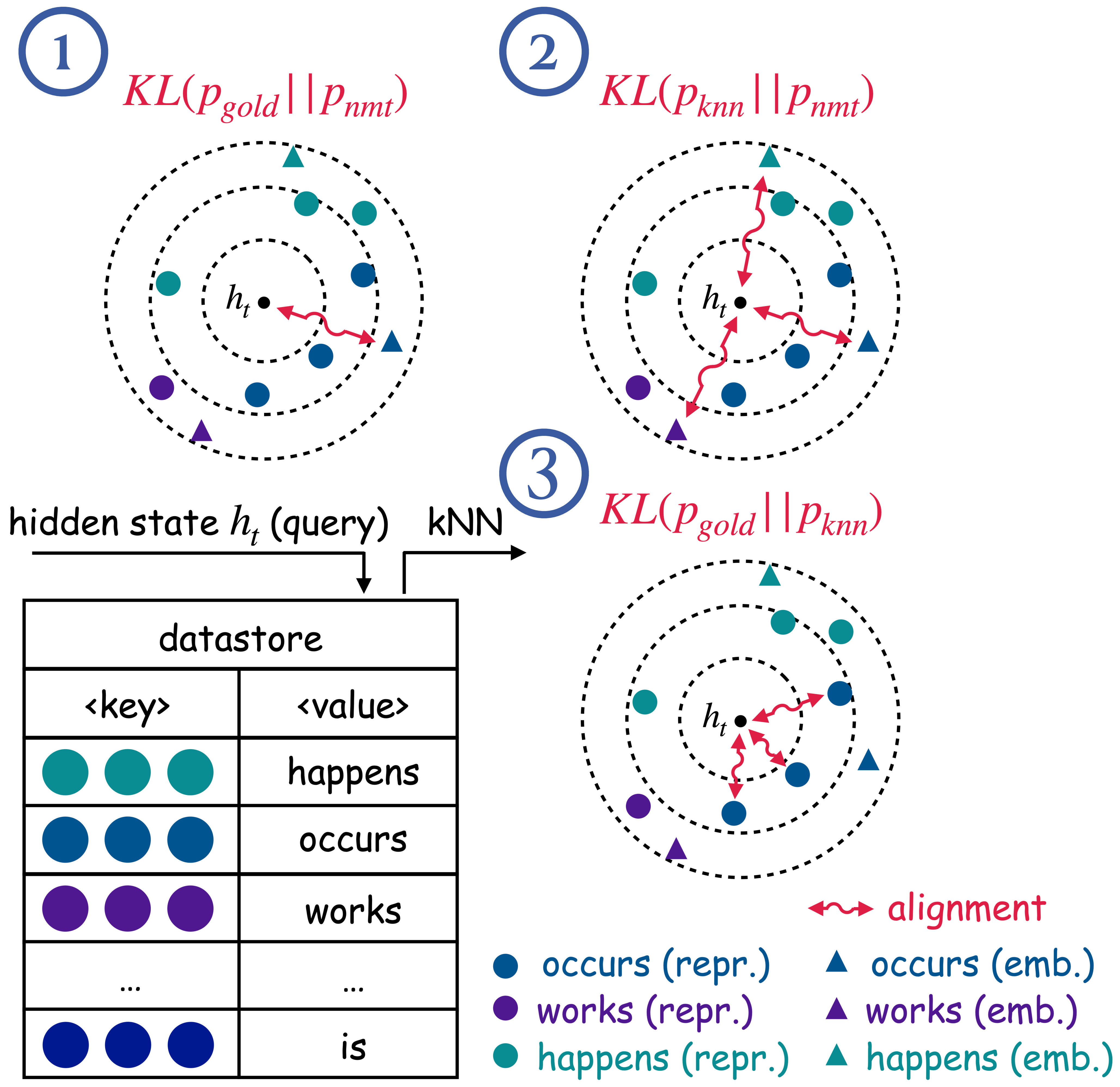
Smoothing Representation Space with INK

- We adjust the representation by aligning three kinds of representations with KL-divergence.



Smoothing Representation with INK

1. Aligning contextualized representations and token embeddings.
2. Aligning contextualized representations and kNN token embeddings.
3. Aligning contextualized representations of the same target token.



Smoothing Representation with INK

- overall training procedure
  - optimizing adapter with the combined learning objective

$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{(X,Y)\in\mathcal{B}} \sum_{t} (\mathcal{L}_t^a + \alpha\mathcal{L}_t^i + \beta\mathcal{L}_t^r)$$

  - refreshing datastore asynchronously

- During inference, we only need to load the off-the-shelf NMT model and tuned adaptation parameters.

Main Results

- INK system achieves the best performance by smoothing the representation space.

Systems	Mem.	Medical		Law		IT		Koran		Avg.	
		COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU
Off-the-shelf NMT	-	46.87	40.00	57.52	45.47	39.22	38.39	-1.32	16.26	35.57	35.03
*k*NN-KD	-	56.20	56.37	68.60	60.65	-1.57	1.48	-13.05	19.60	27.55	34.53
NMT + Datastore Augmentation											
V-*k*NN	$\times$1.7	53.46	54.27	66.03	61.34	51.72	45.56	0.73	20.61	42.98	45.45
A-*k*NN	$\times$1.7	57.45	56.21	69.59	63.13	56.89	47.37	4.68	20.44	47.15	46.79
R-*k*NN†	$\times$1.7	58.05	54.16	69.10	60.90	54.60	45.61	3.99	20.04	46.44	45.18
R-*k*NN	$\times$43.8	57.70	57.12	70.10	**63.74**	57.65	48.50	5.28	20.81	47.68	47.54
NMT + Representation Refinement											
Adapter	$\times$1.0	60.14	56.88	70.87	60.64	66.86	48.21	4.23	21.68	50.53	46.85
INK (ours)	$\times$1.0	**61.64***	**57.75***	**71.13**	61.90*	**68.45***	**49.12***	**8.84***	**23.06***	**52.52**	**47.85**

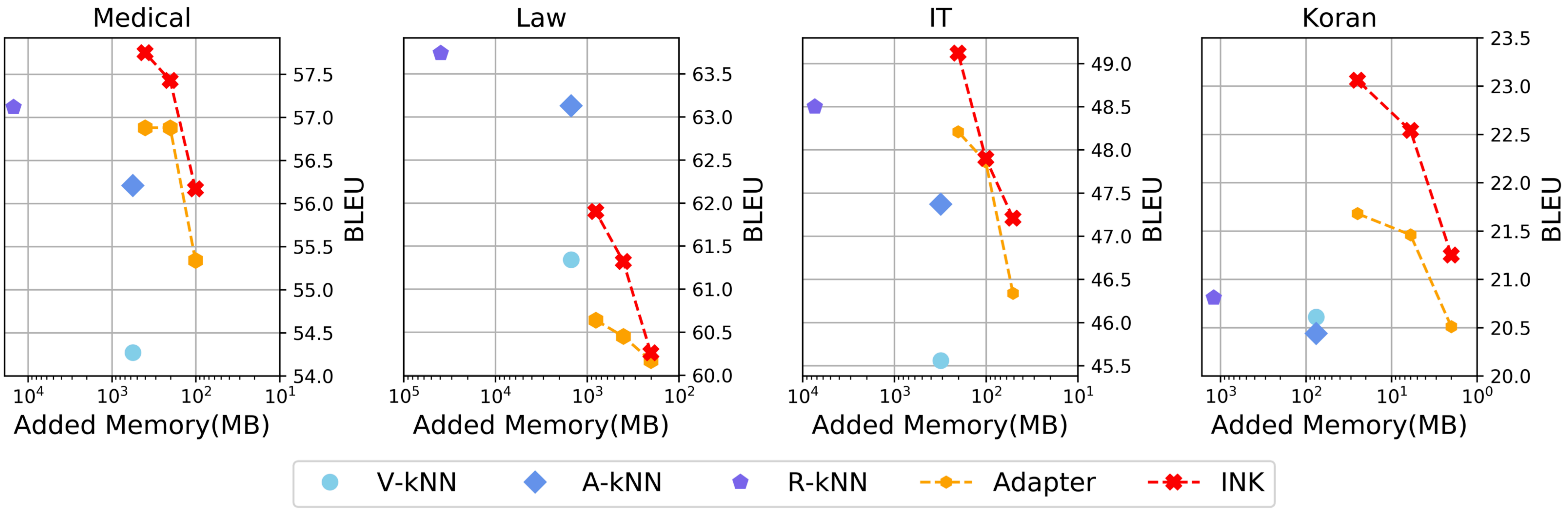
Main Results (Cont.)

- Representation refinement according to kNN knowledge brings larger performance improvement.

Systems	Mem.	Medical		Law		IT		Koran		Avg.	
		COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU
Off-the-shelf NMT	-	46.87	40.00	57.52	45.47	39.22	38.39	-1.32	16.26	35.57	35.03
kNN-KD	-	56.20	56.37	68.60	60.65	-1.57	1.48	-13.05	19.60	27.55	34.53
NMT + Datastore Augmentation											
V-kNN	×1.7	53.46	54.27	66.03	61.34	51.72	45.56	0.73	20.61	42.98	45.45
A-kNN	×1.7	57.45	56.21	69.59	63.13	56.89	47.37	4.68	20.44	47.15	46.79
R-kNN†	×1.7	58.05	54.16	69.10	60.90	54.60	45.61	3.99	20.04	46.44	45.18
R-kNN	×43.8	57.70	57.12	70.10	**63.74**	57.65	48.50	5.28	20.81	47.68	47.54
NMT + Representation Refinement											
Adapter	×1.0	60.14	56.88	70.87	60.64	66.86	48.21	4.23	21.68	50.53	46.85
INK (ours)	×1.0	**61.64***	**57.75***	**71.13**	61.90*	**68.45***	**49.12***	**8.84***	**23.06***	**52.52**	**47.85**

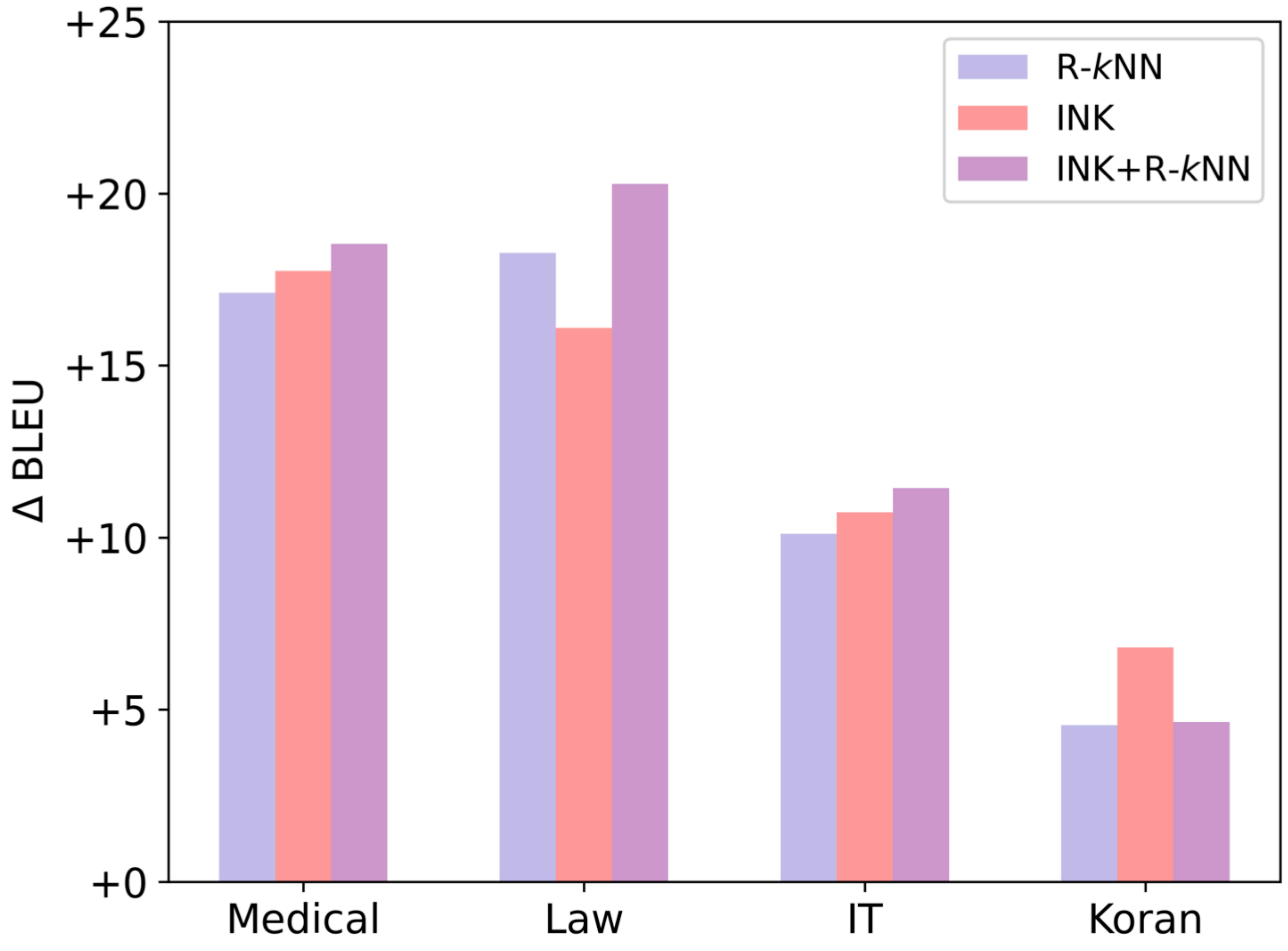
Main Results (Cont.)

- Representation refinement according to kNN knowledge brings larger performance improvement.



Main Results (Cont.)

- Jointly applying adapter and datastore can further smooth predictions.



Conclusion

- INK iteratively refines the representation space of the NMT model according to kNN knowledge.
- INK system achieves an average gain of 1.99 COMET and 1.0 BLEU.
- Compared with kNN-MT baselines, our INK achieves better translation performance with 0.02× memory space and 1.9× inference speed up.

📄 Paper

📖 文字解读

📌 Slides

What Knowledge Is Needed? Towards Explainable Memory for kNN-MT Domain Adaptation

Wenhao Zhu[1,2], Shujian Huang[1,2], Yunzhe Lv[1,2], Xin Zheng[1,2], Jiajun Chen[1,2]

National Key Laboratory for Novel Software Technology, Nanjing University

Collaborative Innovation Center of Novel Software Technology and Industrialization

Motivation

- kNN-MT incorporates the symbolic datstore to assist the neural model, which usually saves all target language token occurences in the parallel corpus.
- The constructed datastore is usually large and possibly redundant.

translation context $(X, Y_{<t})$		hidden state $h(X, Y_{<t})$	target token y_t
Wie wirkt Penizillin?	<bos>		How
Wie wirkt Penizillin?	<bos> How		Does
Wie wirkt Penizillin?	<bos> How does		Penicillin
Wie wirkt Penizillin?	<bos> How does Penicillin		work
Wie wirkt Penizillin?	<bos> How does Penicillin work		?
Wie wirkt Penizillin?	<bos> How does Penicillin work ?		<eos>

key-value datastore

stored knowledge:
generate the value token
at the hidden state key

What Knowledge Does the Neural Model Need?

- The relationship between NMT model and symbolic datastore is unclear.
- Intuitively, the pre-trained NMT model only needs knowledge that remedies its weakness.
- We propose to explore this issue from the point of "local correctness"
  - translation correctness for a single entry (entry correctness)
  - Translation correctness for a given neighborhood (neighborhood correctness).

Local Correctness

- entry correctness
  - Entry correctness describes whether the NMT model could make correct translation for a specific entry.
  - It can be evaluated by comparing target token and prediction token:

translation context $(X, Y_{<t})$		hidden state $h(X, Y_{<t})$	target token y_t	predict token $\hat{y}_t$	
Wie wirkt Penizillin ?	<bos>		How	How	known
Wie wirkt Penizillin ?	<bos> How		Does	Does	known
Wie wirkt Penizillin ?	<bos> How does		Penizillin	Cyanokit	unknown
Wie wirkt Penizillin ?	<bos> How does Penicillin		work	works	unknown
Wie wirkt Penizillin ?	<bos> How does Penicillin work		?	?	known
Wie wirkt Penizillin ?	<bos> How does Penicillin work ?		<eos>	<eos>	known

check

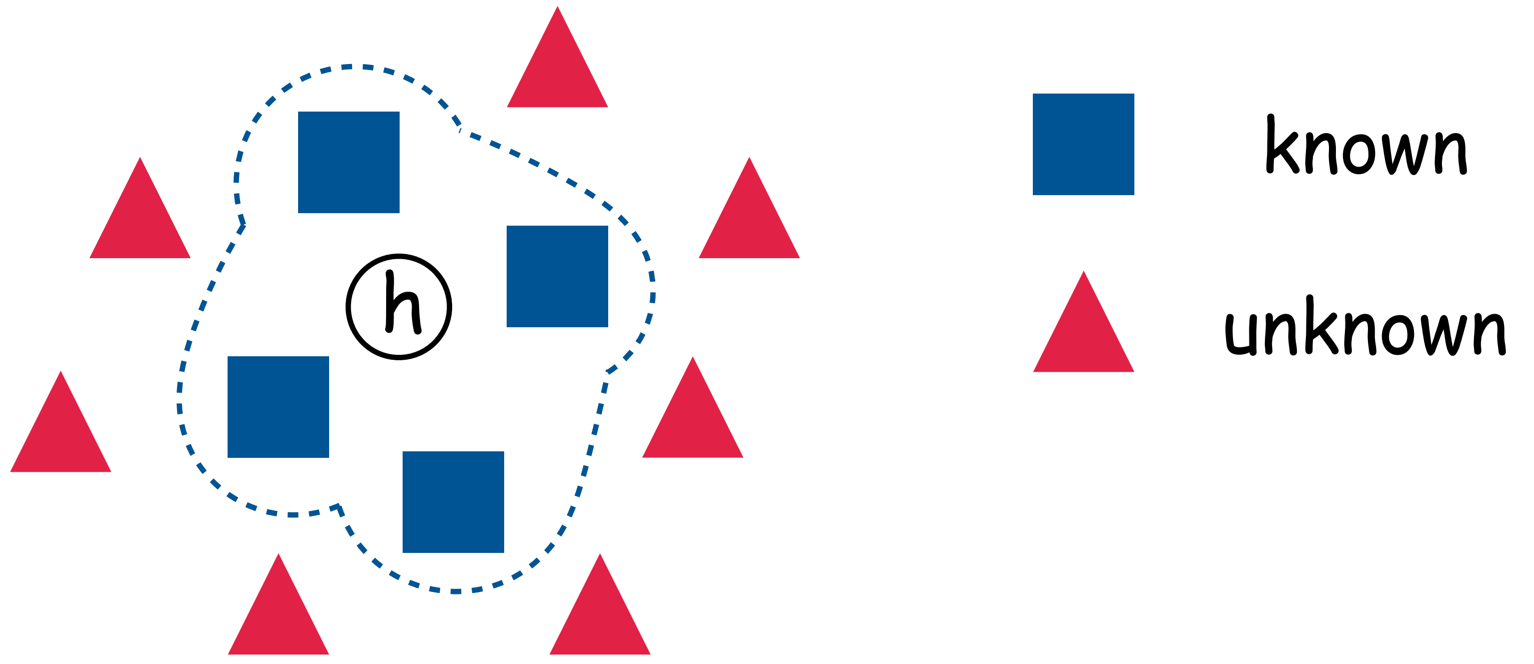
Local Correctness

- neighborhood correctness
  - Neighborhood correctness evaluates the NMT model's prediction on a neighborhood in the representation space.
  - Knowledge margin is proposed as the metric.

knowledge margin

$$km(h) = \arg\max_{t} \forall (h^j, y^j) \in \mathcal{N}_t(h) \text{ is known}$$

$km(h) = 4$



Intuitively, km is the maximum size of the neighborhood of the entry h where the NMT could make correct translation

Local Correctness

- understanding the role of different datastore entries
  - Entries with small km: NMT model tends to fail when context are similar but different. helpful
  - Entries with large km: NMT model generalizes well on these entries. less helpful
- PLAC: Pruning with LocAl Correcness

Algorithm 1 Datastore Pruning by PLAC

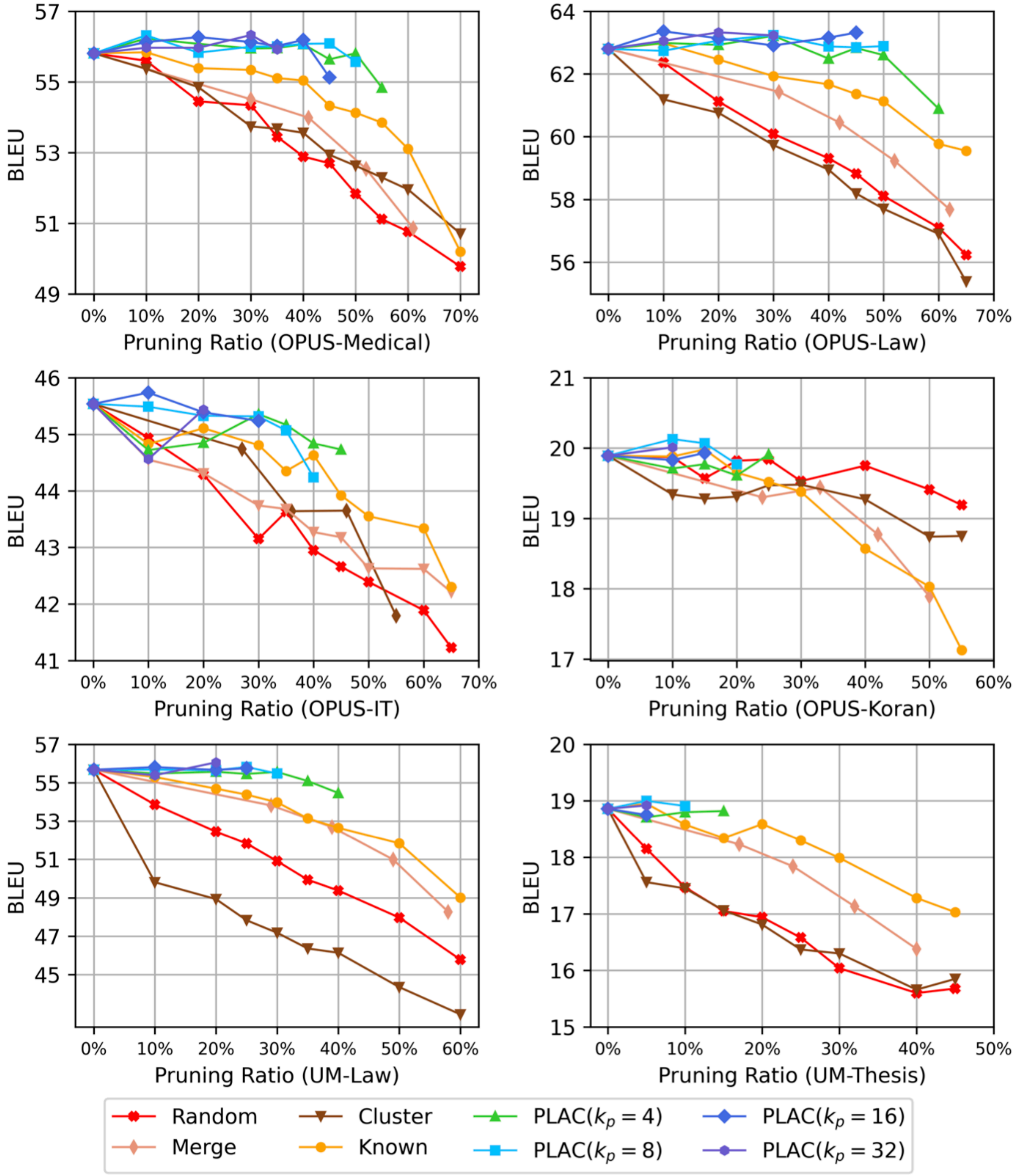
Input: datastore $\mathcal{D}$, the *knowledge margin* threshold k_p, the pruning ratio r
Output: pruned datastore $\mathcal{D}$

$candidates \leftarrow \emptyset$ ▷ step 1: collect
for each entry (h, y) in $\mathcal{D}$ **do**
  if $km(h) \geq k_p$ **then**:
    $candidates \leftarrow candidates \cup (h, y)$
  end if
end for
repeat ▷ step 2: drop
  randomly select entry (h, y) from *candidates*
  remove (h, y) from $\mathcal{D}$
until pruning ratio r is satisfied
return $\mathcal{D}$

Experiment Results

- Pruning with local correctness (PLAC) cuts off 25%-45% datastore entries while achieve comparable performance
  - Previous pruning method (40% -1.4 BLEU, 10% -0.9 BLEU)

	OPUS-Medical			OPUS-Law			OPUS-IT			OPUS-Koran		
	Ratio	BLEU↑	COMET↑	Ratio	BLEU↑	COMET↑	Ratio	BLEU↑	COMET↑	Ratio	BLEU↑	COMET↑
Base	-	39.73	0.4665	-	45.68	0.5761	-	37.94	0.3862	-	16.37	-0.0097
Finetune	-	58.09	0.5725	-	62.67	0.6849	-	49.08	0.6343	-	22.40	0.0551
Adaptive *k*NN	0%	57.98	0.5801	0%	63.53	0.7033	0%	48.39	0.5694	0%	20.67	0.0364
Random	45%	54.08*	0.5677	45%	58.69*	0.6690*	40%	45.54*	0.5314*	25%	20.36	0.0434
Cluster	45%	53.31*	0.5689	45%	58.68*	0.6779*	40%	45.80*	0.5788	25%	20.04*	0.0410
Merge	45%	54.65*	0.5523*	45%	60.60*	0.6776*	40%	45.83*	0.5334*	25%	20.25*	0.0365
Known	45%	56.44*	0.5691	45%	61.61*	0.6885*	40%	45.93*	0.5563	25%	20.35*	0.0338
All Known	73%	42.73*	0.4926*	66%	51.90*	0.6200*	69%	40.93*	0.4604*	56%	17.76*	0.0008*
PLAC (ours)	45%	57.66	0.5773	45%	63.22	0.6953*	40%	48.22	0.5560	25%	20.96	0.0442



Conclusion

- We analyze the local correctness of the neural model's predictions to identify the conditions where the neural model may fail.
- We find that the NMT model often fails when the knowledge margin is small.
- We can safely prune the datastore with the proposed PLAC method, validating our findings about local correctness and translation failures.

📄 Paper | 📖 文字解读 | 📌 Slides | ⚒开源框架