In-Context Learning for Discrete Optimal Transport: Can Transformers Sort?

Hadi Daneshmand
Department of Computer Science, University of Virginia
dhadi@virginia.edu

Abstract

The rapid growth of model sizes and training datasets has created a strong demand for *test-time compute*—the ability to perform inference without additional training. At the core of test-time compute is *in-context learning* (ICL), an emerging capability of large language models (LLMs) that enables them to perform statistical inference directly at test time. Recent progress has shed light on the mechanisms underlying in-context learning in statistical tasks: language models can implement linear regression and classification by iteratively extracting features at test time. This naturally raises a broader question: *Can we analyze ICL beyond statistical learning and extend it to discrete algorithmic tasks relevant to NLP?*

One of the fundamental tasks in NLP can be formulated as discrete optimal transport: matching tokens, with applications ranging from machine translation to mixture-of-experts routing. We show that transformers with softmax self-attention can solve discrete optimal transport via in-context learning when the model parameters are fixed and only the input length and data distribution vary. One implication of this result is that transformers can approximately sort lists of arbitrary length with a provable approximation guarantee.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

1 INTRODUCTION

In-context learning (ICL) is an emerging capability of large language models (LLMs) that allows them to perform statistical inference directly from data provided in their prompt, without parameter updates. Recent work has extensively investigated the mechanism of in-context statistical learning, examining the emergence of in-context linear regression [Ahn et al., 2023, Von Oswald et al., 2023, Garg et al., 2022, Akyürek et al., 2023, Vaswani et al., 2017], classification [Shen et al., 2024, Tarzanagh et al., 2023], matrix completion [Lutz et al., 2025] and reinforcement learning [Wang et al., 2025]. While prior mechanistic studies focus on regression/classification, many NLP tasks are discrete. Can transformers' ICL extend to such combinatorial tasks?

This paper focuses on *discrete optimal transport (OT)* with broad applications in data mining [Peyré et al., 2019], bio-informatics [Schiebinger et al., 2019], and natural language processing [Munkres, 1957]. Given two sets of n points $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$ in $\mathbb{R}^d$, the discrete OT problem seeks a permutation matrix $P^* \in \{0,1\}^{n\times n}$ that minimizes the total matching cost

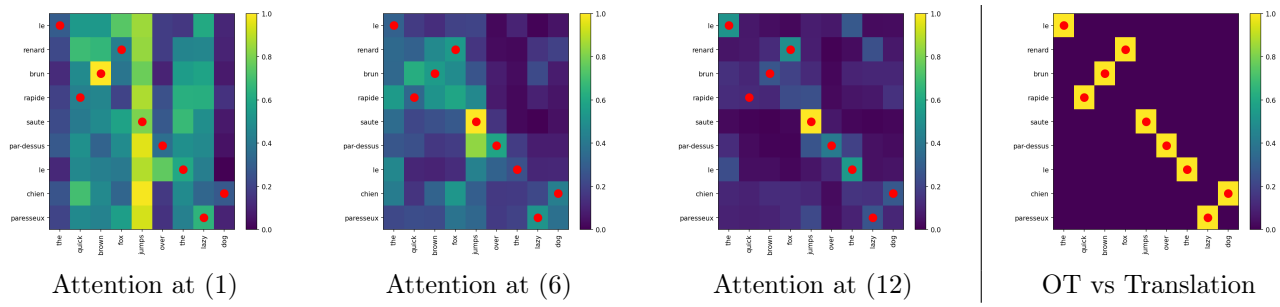
$$P^* = \arg\min_{P\in\Pi_n} \sum_{i,j} P_{ij} \|y_i - x_j\|_2^2, \tag{1}$$

where Π_n is the set of $n \times n$ permutation matrices. Sorting is a special case of discrete OT when $d = 1$ and the y_i's are ordered.

1.1 Translation with OT

We connect optimal transport (OT) to the mechanism of language translation in LLMs by analyzing their internal word embeddings. As an illustrative example, we prompt BERT with the following pair of translated sentences in English and French:

$$\begin{cases} \text{The quick brown fox jumps over the lazy dog (EN)} \\ \text{Le renard brun rapide saute par-dessus le chien paresseux (FR)} \end{cases}$$



Attention at (1) Attention at (6) Attention at (12) OT vs Translation

Figure 1: **Translation with OT.** The rightmost plot shows the optimal transport (OT) solution from (1), computed between the English word embeddings $(x_1, \ldots, x_n)$ and the French word embeddings $(y_1, \ldots, y_n)$. Red dots mark correctly aligned translation pairs. Observe the OT solution matches words with the same meaning. The other plots depict attention-weight heatmaps from layers 1, 6, and 12, showing how the model iteratively approximates the OT solution.

Let $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ denote the vector representations of the English and French words, respectively, from the first encoder layer. Fig. 1 shows that the OT solution in (1) aligns pairs of translated words. See Appendix Figure 9 for the correlation plot. We present further experiments on 1000 sentences in Section 5.2, demonstrating that OT on word embeddings aligns words for translation.

Even more interesting is the pattern of average attention weights (over heads) across the transformer layers. Fig. 1 shows that the attention weights progressively approximate the OT solution (P^* in (1)) as depth increases. Remarkably, this interpretable mechanism emerges naturally after pretraining, without any explicit regularization. In other words, the transformer learns to align translated words across its layers. We present a more extensive experiment on 10,000 translated sentences on Marian model [Junczys-Dowmunt et al., 2018] in Section 5.1, supporting the above observation. But how can we explain this striking algorithmic behavior of attentions? To address this question, we perform additional experiments on a transformer specifically trained to solve OT.

1.2 Observations on OT Mechanism

To analyze the OT mechanism of attention, we trained a transformer to solve discrete OT on small problem instances with $n = 7$ points and evaluated it on larger instances with $n = 9$. For training, we generated synthetic data from a standard neutral distribution widely used to study in-context learning [Garg et al., 2022] (see Appendix A for details). Surprisingly, the transformer can approximate the optimal OT solutions on larger inputs (Fig. 2). This out-of-distribution generalization suggests that transformers can adapt to input sizes beyond those seen in training, a key requirement for in-context learning. This result is particularly important given prior efforts to modify transformers for solving OT [Sander et al., 2022, Tay et al., 2020]. Yet, Fig. 2 demonstrates that standard transformers are capable of solving the optimal transport problem.

A critical factor behind the observed generalization is *prompt engineering*. Careful input augmentation effectively extends the transformer's memory, thereby significantly enhancing its computational capacity to solve OT; see Observation (2) in Fig. 2. We provide further details and insights on this specific prompt construction in the following sections.

The proposed prompt design encourages the attention layers to solve the OT problem iteratively as depth increases. As shown in Fig. 2, initially diffuse attention weights gradually converge layer by layer toward the optimal OT solution. While such *iterative inference* has been analyzed for linear regression [Ahn et al., 2023, Lutz et al., 2025], it remains unclear whether these results extend beyond linear regression.

1.3 Contributions

We establish three main contributions:

Contribution (1): Mechanistic Analysis. We discover a link between the mechanism of transformers for language translation and OT. Our observations show that attention weights align translated words iteratively across the layers, which approximates the OT solution for word embeddings.

To explain this striking observation, we analyze the internal dynamics of feature extraction in transformers. We show that the fundamental building block softmax self-attention [Vaswani et al., 2017] is particularly well suited to *implement* OT. In particular, a single self-attention layer can simulate an iterative first-order optimization method for OT, thereby explaining the iterative inference behavior observed in Fig. 2.

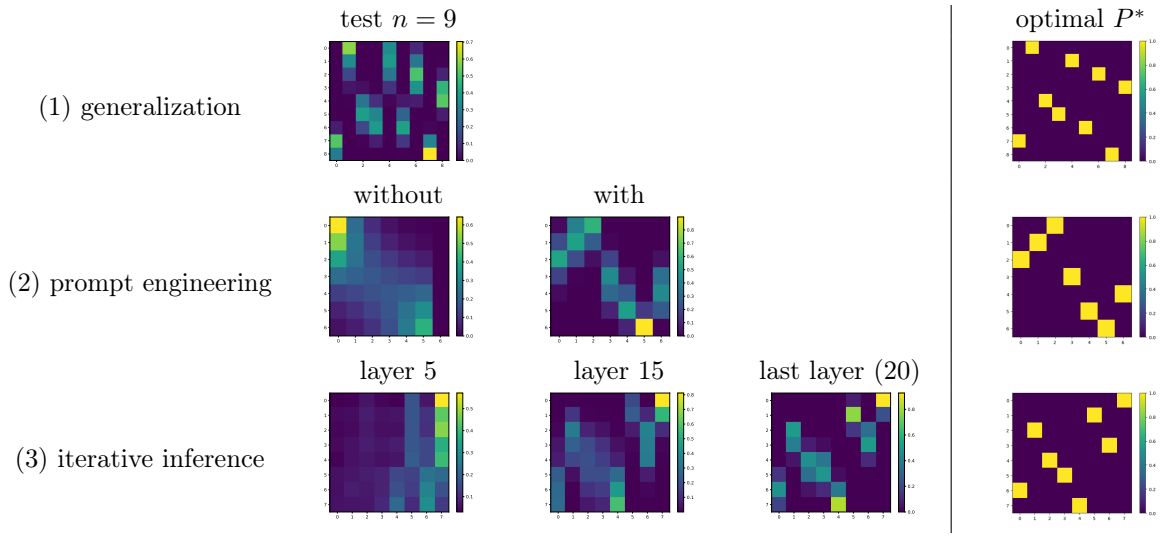


Figure 2: **Observations on In-context Learning for OT.** *(1)* The model is trained to solve OT with 7 data points and evaluated on 9 data points. The left image shows the attention weights, which closely approximate the OT solution shown on the right. *(2)* After specific prompt engineering, the attention weights between tokens estimate the OT solution. Notably, this prompt engineering is used in (1). *(3)* The attention weights evolve across layers, progressively yielding a more accurate approximation of the optimal solution. See Appendix A for details.

Contribution (2): Theoretical guarantees. Leveraging the mechanistic insight, we prove: for any two sets of n points in $\mathbb{R}^d$, a transformer can estimate the OT solution with

$$O\Big(\frac{n^{3/2}}{\text{depth}^{1/2}}\Big)\text{-accuracy for all integers } n. \qquad (2)$$

Consequently, sufficiently deep transformers can solve OT for a wide range of n without modifying their parameters, explaining scale generalization, Observation (1) in Fig. 2.

Contribution (3): Insights on prompt engineering. Crucially, the established results hinge on *prompt engineering*, which provides an extended memory for the attention layers. It also offers a new insight the mechanism of prompt engineering.

A Note on Applications. While our primary focus has been interpretability and mechanistic understanding, our theoretical results have also informed the design of foundation models for downstream applications. In particular, Adduri et al. [2025] develop a novel transformer architecture for single-cell perturbation analysis.

2 BACKGROUND

2.1 Entropy regularization

While OT is constrained to the combinatorial set of permutation matrices, it can be relaxed to optimization over a continuous set. The state-of-the-art method is based on linear programming, namely

$$\widehat{P} = \arg\min_{P\in\mathbb{R}^{n\times n}} \sum_{ij} P_{ij}C_{ij},$$
$$\text{subject to } P \text{ is doubly stochastic,}$$

where $C \in \mathbb{R}^{n\times n}$ is a cost matrix such that $C_{ij} = \|y_i - x_j\|^2$. Comparing the above problem with the original problem in (1), we notice that the constraint requiring P to be a permutation matrix has been relaxed to allowing P to be a doubly stochastic matrix. Recall the solutions of linear programs lie among the extreme points of the constraint set. Since the extreme points of doubly stochastic matrices are permutation matrices [Brockett and Wong, 1991], the above linear program has the same solution as (1), i.e., $P^* = \widehat{P}$.

The above linear program has $O(n^2)$ variables. Due to the quadratic growth with n, solving the linear program becomes computationally challenging for large n. Cuturi [2013] proposes a computationally efficient alternative based on regularization with entropy:

$$P^*_\lambda := \arg\min_{P\in\mathbb{R}^{n\times n}} \sum_{ij} P_{ij}C_{ij} + \lambda P_{ij}\log(P_{ij}), \qquad (3)$$
$$\text{subject to } P \text{ is doubly stochastic}$$

The Lagrangian dual of the above program has only $O(n)$ variables which is considerably fewer than $O(n^2)$ variables for the original linear program. Introducing the dual parameters $v \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$, the

Lagrangian function is defined as follows:

$$\sum_{ij} P_{ij}C_{ij} + \lambda \sum_{ij} P_{ij}\log(P_{ij}) - u^\top (P1_n - 1_n) - v^\top \left(P^\top 1_n - 1_n\right).$$

$P_{ij} = e^{\frac{-C_{ij}+v_j+u_i}{\lambda}-1}$ minimizes the Lagrangian function. This structure inspired the use of Sinkhorn's fixed point iteration to find the solution of the dual problem. In particular, Sinkhorn [1967] proves that there exists a unique doubly stochastic matrix of the form $[P^*_\lambda]_{ij} = e^{\frac{-C_{ij}+v^*_i+u^*_j}{\lambda}-1}$ that is the solution of a simple fixed-point iteration where u^*, v^* are unique up to scaling factors. $[M]_{ij}$ denotes the element of the matrix M located in row i and column j. Leveraging this theorem, Cuturi [2013] proves Sinkhorn recurrence can efficiently find P^*_λ.

Apart from Sinkhorn's recurrence other optimization methods can also solve Lagrangian dual problem. Recall the minimizer $P_{ij} = e^{\frac{-C_{ij}+u_i+v_j}{\lambda}-1}$. Plugging this into the Lagrangian function yields

$$\min_{v,u\in\mathbb{R}^n} \underbrace{\lambda\left(\sum_{ij} e^{\frac{-C_{ij}+u_i+v_j}{\lambda}-1}\right) - \sum v_i - \sum_i u_i}_{L(u,v)} \quad (4)$$

It is easy to check that L is convex in u and v as its Hessian is diagonally dominant, hence positive semi-definite. Thus, standard first-order optimization can optimize L such as gradient descent with adaptive coordinate-wise stepsizes:

$$\begin{cases} u_{\ell+1} = u_\ell - D_\ell \nabla_u L(u_\ell, v_\ell) \\ v_{\ell+1} = v_\ell - D'_\ell \nabla_v L(u_\ell, v_\ell) \end{cases}, \quad (5)$$

where $\nabla_u L$ denotes the gradient of L with respect to u and $D_\ell, D'_\ell \in \mathbb{R}^{n\times n}$ are diagonal matrices with positive diagonal elements which include stepsize for each coordinate of vectors u_ℓ and v_ℓ. We will prove that self-attention in transformers can simulate the above iterations.

2.2 Soft-max Self-attention

Attention layers are fundamental building blocks of neural networks, developed over decades of research. Hochreiter and Schmidhuber [1997] pioneered this development by proposing an attention mechanism for Recurrent Neural Networks (RNNs) inspired by human cognition. Graves et al. [2014] employs the attention mechanism to develop a memory system for a parametric version of the Turing machine. Bahdanau et al. [2014] adapts this attention mechanism in neural Turing machine to design a powerful model for machine translation. While attention was originally introduced for recurrent models, Vaswani et al. [2017] proposes non-recurrent attention layers, combined with residual connections [He et al., 2016], thereby significantly enhancing the training of attention weights.

Attention layers rely on a convex combination. Let $Z \in \mathbb{R}^{n\times d}$. An attention layer is a function denoted by $\text{atten}_\theta : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ with parameters $\theta := [K, Q, V \in \mathbb{R}^{d\times d}]$ defined as

$$\text{atten}_\theta(Z) = AZV, \quad A_{ij} = \frac{e^{\langle Kz_i, Qz_j\rangle}}{\sum_{j=1}^n e^{\langle Kz_i, Qz_j\rangle}},$$

where z_i and z_j are rows of Z. The convex combination of data points induces a local dependency between the representations of tokens, thereby capturing the contextual relationships among them.

Tay et al. [2020] investigate whether attention layers can perform sorting. Since standard self-attention layers cannot directly implement Sinkhorn's iteration, Tay et al. [2020], Sander et al. [2022] propose a novel attention mechanism called *Sinkhorn attention*. However, because standard self-attention remains widely used, we investigate whether the standard self-attention can approximate the OT solution.

2.3 The Engineered Prompt

We propose a particular input denoted by $Z_0 \in \mathbb{R}^{(n+1)\times(2d+9)}$ to encode the assignment problem:

$$Z_0 = \begin{bmatrix} x_1 & y_1 & \|x_1\|^2 & \|y_1\|^2 & 1_4 & 0_3 \\ x_2 & y_2 & \|x_2\|^2 & \|y_2\|^2 & 1_4 & 0_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ x_n & y_n & \|x_n\|^2 & \|y_n\|^2 & 1_4 & 0_3 \\ 0_d & 0_d & 0 & 0 & -v_4 & 0_3 \end{bmatrix}. \quad (6)$$

The elements in blue are the original prompts, which are sufficient for the assignment. The elements in red are carefully engineered. 1_4 denotes the all-ones 4-dimensional vector, 0_d denotes the all-zeros d-dimensional vector, and $v_4 = [0, 0, 0, 1]$.

Our analysis shows that the engineered prompt gives the transformer enough input structure and memory to simulate optimization methods for optimal transport (OT). The appended zeros create an augmented representation across layers, allowing attention to store and update gradient-descent iterates. In experiments, using the natural encoding of (x_i, y_i) in the prompt reduces performance (see Observation (2) in Fig. 2).

2.4 Transformer

We consider a specific transformer architecture composed of multiple soft-max self-attention with residual

connections. Let Z_ℓ denote the intermediate representation of the input Z_0 at layer ℓ which obeys the following recurrence

$$Z_{\ell+1} = Z_\ell + \sum_{j=1}^{2} \text{atten}_{\theta_j^{(\ell)}}(Z_\ell) B_j^{(\ell)}, \tag{7}$$

where $\theta_j^{(\ell)}$ are parameters of the attention layers, $B_j^{(\ell)} \in \mathbb{R}^{d' \times d'}$ are the weights to combine attention heads. Remarkably, the model has two attention heads. Remarkably, all our results extend to transformer blocks that include both attention and feedforward layers. This is because feedforward layers with zero weights, combined with residual connections, effectively reduce the architecture to one consisting solely of self-attention layers. Moreover, restricting the analysis to two attention heads is not a limitation: any transformer with more heads can be reduced to an equivalent two-head attention block by zeroing out the mixing weights of the additional heads. Finally, although we do not explicitly include attention masks in our formulations, appropriate masking can be incorporated without affecting the validity of our analysis.

2.5 Notations

Table 1: Notation summary

Symbol	Definition
n	The number of points in OT
d	Dimension
$\{x_i/y_i\}_{i=1}^n$	Sets of points in $\mathbb{R}^d$ for OT
$C \in \mathbb{R}^{n\times n}$	Cost with $C_{ij} = \|y_i - x_j\|_2^2$
Π_n	Set of permutation matrices
$P^* \in \{0,1\}^{n\times n}$	Optimal transport in (1)
$\lambda > 0$	Regularization parameter in (3)
$P_\lambda^* \in \mathbb{R}^{n\times n}$	Optimal regularized OT plan
$u, v \in \mathbb{R}^n$	Dual parameters
$Z_\ell \in \mathbb{R}^{n\times d}$	Token representations at layer ℓ
$A^{(\ell)} \in \mathbb{R}^{n\times n}$	Attention weight at layer ℓ

Let $Z_0 \in \mathbb{R}^{n\times d}$ denote the input to the transformer. Its representation at layer ℓ is denoted by Z_ℓ. The input matrix Z_0 contains the points $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ from the assignment problem. Let P^* be the optimal assignment solution, and let P_λ^* be its approximation obtained via entropy regularization. The transformer is parameterized by θ, which includes all attention parameters $\theta_j^{(\ell)}$ at layer ℓ and attention head j. The vectors u_ℓ and v_ℓ denote the gradient descent iterates on the dual objective function L, as defined in (4). For a matrix M, we use the notation $[M]_{i,j}$ to refer to the entry in row i and column j. See Table 1 for a summary.

3 THE OT MECHANISM OF SOFTMAX ATTENTION

What is given to a language model is ultimately a vector representation of words (word embeddings). How does the language model matches words with the same meaning given the word embeddings? Fig. 2 shows the process of alignment is iterative across the layers. We mathematically prove this iterative alignment by linking attention weights to dual OT. In particular, the next theorem states that ℓ attention layers stacked in transformers can simulate ℓ-iterations of gradient descent on the dual objective function L defined in (4).

Theorem 3.1 (Dual OT with Transformers)

Let $D_\ell, D'_\ell \in \mathbb{R}^{n\times n}$ are diagonal matrices whose diagonal elements are

$$\begin{cases} (\gamma_\ell [D_\ell]_{ii})^{-1} = \sum_j e^{(-C_{ij} + [u_\ell]_i + [v_\ell]_j)/\lambda - 1} + 1, \\ (\gamma_\ell [D'_\ell]_{jj})^{-1} = \sum_i e^{(-C_{ij} + [u_\ell]_i + [v_\ell]_j)/\lambda - 1} + 1. \end{cases}$$

There exists a configuration of parameters *independent from inputs and* n such that

$$\begin{cases} [Z_\ell]_{(1:n),(2d+7)} = u_\ell - D_\ell \nabla_u L(u_\ell, v_\ell) \\ [Z_\ell]_{(1:n),(2d+8)} = v_\ell - D'_\ell \nabla_v L(u_\ell, v_\ell) \end{cases},$$

holds for all integer values of n, where u_ℓ and v_ℓ are gradient descent in (5) iterations starting from $u_0 = v_0 = 0$.

Notably, the above result supports the *"iterative inference hypothesis"* [Jastrzebski et al., 2018], that links the mechanism of deep neural networks to optimization methods. This hypothesis postulates that residual connections enable deep networks to implicitly implement gradient descent across layers to tackle complex tasks. The hypothesis is based on striking observations on the underlying mechanisms of Convolutional Neural Networks (CNNs) [Alain, 2016]. Previous studies have theoretically proven this hypothesis for solving least-squares using transformers [Ahn et al., 2023, Von Oswald et al., 2023, Akyürek et al., 2023]. Building on these studies, we demonstrate that transformers can implement gradient descent for OT and connect it to language translation.

We highlight that the established result holds for standard **softmax self-attention**. In particular, we show that softmax attention is especially well-suited to implementing gradient descent on the objective function L, as defined in (4). Remarkably, attention mechanisms cannot simulate the gradient of an arbitrary function and exhibit inherent limitations in their computational capabilities. In particular, *linear attention*

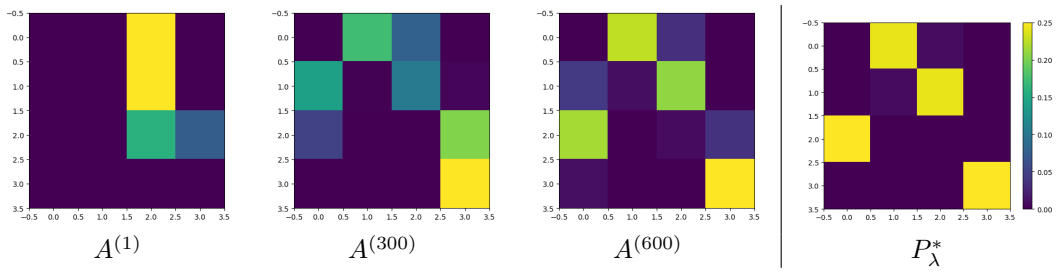


Figure 3: **Convergence of attention matrices.** The plotted matrices are attention weights in layers (1), (300) and (600). We observe these matrices converge to the regularized OT solution (the rightmost plot), which is proven by Theorem 3.2.

is well-suited for problems such as linear regression, where the underlying objective is quadratic [Ahn et al., 2023, Von Oswald et al., 2023]. In contrast, *softmax attention* is better aligned with tasks involving token matching.

Prompt engineering is essential for the proof of Theorem 3.1. Expanding the input size by adding columns and rows creates an extended data representation matrix across the layers. Attention layers can utilize a part of this matrix *as memory to store the iterates of gradient descent* . Furthermore, the input dependent part of the prompt supplies the necessary statistics for the attention layers to implement gradient descent.

By connecting the intermediate data representations to a well-established algorithm, we can leverage powerful theoretical tools to prove that a transformer with frozen weights can approximate OT solution. We first present an informal version of the result below and defer the full theorem and proof to Appendix D.

Theorem 3.2 (Approximation Bound)

Informal statement: There exists a choice of attention parameters, independent of n, d, and the input Z_0, such that the attention weights at layer ℓ approximate the regularized optimal transport solution P^*_λ with error

$$O\left(\frac{n^{1/2}e^{1/\lambda}}{\sqrt{\ell}}\right),$$

where $\lambda > 0$ is the regularization parameter.

The error bound provided above is in terms of the Hilbert projective metric, a well-established measure for analyzing the convergence behavior of the Sinkhorn algorithm [Franklin and Lorenz, 1989]. According to the theorem, the attention matrices converge to P^*_λ at a rate of $O\left(\frac{1}{\text{depth}^{1/2}}\right)$, implying that performance improves with increasing depth. This convergence holds for any integer n. Thus, a sufficiently deep transformer can approximate solution of OT for broad choice of n without changes in parameters. This result mathematically proves transformers are capable of seen generalization in Fig. 2.

An application of the last theorem is that transformers can be used to sort lists. As discussed, sorting is a specific instance of assignment problem for $d = 1$ with $y_1 < \cdots < y_n$. Thus, transformer can sort with an error that vanishes with λ. Graves et al. [2014] experimentally demonstrate that the neural Turing machine can sort. Here, we theoretically prove this capability for transformers by establishing an approximation bound.

Notably, the proof of the last theorem is constructive; it provides an explicit form of the parameters. This explicit characterization enables us to experimentally validate the result by directly instantiating the parameter choices. Fig. 3 shows that the estimate of the OT solution improves as the network depth increases, confirming the inverse dependence of the bound in the theorem on depth. Similarly, Fig. 4 demonstrates that a transformer with frozen weights can solve OT for varying values of n, explaining the out-of-distribution generalization observed in Fig. 2. In these experiments, we use POT library [Flamary et al., 2021] to estimate P^*_λ.

4 RELATED WORKS

Rank collapse of attention. The last theorem establishes that rank collapse in attention layers can be avoided. A substantial body of work has shown that attention matrices tend to become low-rank as network depth increases [Geshkovski et al., 2023, Wu et al., 2024, Dong et al., 2021] across a wide range of parameter choices. Specifically, Dong et al. [2021] demonstrate that attention layers lacking residual connections are particularly susceptible to this collapse, while Geshkovski et al. [2023] show that even with residual connections, attention with symmetric weights may still exhibit rank collapse. This degradation substantially limits expressive power and challenges training [Daneshmand et al., 2020], motivating a line of research into techniques that mitigate the collapse with normaliza-

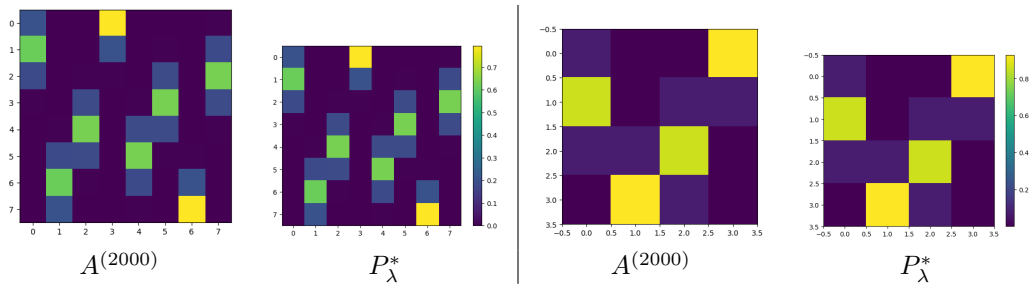


Figure 4: **Sample Size.** left: $n = 8$, right: $n = 4$. The transformer's weights remain unchanged. Observe the transformer can solve the OT problem for both values of n, demonstrating a form of out-of-distribution generalization proven in Thm. 3.1.

tion layers [Meterez et al., 2024, Daneshmand et al., 2021, Joudaki et al., 2023].Our last theorem demonstrates that rank collapse can be avoided through a specific prompt engineering combined with particular parameter choices. The key insight is that P^*_λ serves as an approximation of the full-rank permutation matrix P^*, ensuring that attention matrices maintain high rank across layers.

In-context Learning. Our study is motivated by a recent line of research exploring how transformers can perform regression using in-context samples Garg et al. [2022], Von Oswald et al. [2023], Akyürek et al. [2023], Ahn et al. [2023]. Garg et al. [2022] propose analyzing regression tasks encoded directly within the prompts of language models. Von Oswald et al. [2023] show that linear attention mechanisms can implement gradient descent on the mean squared error to solve such tasks. Building on this, Ahn et al. [2023] investigate how data distribution influences the in-context learning behavior of transformers. Their results suggest that data can guide linear attention to learn a wider class of algorithms, including preconditioning strategies for ill-conditioned problems. In this work, we extend these ideas to the assignment problem—a fundamental computation with applications in natural language processing.While pervious works mostly use a linear attention for analysis, here we study the mechanism of soft-max self attention which is widely used in practice.

Prompt Engineering Prompting language models is an art as a proper prompting can substantially enhance their response. For instance, using phrases like "let's think step by step" into prompts has been shown to help language models to produce more accurate solutions with logical reasoning [Kojima et al., 2022]. This insight has inspired a growing body of research aimed at automating prompting to enhance the performance of language models. Among the key directions are two prominent areas of focus: the automation of prompt engineering through reinforcement learning-based chain-of-thought prompting [Wei et al., 2022, Zhang et al., 2023, SU et al., 2023, Zhou et al., 2023], and prompt optimization techniques [Cheng et al., 2024, Pryzant et al., 2023, Rubin et al., 2022]. We study the essence of prompt engineering for in-context learning.

5 EXPERIMENTS

So far, we have provided a mechanistic analysis of token alignment in LLMs, showing that attention weights can iteratively align tokens across transformer layers. In this section, we investigate whether such token alignment also emerges in language translation.

Dataset. Experiments mostly are conducted on a subset of English–French sentence pairs from the WMT14 dataset [Bojar et al., 2014]. Specifically, we used the training split.

Implementation. Code is publicly available at the following GitHub repository: `https://github.com/hadidaneshmand/aistats26-incontextOT`. Our implementation used minimal assistance from GenAI, as documented in the repository, where only the toy-example in Fig. 1 is mainly implemented by GenAI. We additionally validated our outputs by comparing them with the GenAI implementation and obtained similar results. All experiments were run on a single NVIDIA RTX PRO 6000 Blackwell Workstation Edition. Reproducing the reported results is estimated to take less than one day.

5.1 Attention Alignment for Translation

According to Theorem 3.2, the attention weights should progressively improve token alignment across layers. We assess this emergence in the pretrained Marian model [Junczys-Dowmunt et al., 2018] and multilingual BERT model [Devlin et al., 2019].

Attention. For each transformer layer ℓ, the model outputs per-head self-attention matrices $\{A^{(\ell,k)}\}_{k=1}^{H}$

where H denotes the number of heads. We average across heads to obtain

$$A^{(\ell)} = \frac{1}{H}\sum_{k=1}^{H} A^{(\ell,k)} \in \mathbb{R}^{n\times n},$$

and then extract the French→English attention block when the input consists of concatenated English–French sentence pairs. Notably, the way this block is extracted depends on the model:

- In BERT, the attention matrix has size $(\#\text{all tokens})^2$, so we extract the corresponding off-diagonal block of the attention matrix.
- In Marian, the attention matrix has size $\#(\text{French tokens}) \times \#(\text{English tokens})$, since translation is performed in the decoder.

Since each word may be split into multiple tokens, we average the attention weights over the tokens corresponding to each word in order to convert token-level attention weights into word-level attention weights.

Evaluation Metric To obtain reliable ground-truth word correspondences, we use the MUSE English–French dictionary [Lample et al., 2017, Conneau et al., 2017a], which provides semantically equivalent translation pairs. To evaluate how effectively attention captures translation correspondences, we compute the total attention mass assigned to translated word pairs:

$$\text{corr}_\ell = \sum_{\text{word}} A^{(\ell)}(\text{word}, \text{translation of word}). \quad (8)$$

Fig. 5 shows that corr_ℓ increases with ℓ, indicating that deeper layers produce attention maps that align translated word pairs more accurately. This progressive improvement parallels Theorem 3.2 and the empirical observation in Fig. 1.

This observation also extends to the BERT model, with a slight difference. As shown in Fig. 6, attention iteratively allocates more weight to translated words across layers 1–5 and 7–12. This result, together with the observation in the next section, suggests that BERT solves OT twice across its layers.

5.2 Embedding OT and Translation

We show that OT connects with the mechanism of translation in language models. In particular, we substantiate translated-word alignment to OT, supporting the observation in Fig. 2 via a large-scale evaluation on 1000 sentence pairs from the WMT14 English–French translation dataset [Bojar et al., 2014]. To obtain

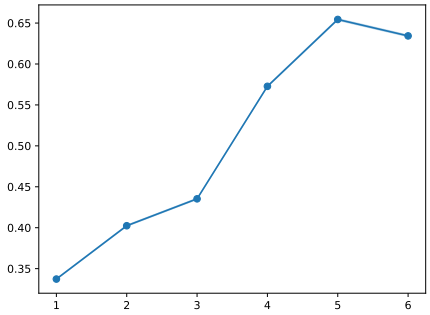


Figure 5: **Attention dynamics for translation.** x-axis: transformer layer index ℓ. y-axis: The metrics defined in (8); Increasing metric values indicate that attention weights across layers progressively provide better estimates of translated word alignments, closely resembling the attention dynamics for OT. Mean on 10000 sentences from WMT14 English–French translation dataset. The confidence interval is not visible due to small variance.

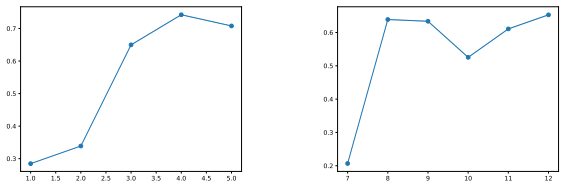


Figure 6: **Attention Dyanmics for BERT:** x-axis: transformer layer index ℓ. y-axis: The metric defined in (8). Observe: attention progressively allocates more weight to translated words across the layers 1-5 and 7-12. This result implies that attention may solve OT twice across the layers. Mean over 8,000 sentences is reported.

ground-truth word alignments, we use the MUSE bilingual dictionary [Lample et al., 2018, Conneau et al., 2017b], which provides type-level translation pairs. We then compare these ground-truth alignments with word matching produced by OT applied to word embeddings from the pretrained BERT model, computed by POT library [Flamary et al., 2021]. Fig. 7 reports the proportion of matched translated word pairs obtained from OT on word embeddings across BERT layers. We observe that the matching rate is consistently high, indicating that solving OT on word embeddings can effectively align words with their translations. This finding further supports the results in Fig. 1, suggesting that solving OT with attention layers (seen in the last subsection) contributes to the translation mechanism in pretrained large language models.

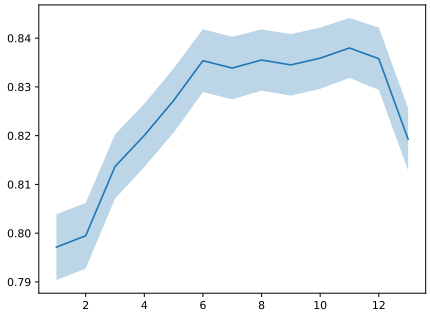


Figure 7: **OT and translation.** x-axis: layer index; y-axis: proportion of matched translated word pairs identified by applying OT to word embeddings at different BERT layers. This shows that OT on word embeddings can effectively recover translation correspondences between words. Mean and 80% confidence internal on 1000 sentences from WMT14 English–French translation dataset.

6 DISCUSSIONS

We studied the mechanism of token alignment, with optimal transport (OT), in attention layers. We experimentally link token alignment to language translation, showing that attention weights can iteratively match semantically equivalent words, iteratively across the layers. We then analyzed the underlying mechanism of OT in transformers, proving that attention layers can approximate the OT alignment up to an error that vanishes with depth.

Our findings open several promising avenues for future research. In particular, we outline four directions that can deepen our understanding of token alignment mechanisms in large language models.

Depth Efficient Guarantees. According to Theorem 3.2, a transformer with depth $O(\epsilon^{-2})$ can achieve an $O(\epsilon)$-accurate solution, following the established convergence rate for gradient descent with adaptive step sizes. However, only $O(\log(1/\epsilon))$ Sinkhorn iterations are needed for the same accuracy. We believe this gap arises from a loose convergence analysis, which can be refined in future work.

Learning with Small Prompts. We prove that a transformer with fixed parameters can solve OT for any arbitrary n. This striking generalization has practical benefits: it can drastically reduce both training time and memory usage for assignment tasks, since the transformer can be trained on prompts with a constant number of tokens. A natural question arises: what is the minimal value of n sufficient for the model to learn the assignment task?

Prompt Engineering. We theoretically and experimentally demonstrate that prompt engineering is essential for in-context assignment. However, the underlying mechanism of prompt engineering remains understudied in a broader context. Our findings motivate further study of prompt engineering from a computational perspective, highlighting its role in enhancing the computational expressivity of transformers.

Generalized Cost Functions. The OT problem can be formulated with a general cost function, beyond the standard Euclidean distance $\|x_i - y_j\|^2$. An important question is whether standard softmax attention can solve the assignment problem with a general cost function $C(x_i, y_j)$. We conjecture that a combination of attention layers and feedforward networks can approximate solutions in general cases.

Akhnowldgements

We thank Jacob Andreas for his helpful discussions and for inspiring the experiment in Fig. 2.2. We also thank Amir Reiszade for his insightful discussions on the related literature on in-context learning. We acknowledge partial funding from the NSF TRIPODS program (award DMS-2022448).

https://github.com/hadidaneshmand/aistats26-incontextOT

https://hadidaneshmand.github.io/papers/aistats26.html

References

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=LziniAXEI9`.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, Joao Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 35151–35174. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/von-oswald23a.html`.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598, 2022.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas,

Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wei Shen, Ruida Zhou, Jing Yang, and Cong Shen. On the training convergence of transformers for in-context classification. 2024.

Davoud Ataee Tarzanagh, Yingcong Li, Christos Thrampoulidis, and Samet Oymak. Transformers as support vector machines. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023. URL https://openreview.net/forum?id=gLwzzmh79K.

Patrick Lutz, Aditya Gangrade, Hadi Daneshmand, and Venkatesh Saligrama. Linear transformers implicitly discover unified numerical algorithms. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=XxR70zr9Sf.

Jiuqi Wang, Ethan Blaser, Hadi Daneshmand, and Shangtong Zhang. Transformers can learn temporal difference methods for in-context reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Pj06mxCXPl.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11 (5-6):355–607, 2019.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957. doi:10.1137/0105003. URL https://doi.org/10.1137/0105003.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. Marian: Fast neural machine translation in C++. In Fei Liu and Thamar Solorio, editors, *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi:10.18653/v1/P18-4020. URL https://aclanthology.org/P18-4020/.

Michael E Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*, pages 3515–3530. PMLR, 2022.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR, 2020.

Abhinav K Adduri, Dhruv Gautam, Beatrice Bevilacqua, Alishba Imran, Rohan Shah, Mohsen Naghipourfar, Noam Teyssier, Rajesh Ilango, Sanjay Nagaraj, Mingze Dong, et al. Predicting cellular responses to perturbation across diverse contexts with state. *BioRxiv*, pages 2025–06, 2025.

Roger W Brockett and Wing Shing Wong. A gradient flow for the assignment problem. In *New Trends in Systems Theory: Proceedings of the Università di Genova-The Ohio State University Joint Conference, July 9–11, 1990*, pages 170–177. Springer, 1991.

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

Richard Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Stanisław Jastrzebski, Devansh Arpit, Nicolas Ballas, Vikas Verma, Tong Che, and Yoshua Bengio. Residual connections encourage iterative inference. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJa9iHgAZ.

Guillaume Alain. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Joel Franklin and Jens Lorenz. On the scaling of multidimensional matrices. *Linear Algebra and its applications*, 114:717–735, 1989.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL `http://jmlr.org/papers/v22/20-451.html`.

Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. *Advances in Neural Information Processing Systems*, 36:57026–57037, 2023.

Xinyi Wu, Amir Ajorlou, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. On the role of attention masks and layernorm in transformers. *Advances in Neural Information Processing Systems*, 37:14774–14809, 2024.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International conference on machine learning*, pages 2793–2803. PMLR, 2021.

Hadi Daneshmand, Jonas Kohler, Francis Bach, Thomas Hofmann, and Aurelien Lucchi. Batch normalization provably avoids ranks collapse for randomly initialised deep networks. *Advances in Neural Information Processing Systems*, 33:18387–18398, 2020.

Alexandru Meterez, Amir Joudaki, Francesco Orabona, Alexander Immer, Gunnar Ratsch, and Hadi Daneshmand. Towards training without depth limits: Batch normalization without gradient explosion. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=xhCZD9hiiA`.

Hadi Daneshmand, Amir Joudaki, and Francis Bach. Batch normalization orthogonalizes representations in deep random networks. *Advances in Neural Information Processing Systems*, 34:4896–4906, 2021.

Amir Joudaki, Hadi Daneshmand, and Francis Bach. On the impact of activation and normalization in obtaining isometric embeddings at initialization. *Advances in Neural Information Processing Systems*, 36: 39855–39875, 2023.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 2022.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=5NTt8GFjUHkr`.

Hongjin SU, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=qY1hlv7gwg`.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=WZH7099tgfM`.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. Black-box prompt optimization: Aligning large language models without model training. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3201–3219, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.acl-long.176. URL `https://aclanthology.org/2024.acl-long.176/`.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL `https://openreview.net/forum?id=WRYhaSrThy`.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 2655–2671, 2022.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ale s Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W14/W14-3302`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi:10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423/`.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*, 2017.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017a.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=H196sainb`.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. 2017b.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 12 2014.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Yes]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

A Experimental settings

Training loss. We generate random data and train a transformer to solve the assignment task by minimizing

$$\min_{\theta} \mathbb{E} \left\| f_\theta \left(\begin{bmatrix} x_1 & y_1 & \blacksquare \\ x_2 & y_2 & \blacksquare \\ \vdots & \vdots & \blacksquare \\ x_n & y_n & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \end{bmatrix} \right) - P_* \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \right\|^2, \tag{9}$$

where the expectation is taken over randomly generated inputs $x_1, \ldots, x_n$, and the ■ marks engineered part of the input in (6). $f_\theta : \mathbb{R}^{(n+1)\times(2d+9)} \to \mathbb{R}^n$ is the output function of a transformer with parameters θ. To generate outputs, we use an attention layer defined as

$$f_\theta(Z_0) = [\text{atten}_\theta([Z_\ell]_{1:n,:})]_{:,0:d} \tag{10}$$

where $[Z_\ell]_{1:n,:}$ denotes the first n rows of Z_ℓ in (7). The above indexing allows us to generate the outputs of size $n \times d$.

Training data. The expectation in the training loss (9) is taken over 1000 random samples generated with $n = 7$. $x_1, \ldots, x_n$ are a random permutation of $[1/n, 2/n, \ldots, n/n]$, and we set $y_i = i/n$ in our experiments.

Optimization. For training, we run 10^4 iterations of Adam [Kingma and Ba, 2014] with stepsize 0.001. Parameters are initialized from a Gaussian distribution with covariance matrix $1/(2d+9)$. We set $B_j^{(\ell)} = (1/20) I_{2d+9}$ where I_k is the square identity matrix of size k. We optimize the attention parameters $\theta := \{[K^{(m,j)}, Q^{(m,j)}, V^{(m,j)}]\}_{m=1}^{\ell}$ using the reparameterization $P^{(\ell,j)} = K^{(\ell,j)} Q^{(\ell,j)}$ and optimize $P^{(\ell,j)}$.

Without prompt engineering. To study the impact of prompt engineering in Fig. 2 Observation (2), we remove additional columns in the engineered prompt as

$$Z' = \begin{bmatrix} x_1 & y_1 & 0 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 0 \end{bmatrix} \in \mathbb{R}^{n\times 3}. \tag{11}$$

We trained the model on exactly the same training data with the same optimization settings.

B The Aligning Transformer

The theoretical and experimental results presented are based on a specific choice of parameters for the attention layers. These parameters are used both to generate the plots in figures 3 and 4 and also used to prove the main theorems.

Recall that the hidden representations in the transformer obeys

$$Z_{\ell+1} = Z_\ell + \sum_{j=1}^{2} \text{atten}_{\theta_j^{(\ell)}}(Z_\ell) B_j^{(\ell)},$$

$\lambda P^{(\ell,1)}$	$\begin{bmatrix}\begin{bmatrix}0_{d\times d}\\0_{p\times d}\end{bmatrix} & 2\begin{bmatrix}I_{d\times d}\\0_{p\times d}\end{bmatrix} & \begin{bmatrix}0_d\\0_p\end{bmatrix} & \begin{bmatrix}0_d\\-e^{(p)}_{d+3}\end{bmatrix} & \begin{bmatrix}0_d\\-e^{(p)}_{d+1}\end{bmatrix} & \begin{bmatrix}0_d\\e^{(p)}_{d+7}\end{bmatrix} & \begin{bmatrix}0_d\\-\lambda e^{(p)}_{d+5}\end{bmatrix} & \begin{bmatrix}0_{d\times 2}\\0^{(p)}_{p\times 2}\end{bmatrix} & \begin{bmatrix}0_d\\e^{(p)}_{2d+5}\end{bmatrix} & \begin{bmatrix}0_d\\0_p\end{bmatrix}\end{bmatrix}$
$\lambda P^{(\ell,2)}$	$\begin{bmatrix}2\begin{bmatrix}0_{d\times d}\\I_{d\times d}\\0_{9\times d}\end{bmatrix} & \begin{bmatrix}0_{d\times d}\\0_{d\times d}\\0_{9\times d}\end{bmatrix} & \begin{bmatrix}0_d\\I_d\\-e^{(9)}_3\end{bmatrix} & \begin{bmatrix}0_d\\0_d\\0_3\end{bmatrix} & \begin{bmatrix}0_d\\I_d\\-e^{(9)}_2\end{bmatrix} & \begin{bmatrix}0_d\\I_d\\e^{(9)}_8\end{bmatrix} & \begin{bmatrix}0_d\\I_d\\-\lambda e^{(9)}_5\end{bmatrix} & \begin{bmatrix}0_d\\0_d\\0_3\end{bmatrix} & \begin{bmatrix}0_d\\I_d\\e^{(9)}_5\end{bmatrix} & \begin{bmatrix}0_{d\times 2}\\0_{d\times 2}\\0_{3\times 2}\end{bmatrix}\end{bmatrix}$
$[V^{(\ell,1)}]_{ij}$	$= \begin{cases}1 & i = 2d+6 \text{ and } j = 2d+7\\ 0 & \text{otherwise}\end{cases}$
$[V^{(\ell,2)}]_{ij}$	$= \begin{cases}1 & i = 2d+6 \text{ and } j = 2d+8\\ 0 & \text{otherwise}\end{cases}$
$B^{(\ell)}_j$	$\gamma_\ell I_{(2d+9)\times(2d+9)}$

Table 2: **Parameters for OT:** Our proof is constructive since it specifies the choice of parameters.

where $\{\theta^{(\ell)}_j\}$ denotes the parameters of attention head j at layer ℓ containing three matrices as $\theta^{(\ell)}_j = \{K^{(\ell,j)}, Q^{(\ell,j)}, V^{(\ell,j)} \in \mathbb{R}^{d\times d}\}$.

We define $P^{(\ell,j)} = K^{(\ell,j)}(Q^{(\ell,j)})^\top$. Let $p = d+9$, and let $e^{(k)}_i \in \mathbb{R}^k$ denote the i-th standard basis vector, defined by $[e_i]_j = \begin{cases}1 & \text{if } i = j,\\ 0 & \text{otherwise}\end{cases}$. Based on these definitions, Table 2 summarizes the choice of parameters for OT.

C Proof of Theorem 3.1

We demonstrate that two attention heads can jointly implement a single step of gradient descent (with adaptive step sizes) on $L(u,v)$. By induction, multiple attention heads can implement several iterations of gradient descent with adaptive step sizes. The proof is constructive, explicitly determining the choice of parameters specified in Section B.

Recall $u_\ell, v_\ell \in \mathbb{R}^n$ are iterations of gradient descent (with adaptive coordinate-wise stepsize) on Lagrangian function L defined in (5):

$$\begin{cases}u_{\ell+1} = u_\ell - D_\ell \nabla_u L(u_\ell, v_\ell)\\ v_{\ell+1} = v_\ell - D'_\ell \nabla_v L(u_\ell, v_\ell)\end{cases}, L(u,v) := \lambda\left(\sum_{ij} e^{(-C_{ij}+u_i+v_j)/\lambda - 1}\right) - \sum v_i - \sum_i u_i$$

where the coordinate-wise stepsizes are

$$(\gamma_\ell [D_\ell]_{ii})^{-1} = \sum_j e^{(-C_{ij}+[u_\ell]_i+[v_\ell]_j)/\lambda - 1} + 1, \quad (\gamma_\ell [D'_\ell]_{jj})^{-1} = \sum_i e^{(-C_{ij}+[u_\ell]_i+[v_\ell]_j)/\lambda - 1} + 1.$$

Define $x = [x_1, \ldots, x_n], y = [y_1, \ldots, y_n] \in \mathbb{R}^{n\times d}$ and $x^2 = [\|x_1\|^2, \ldots, \|x_n\|^2], y^2 = [\|y_1\|^2, \ldots, \|y_n\|^2] \in \mathbb{R}^n$. Let $H^{(\ell)} \in \mathbb{R}^{n\times n}$ obeys

$$H^{(\ell)} = \tfrac{1}{\lambda}\left(-x^2 1_n^\top + 2xy^\top - 1_n (y^2)^\top + u_\ell 1_n^\top + 1_n (v_\ell)^\top\right) - 1_n 1_n^\top \tag{12}$$

Define $M^{(\ell)} \in \mathbb{R}^{n\times n}$ defines as $M_{ij}^{(\ell)} = e^{H_{ij}^{(\ell)}}$. Gradient descent thus follows:

$$u_{\ell+1} = u_\ell - D_\ell(M^{(\ell)}1_n - 1_n) \tag{13}$$
$$v_{\ell+1} = v_\ell - D'_\ell((M^{(\ell)})^\top 1_n - 1_n) \tag{14}$$

Induction statement. We assume the statement holds for ℓ and then prove it for $\ell+1$. Thus, the induction hypothesis is

$$Z_\ell = \begin{bmatrix} x_1 & y_1 & \|x_1\|^2 & \|y_1\|^2 & 1_4 & [u_\ell]_1 & [v_\ell]_1 & 0 \\ x_2 & y_2 & \|x_2\|^2 & \|y_2\|^2 & 1_4 & [u_\ell]_2 & [v_\ell]_2 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & \|x_n\|^2 & \|y_n\|^2 & 1_4 & [u_\ell]_d & [v_\ell]_d & 0 \\ 0 & 0 & 0 & 0 & -v_4 & ? & ? & 0 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(2d+9)},$$

where 1_4 denotes the all-ones 4-dimensional vector, $v_4 = [0,0,0,1]$, and $[u]_i$ denotes element i of vector u. It is easy to check that the above equation holds for $\ell = 0$ as $u_0 = v_0 = 0_n$. The choice of $w_v^{(\ell,j)}$ ensure that only the $2d+7$-th and $2d+8$-th columns of Z_ℓ change with ℓ, which are highlighted in . We prove that

$$Z_{\ell+1} = \begin{bmatrix} x_1 & y_1 & \|x_1\|^2 & \|y_1\|^2 & 1_4 & [u_{\ell+1}]_1 & [v_{\ell+1}]_1 & 0 \\ x_2 & y_2 & \|x_2\|^2 & \|y_2\|^2 & 1_4 & [u_{\ell+1}]_2 & [v_{\ell+1}]_2 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & \|x_n\|^2 & \|y_n\|^2 & 1_4 & [u_{\ell+1}]_d & [v_{\ell+1}]_d & 0 \\ 0 & 0 & 0 & 0 & -v_4 & ? & ? & 0 \end{bmatrix} \in \mathbb{R}^{(n+1)\times(2d+9)},$$

Indeed, the extended prompt offers sufficient memory to store the iterates of gradient descent.

Induction proof. We begin by computing the output of the first attention head in layer $\ell+1$, step by step. Through straightforward algebra, we obtain the following:

$$Z_\ell P^{(\ell,1)} = \begin{bmatrix} 0 & 2x/\lambda & 0_n & -1_n/\lambda & -\|x\|^2/\lambda & u_\ell/\lambda & -1_n & 0_{2\times n} & 1_n/\lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0_{2\times 1} & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(n+1)\times d'}$$

where matrices x and y are defined earlier in the proof. With these notations and the preceding equation, we proceed as follows:

$$Z_\ell P^{(\ell,1)} Z_\ell^\top = \begin{bmatrix} H^{(\ell)} & 0_n \\ 0_n^\top & 0 \end{bmatrix}$$

which obtains

$$\exp(Z_\ell P^{(\ell,1)} Z_\ell^\top) = \begin{bmatrix} M^{(\ell)} & 1_n \\ 1_n^\top & 1 \end{bmatrix} \tag{15}$$

Furthermore, the choice of parameters $V^{(\ell,1)}$ obtains

$$Z_\ell V^{(\ell,1)} = -\gamma \begin{bmatrix} 0_n & \dots & 0_n & 1_n & 0_n & 0_n \\ 0 & \dots & 0 & -1 & 0 & 0 \end{bmatrix}$$

Stitching all equations together yields

$$\text{atten}_{\theta_1^{(\ell)}}(Z_\ell)B_1^{(\ell)} = \begin{bmatrix} 0_n & \dots & 0_n & -D_\ell(M^{(\ell)}1_n - 1_n) & 0_n & 0_n \\ 0 & \vdots & 0 & n-1 & 0 & 0 \end{bmatrix}$$

Observe that the matrix above contains the gradient $\nabla_u L(u_\ell, v_\ell)$ from Eq. (14), which is required to compute the next gradient descent iterate $u_{\ell+1}$. Similarly, we can show that

$$\text{atten}_{\theta_2^{(\ell)}}(Z_\ell)B_2^{(\ell)} = \begin{bmatrix} 0_n & \dots & 0_n & 0_n & -D'_\ell((M^{(\ell)})^\top 1_n - 1_n) & 0_n \\ 0 & \vdots & 0 & 0 & n-1 & 0 \end{bmatrix},$$

which computes $v_{\ell+1}$ in parallel using a second attention head. Thus, substituting the last two equations into (7) concludes the induction proof.

D Theorem 3.2

D.1 Formal statement

Attention matrices can approximate the entropy-regularized assignment solution, denoted by P^*_λ, as defined in (3). Consider the block of *attention matrices* $A^{(\ell)} \in \mathbb{R}^{n\times n}$, defined by

$$A^{(\ell)}_{ij} = \frac{e^{\langle K^{(\ell,1)} z^{(\ell)}_i, Q^{(\ell,1)} z^{(\ell)}_j\rangle}}{\sum_{j=1}^n e^{\langle K^{(\ell,1)} z^{(\ell)}_i, Q^{(\ell,1)} z^{(\ell)}_j\rangle}} \tag{16}$$

where $z^{(\ell)}_i$ is the i-th row of Z_ℓ, representing the token embeddings at layer ℓ, and $Q^{(\ell,1)}$, $K^{(\ell,1)}$ are the query and key weight matrices of an attention head at layer ℓ.

We prove the attention matrix $A^{(\ell)}$ converges to P^*_λ in an appropriate metric for certain choice of parameters of attention layers. As discussed, solution P^*_λ in (3) can be written as

$$\begin{aligned} P^*_\lambda &= \mathrm{diag}(p^*) Q \mathrm{diag}(q^*) \\ & p^*, q^* \in \mathbb{R}^n_+, \quad Q \in \mathbb{R}^{n\times n}_+, \end{aligned} \tag{17}$$

where $\mathrm{diag}(v)$ is a diagonal matrix whose diagonal element in row i is v_i, $\quad Q_{ij} = e^{-\frac{C_{ij}}{\lambda}-1}$, $p^*_i = e^{v^*_i/\lambda}$ and $q^*_j = e^{u^*_j/\lambda}$. It is easy to check that replacing p^* and q^* with cp^* and q^*/c leads to the same matrix P^*_λ for all $c \in \mathbb{R}_+$. Franklin and Lorenz [1989] introduce a metric that accounts for this particular scaling invariance:

$$\mu(q, q') = \log\left(\max_{ij} \frac{q_i q'_j}{q_j q'_i}\right). \tag{18}$$

Remarkably, μ is a metric that satisfies the triangle inequality [Franklin and Lorenz, 1989]. However, μ is not a norm, as $\mu(q, q') = 0$ only implies that there exists a constant c such that $q = cq'$. The next theorem establishes an explicit convergence rate for the attention matrices $A^{(\ell)}$ to P^*_λ in μ.

Theorem D.1 (Formal version of Thm. 3.2: Convergence to P^*_λ in the Hilbert projective metric)**.** Let $C \in \mathbb{R}^{n\times n}$ be a cost matrix with entries C_{ij}, and fix $\lambda > 0$. Define

$$Q \;\in\; \mathbb{R}^{n\times n}_+, \qquad Q_{ij} \;=\; e^{-\,C_{ij}/\lambda - 1}.$$

Let $P^*_\lambda \in \mathbb{R}^{n\times n}_+$ denote the entropy-regularized optimal transport plan, which admits the factorization

$$P^*_\lambda \;=\; \mathrm{diag}(p^*)\, Q\, \mathrm{diag}(q^*) \quad \text{for some } p^*, q^* \in \mathbb{R}^n_+,$$

unique up to positive rescaling of p^*, q^*. Consider a transformer with a fixed choice of parameters (independent of n, d, and the input) as constructed in Appendix B, and let $A^{(\ell)} \in \mathbb{R}^{n\times n}_+$ denote the attention matrix (for a specified head/block) at depth ℓ. For each $\ell \geq 1$ there exist $p_\ell, q_\ell \in \mathbb{R}^n_+$ such that

$$A^{(\ell)} \;=\; \mathrm{diag}(p_\ell)\, Q\, \mathrm{diag}(q_\ell).$$

Define the Hilbert projective metric

$$\mu(u, v) \;=\; \log\!\left(\max_{i,j} \frac{u_i v_j}{u_j v_i}\right), \qquad u, v \in \mathbb{R}^n_+,$$

and let

$$\phi(Q) \;=\; \max_{i,j,k,\ell} \frac{Q_{ik}\, Q_{j\ell}}{Q_{jk}\, Q_{i\ell}}, \qquad \eta \;=\; \frac{\sqrt{\phi(Q)} - 1}{\sqrt{\phi(Q)} + 1} \in [0, 1).$$

Set

$$\frac{1}{4}\, r^2 \;=\; \|p_1 - p^*\|_2^2 \;+\; \|q_1 - q^*\|_2^2.$$

Then, provided $\ell \;\geq\; 64\, n\, e^{3r/\lambda}\, r$, there exists an index $k \leq \ell$ such that

$$\min_{k\leq \ell} \max\{\, \mu(p_k, p^*),\; \mu(q_k, q^*)\,\} \;\leq\; \frac{36\sqrt{n}\, e^{1.5r/\lambda}}{\sqrt{\ell}\,(1-\eta)}.$$

Remark D.2. We improved the bound in Thm. 3.2 from $O\big(\frac{n^{3/2}}{\ell^{1/2}}\big) \to O\big(\sqrt{\frac{n}{\ell}}\big)$ while preparing the supplementary material.

Remark D.3 (On scaling, metric choice, and non-monotonicity). (i) The factorization of P_λ^* is invariant to the rescaling $(p^*, q^*) \mapsto (c\,p^*, c^{-1}q^*)$, $c > 0$; the metric μ is designed to quotient out this invariance. (ii) The "$\min_{k\le\ell}$" accounts for possible non-monotone progress across layers; the bound holds for the best iterate up to depth ℓ. (iii) The constant r encodes the initialization distance to (p^*, q^*), and the quantity η is the classical Sinkhorn contraction factor [Franklin and Lorenz, 1989].

D.2 Proof

Challenge: According to Thm. 3.1, a transformer can implement gradient descent. Therefore, the proof casts to analyzing gradient descent (with specific coordinate-wise stepsizes) on the convex L. However, we cannot directly apply existing convergence results from convex optimization. The existing convergence results for smooth convex optimization are in terms of function value L when L is not strongly convex[1]. But, the theorem statement aims at the convergence to a minimizer.

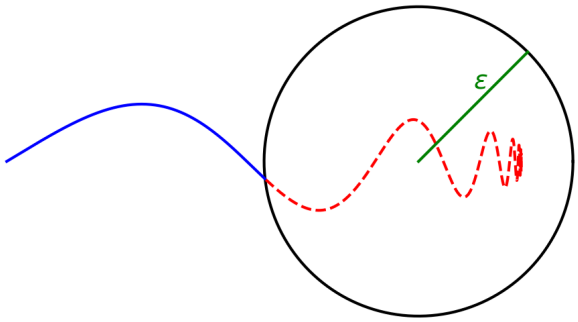


Figure 8: **Proof techinque for Theorem 3.2.** We first prove that the attention matrix converges to a local neighborhood of the set of doubly stochastic matrices. This convergence is illustrated by the blue curves converging to a small circle. Next, we show that this convergence implies convergence to the minimizer. To establish this, we hypothetically run Sinkhorn's iterations and leverage their known convergence rate. The red curve illustrates these hypothetical Sinkhorn iterations.

Proof Sketch. The proof consists of two key steps: (i) the convergence of attention matrix $A^{(\ell)}$ to a matrix that is approximately doubly stochastic, and (ii) a hypothetical simulation of Sinkhorn's recurrence. See Figure 8 for the illustration.

(i) As shown in Thm. 3.1, the transformer can perform gradient descent with an adaptive step size on the convex function L. Since L is convex, gradient descent is guaranteed to converge to a stationary point where the gradient norm becomes zero. Specifically, it is straightforward to verify that $\nabla_u L = M\mathbf{1} - \mathbf{1}$ and $\nabla_v L = M^\top \mathbf{1} - \mathbf{1}$, where $M_{ij} = \exp\left(\frac{-C_{ij}+u_i+v_j}{\lambda} - 1\right)$. Therefore, small gradients for u and v imply that M is close to being doubly stochastic.

(ii) We demonstrate that when the matrix M is approximately doubly stochastic, it is near the desired solution P_λ^*. To establish this, we (hypothetically) run Sinkhorn's recurrence starting from M and use its contraction property proven by [Franklin and Lorenz, 1989].

Before elaborating on the details of (i) and (ii), we present two propositions.

Preliminaries. Define the functions $row : \mathbb{R}_+^{n\times n} \to \mathbb{R}_+^n$ and $col : \mathbb{R}_+^{n\times n} \to \mathbb{R}_+^n$ as

$$\mathrm{row}(A)_i = \frac{1}{\sum_j A_{ij}}, \quad \mathrm{col}(A)_j = \frac{1}{\sum_i A_{ij}}.$$

[1]It is easy to check that L is not strongly convex since it Hessian has a zero eigenvalue.

We also introduce functions $f, g : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ defined as

$$f(A) = A\text{diag}(\text{col}(A)), \quad g(A) = \text{diag}(\text{row}(A))A.$$

Indeed, $f(A)$ (resp. $g(A)$) normalizes the columns (resp. rows) of A by a scaling factor of their average. We will later use f and g to formulate Sinkhorn's recurrence, which iteratively normalizes the rows and columns of a matrix with positive elements. The next proposition proves that an almost doubly stochastic matrix remains almost doubly stochastic under f and g. To formulate the statement, we introduce a set containing matrices that almost doubly stochastic matrices:

$$\mathcal{S}_\epsilon := \left\{A \in \mathbb{R}_+^{n\times n} \mid \|A1_n - 1_n\|_\infty \le \epsilon \text{ and } \|A^\top 1_n - 1_n\|_\infty \le \epsilon\right\}.$$

Proposition D.4. Suppose that $A \in \mathcal{S}_\epsilon$; then $f(A) \in \mathcal{S}_{3\epsilon}$ and $g(A) \in \mathcal{S}_{3\epsilon}$, as long as $\epsilon < 1/3$.

Recall the metric d defined in (18). The next proposition establishes a particular property of f and g with respect to d.

Proposition D.5. Let $A \in \mathcal{S}_\epsilon$ be decomposed as $A = \text{diag}(w)Q\text{diag}(q)$, where $w, q \in \mathbb{R}_+^n$.

(i) For $f(A) = \text{diag}(w)Q\text{diag}(q')$, $\mu(q, q') \le 4\epsilon$ holds for $\epsilon < \frac{1}{4}$.

(ii) For $g(A) = \text{diag}(w')Q\text{diag}(q)$, $\mu(w, w') \le 4\epsilon$ holds for $\epsilon < \frac{1}{4}$.

Convergence Analysis. According to Theorem 3.1, there is a choice of parameters such that

$$A_{ij}^{(\ell)} = e^{\frac{-C_{ij}+[u_\ell]_i+[v_\ell]_j}{\lambda}-1},$$

where u_ℓ and v_ℓ are the iterates defined in (5). The following lemma establishes that, as the number of iterations ℓ increases, $A^{(\ell)}$ meets a neighborhood of doubly stochastic matrices.

Lemma D.6. For $\gamma_k^{-1} = (n+2)e^{2r/\lambda}$, there exists a $k \le \ell$ such that

$$A^{(k)} \in \mathcal{S}_\epsilon, \quad \text{where } \epsilon^2 := \left(\frac{1}{\ell}\right) 3ne^{3r/\lambda}r.$$

Notably, the matrix $A^{(k)}$ has a specific structure that ensures $A^{(k)} \in \mathcal{S}_\epsilon$ is sufficient to approximate P_λ^*. To prove this statement, we leverage the contraction property of Sinkhorn's recurrence.

Contractive Sinkhorn's Process. According to the last lemma, there exists an iteration $k \le \ell$ such that $A^{(k)} \in \mathcal{S}_\epsilon$. We then apply Sinkhorn's recurrence starting from $A_1 = g(A^{(k)})$ as:

$$A_{m+1/2} = f(A_m), \quad A_{m+1} = g(A_{m+1/2}).$$

Notably, we utilize the above recurrence solely for the proof; hence, there is no need for a transformer to implement this recurrence. According to the definition, A_m can be decomposed as $\text{diag}(w_m)Q\text{diag}(q_m)$, where $Q_{ij} = e^{-C_{ij}/\lambda-1}$ and $q_m, w_m \in \mathbb{R}_+^n$. Sinkhorn [1967] proves that there exist unique vectors $w^*, q^* \in \mathbb{R}_+^n$ such that $P_\lambda^* = \text{diag}(w^*)Q\text{diag}(q^*)$, where $w_i^* = e^{u_i^*/\lambda}$ and $q_j^* = e^{v_j^*/\lambda}$. [Franklin and Lorenz, 1989] establish the linear convergence of (w_m, q_m) to (w^*, q^*):

$$\begin{cases}\mu(w_{m+1}, w^*) \le \eta\mu(w_m, w^*) \\ \mu(q_{m+1}, q^*) \le \eta\mu(q_m, q^*)\end{cases}, \quad \eta = \frac{\phi(A_1)^{1/2}-1}{\phi(A_1)^{1/2}+1} < 1, \tag{19}$$

where $\phi(A) = \max_{ijkl} \frac{A_{ik}A_{jl}}{A_{jk}A_{il}}$. Since $A_1 = \text{diag}(w_1)Q\text{diag}(q_1)$, we have $\phi(A_1) = \phi(Q)$.

Propositions D.4 and D.5 enable us to demonstrate that there exists a constant c such that cA_1 lies within a neighborhood of P_λ^*. Proposition D.4 combined with Lemma D.6 ensure $A_1 \in \mathcal{S}_{3\epsilon}$. Thus, we can apply Proposition D.5 to obtain: $\mu(q_2, q_1) \le 12\epsilon$. Using Proposition D.4 again, we find that $A_{1+1/2} \in \mathcal{S}_{9\epsilon}$. Consequently, we can invoke Proposition D.5 once more to yield: $\mu(w_2, w_1) \le 36\epsilon$. Applying the triangle inequality together with 19 completes the proof:

$$36\epsilon \ge \mu(w_2, w_1) \ge \mu(w_1, w^*) - \mu(w_2, w^*) \ge (1-\eta)\mu(w_1, w^*)$$
$$12\epsilon \ge \mu(q_2, q_1) \ge \mu(q_1, q^*) - \mu(q_2, q^*) \ge (1-\eta)\mu(q_1, q^*).$$

E Proof of Lemma D.6

Notations. Define the concatenated vector of iterates as

$$\theta_k = \begin{bmatrix} u_k \\ v_k \end{bmatrix},$$

and consider the following block diagonal matrix:

$$\Lambda_k = \begin{bmatrix} D_k & 0 \\ 0 & D'_k \end{bmatrix},$$

where D_k and D'_k are diagonal matrices at iteration k defined in Theorem 3.1. Define the ball $B(r) = \{\theta \in \mathbb{R}^n \mid \|\theta\| \leq r\}$. If $\theta_k \in B(r)$, then

$$\frac{\gamma_k}{n\exp(r/\lambda)+1} I_n \preceq \Lambda_k \preceq \gamma_k I_n. \tag{20}$$

Smoothness of L. The Hessian of L has the following form

$$\nabla^2 L := \begin{bmatrix} \nabla^2_{uu} L & \nabla^2_{uv} L \\ \nabla^2_{vu} L & \nabla^2_{vv} L \end{bmatrix} = \begin{bmatrix} \mathrm{diag}(\sum_i M_{ij}) & M \\ M^\top & \mathrm{diag}(\sum_j M_{ij}) \end{bmatrix} \tag{21}$$

We will prove that the Hessian bounded within the domain $\theta \in B(r)$. For all $v := \begin{bmatrix} s \in \mathbb{R}^n \\ s' \in \mathbb{R}^n \end{bmatrix}$ such that $\|v\|^2 = 1$, we have

$$v^\top \nabla^2 L v = \|s\|^2_{\mathrm{diag}(\sum_i M_{ij})} + 2 s^\top M s' + \|s'\|^2_{\mathrm{diag}(\sum_j M_{ij})} \tag{22}$$

where $\|v\|^2_A = v^\top A v$. Recall $M_{ij} = e^{\frac{-C_{ij}+u_i+v_j}{\lambda}-1}$, hence

$$\sum_{ij} s_i s'_j M_{ij} = \sum_{ij} s_i e^{u_i/\lambda} s_j e^{v_j/\lambda} e^{-C_{ij}/\lambda - 1} \tag{23}$$

$$\leq \sum_{ij} s_i e^{u_i/\lambda} s_j e^{v_j/\lambda} \tag{24}$$

$$\leq \sqrt{\sum_i s_i^2 e^{2u_i/\lambda}} \sqrt{\sum_i (s'_i)^2 e^{2v_i/\lambda}} \tag{25}$$

$$\leq e^{2r/\lambda} \tag{26}$$

It is easy to check that $\|\mathrm{diag}(\sum_i M_{ij})\|$ and $\|\mathrm{diag}(\sum_j M_{ij})\|$ are bounded by $ne^{r/\lambda}$. Replacing these inequalities into (22) yields

$$v^\top \nabla^2 L v \leq n e^{r/\lambda} \left(\underbrace{\|s\|^2 + \|s'\|^2}_{=1} \right) + 2e^{2r/\lambda} \leq (n+2) e^{2r/\lambda}. \tag{27}$$

Thus, $L(\theta)$ is ζ-smooth for $\zeta := (n+2)e^{2r/\lambda}$ when $\theta \in B(r)$.

Boundedness of iterates. The recurrence relation of the iterates defined in (5) leads to the following inequality:

$$\|\theta_{k+1} - \theta^*\|^2_{\Lambda_k^{-1}} = \|\theta_k - \theta^*\|^2_{\Lambda_k^{-1}} - 2\langle \theta_k - \theta^*, \nabla L(\theta_k)\rangle + \|\nabla L(\theta_k)\|^2_{\Lambda_k}, \tag{28}$$

recall $\|v\|^2_A = v^\top A v$. Since L is ζ-smooth within $B(r)$, by Theorem 2.1.5 of Nesterov [2013], we have

$$\langle \nabla L(\theta), \theta - \theta^* \rangle \geq \frac{1}{\zeta} \|\nabla L(\theta)\|^2. \tag{29}$$

Substituting the above inequality into (28) yields

$$\|\theta_{k+1}-\theta^*\|^2_{\Lambda_k^{-1}} \leq \|\theta_k-\theta^*\|^2_{\Lambda_k^{-1}} - \left(\frac{2}{\zeta}-\gamma_k\right)\|\nabla L(\theta_k)\|^2. \tag{30}$$

Let $\Delta_k := \|\theta_k-\theta^*\|^2_{\Lambda_k^{-1}}$. For $\gamma_k = \frac{1}{\zeta}$, the above inequality ensures that Δ_k is monotonically decreasing:

$$\Delta_{k+1} \leq \Delta_k - \left(\frac{2}{\zeta}-\gamma_k\right)\|\nabla L(\theta_k)\|^2 \leq \Delta_k.$$

To maintain $\theta_k \in B(r)$ for all k, choose r such that

$$\|\theta_k\| \leq \Delta_k + \|\theta^*\|_{\Lambda_k^{-1}} \leq \|\Delta_1\|_{\Lambda_1^{-1}} + \|\theta^*\|_{\Lambda_1^{-1}} \leq 2\left(\|\theta_1-\theta^*\| + \|\theta^*\|\right) = r.$$

We now show that $\theta_k \in B(r)$ concludes the proof.

Convergence to a stationary point. Since $\theta_k \in B(r)$, we can take the average of (30) over $k = 1, \ldots, \ell$:

$$\sum_{k=1}^{\ell}\|\nabla L(\theta_k)\|^2 \leq \zeta\left(\sum_{k=1}^{\ell}\Delta_k - \Delta_{\ell+1}\right) \leq \zeta\Delta_1 \leq \zeta\left(ne^{r/\lambda}+1\right)r.$$

The above inequality leads to the following bound on the minimum gradient norm:

$$\min_{k\leq\ell}\|\nabla L(\theta_k)\|^2 \leq \tfrac{1}{\ell}\sum_{k=1}^{\ell}\|\nabla L(\theta_k)\|^2 \leq \left(\frac{1}{\ell}\right)\zeta(ne^{r/\lambda}+1)r. \tag{31}$$

Closeness to Doubly Stochastic Matrices. By definition,

$$\nabla L(\theta_k) = \begin{bmatrix} M^{(k)}1_n - 1_n \\ (M^{(k)})^\top 1_n - 1_n \end{bmatrix}, \tag{32}$$

where $\mathbf{1}$ denotes the vector of all ones. Substituting the expression for $\nabla L(\theta_k)$ into (31) gives

$$\min_{k\leq\ell}\left(\|M^{(k)}1_n - 1_n\|^2 + \|(M^{(k)})^\top 1_n - 1_n\|^2\right) \leq \frac{\zeta(n\exp(r/\lambda)+1)r}{\ell}. \tag{33}$$

F Proof of Proposition D.4

We prove $f(A) \in \mathcal{S}_{3\epsilon}$ and the proof for $g(A) \in \mathcal{S}_{3\epsilon}$ follows exactly the same. Since $A \in \mathcal{S}_\epsilon$, the following two inequalities hold

$$\left|\sum_i A_{ij} - 1\right| \leq \epsilon \implies \sum_i A_{ij} \geq 1 - \left|1 - \sum_i A_{ij}\right| \geq 1-\epsilon \tag{34}$$

Using the above two inequalities, we proceed as

$$\begin{aligned} |\tfrac{A_{ij}}{\sum_i A_{ij}} - A_{ij}| &= A_{ij}|1 - \tfrac{1}{\sum_i A_{ij}}| && (35)\\ &\leq \tfrac{A_{ij}\epsilon}{\sum_i A_{ij}} && (36)\\ &\leq \tfrac{A_{ij}\epsilon}{1-\epsilon}. && (37)\end{aligned}$$

We use the above inequality to complete the proof:

$$\begin{aligned} \left|\sum_j \tfrac{A_{ij}}{\sum_i A_{ij}} - 1\right| &\leq \left|\sum_j \tfrac{A_{ij}}{\sum_i A_{ij}} - \sum_j A_{ij}\right| + |\sum_j A_{ij} - 1| && (38)\\ &\leq \sum_j |\tfrac{A_{ij}}{\sum_i A_{ij}} - A_{ij}| + \epsilon && (39)\\ &\leq \tfrac{\epsilon}{1-\epsilon}\sum_j A_{ij} + \epsilon && (40)\\ &\leq \epsilon\left(1 + \tfrac{1+\epsilon}{1-\epsilon}\right) && (41)\end{aligned}$$

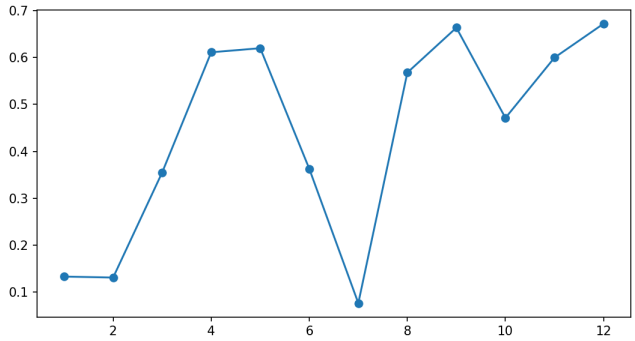


Figure 9: **Attention vs. OT Correlation.** The vertical axis reports the correlation between the attention heatmap and the OT solution across layers of the multilingual BERT model. The horizontal axis indexes the layer. We observe that attention patterns increasingly align with the OT plan in deeper layers, yielding a better approximation of the OT solution computed from word embeddings. This pattern is observable in Figure 1

G Proof of Proposition D.5

We prove part (i), and the proof for part (ii) follows exactly the same. The following inequality holds for $A \in \mathcal{S}_\epsilon$:

$$\forall j : \left| \sum_i A_{ij} - 1 \right| \leq \epsilon. \tag{42}$$

Using the above inequality, we get:

$$|q'_j - q_j| = \left| \tfrac{q_j}{\sum_i A_{ij}} - q_j \right| \tag{43}$$

$$= q_j \left| \frac{1}{\sum_i A_{ij}} - 1 \right| \tag{44}$$

$$\leq q_j \left(\tfrac{\epsilon}{1-\epsilon} \right) \tag{45}$$

Plugging the above inequality into μ concludes the proof:

$$\frac{q_i q'_j}{q_j q'_i} \leq \frac{1}{1-2\epsilon} \implies \mu(q, q'_i) \leq \log(\tfrac{1}{1-2\epsilon}) \leq \tfrac{2\epsilon}{1-2\epsilon}. \tag{46}$$

H Supplementary experiments

Attention–OT correlation. Figure 1 shows that, across two English–French sentence pairs, the attention maps progressively become closer to the optimal-transport (OT) plans computed from the corresponding word embeddings. To quantify this trend, we report the correlation between the attention weights and the OT solution.

Additional Results. Figure 10 repeats the experiment in Figure 2 for a larger n.

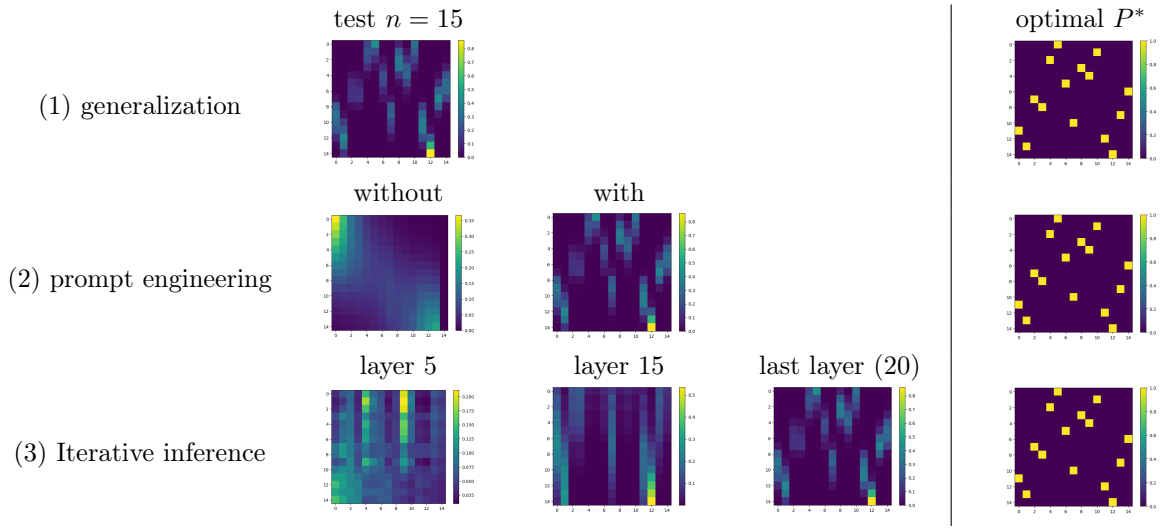


Figure 10: **Observations on In-context Learning for OT.** We extend the experiments shown in Fig. 2 to larger values of$n$. *(1)* The model is trained to solve OT with 7 data points and evaluated on 15 data points. The left image shows the attention weights, which closely approximate the OT solution shown on the right. *(2)* After specific prompt engineering, the attention weights between tokens estimate the OT solution. Notably, this prompt engineering is used in (1). *(3)* The attention weights evolve across layers, progressively yielding a more accurate approximation of the optimal solution. See Appendix A for details.