Pervasive and Mobile Computing 100 (2024) 101912

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Seeing the world from its words: All-embracing Transformers for fingerprint-based indoor localization

Son Minh Nguyen [a,*], Duc Viet Le [a], Paul J.M. Havinga [a,b]

[a] University of Twente, Enschede, The Netherlands
[b] TNO, The Netherlands

ARTICLE INFO

Keywords:
RSS fingerprints
Indoor localization
Deep learning
Transformers

ABSTRACT

In this paper, we present all-embracing Transformers (AaTs) that are capable of deftly manipulating attention mechanism for Received Signal Strength (RSS) fingerprints in order to invigorate localizing performance. Since most machine learning models applied to the RSS modality do not possess any attention mechanism, they can merely capture superficial representations. Moreover, compared to textual and visual modalities, the RSS modality is inherently notorious for its sensitivity to environmental dynamics. Such adversities inhibit their access to subtle but distinct representations that characterize the corresponding location, ultimately resulting in significant degradation in the testing phase. In contrast, a major appeal of AaTs is the ability to focus exclusively on relevant anchors in RSS sequences, allowing full rein to the exploitation of subtle and distinct representations for specific locations. This also facilitates disregarding redundant clues formed by noisy ambient conditions, thus enhancing accuracy in localization. Apart from that, explicitly resolving the representation collapse (*i.e.*, none-informative or homogeneous features, and gradient vanishing) can further invigorate the self-attention process in transformer blocks, by which subtle but distinct representations to specific locations are radically captured with ease. For that purpose, we first enhance our proposed model with two sub-constraints, namely covariance and variance losses at the *Anchor2Vec*. The proposed constraints are automatically mediated with the primary task towards a novel multi-task learning manner. In an advanced manner, we present further the ultimate in design with a few simple tweaks carefully crafted for transformer encoder blocks. This effort aims to promote representation augmentation via stabilizing the inflow of gradients to these blocks. Thus, the problems of representation collapse in regular Transformers can be tackled. To evaluate our AaTs, we compare the models with the state-of-the-art (SoTA) methods on three benchmark indoor localization datasets. The experimental results confirm our hypothesis and show that our proposed models could deliver much higher and more stable accuracy.

1. Introduction

The ever-increasing worldwide demand for smart space ecosystems, exemplified by smart buildings, smart warehouses, and smart hospitals over the last decade has fueled indoor localization systems profusely to become an indispensable enabler for many context-aware services, including but not limited to wayfinding, asset tracking, and patient monitoring. Leveraging the ubiquity of WiFi, and Bluetooth signals, Received-Signal-Strength (RSS) fingerprinting methodologies have emerged as the predominant approach for

[*] Corresponding author at: University of Twente, Enschede, The Netherlands.
E-mail addresses: m.s.nguyen@utwente.nl (S.M. Nguyen), v.d.le@utwente.nl (D.V. Le), p.j.m.havinga@utwente.nl (P.J.M. Havinga).

https://doi.org/10.1016/j.pmcj.2024.101912
Received 18 August 2023; Received in revised form 28 February 2024; Accepted 6 March 2024
Available online 11 March 2024

indoor localization where a massive number of off-the-shelf deep learning architectures, commonly employed in vision and natural language processing (NLP) tasks, come into play. In contrast to traditional methods like multilateration techniques based on Time and Angle of Arrival, which necessitate strict time synchronization between radio emitters and receivers, RSS fingerprinting approaches offer greater flexibility. These approaches undergo an initial offline phase to establish a radio map (*i.e.*, fingerprint database), where sequences of RSS anchors[1] (*i.e.*, RSS fingerprints) and their associated locations are manually surveyed. Subsequently, in the online phase, users' locations are determined by matching observed fingerprints against the established database.

Despite certain achievements in indoor localization through simple kNN enhancement methodologies [1,2], their performance is often inclined to degrade, particularly in scenarios involving sparse radio maps and dynamic environments. Additionally, these methods typically require a significant look-up time for inference. Factors such as temperature, humidity, the presence of static and moving objects, and power-constraint policies at emitters contribute to challenges such as reflection, scattering, and absorption in radio propagation, rendering RSS fingerprints highly volatile and thus impacting localization accuracy. The emergence of CNN [3,4] and RNN [5]-based models has partially mitigated these challenges through the incorporation of architectural inductive biases [6], namely multichanneling, downsampling, weight sharing, and locality, all of which prove beneficial to representation learning. This enhancement enables the models to navigate complex indoor environments more effectively, partly mitigating issues related to dynamic conditions and varying radio signal characteristics.

While these methods leverage CNN- and RNN-related inductive biases successfully, the input transformation used for CNNs and sequential computation in RNNs introduce artifacts that fabricate spatial correlations among anchors in RSS sequences. This leads to local explorations of superficial correlations between sequence anchors and the considered locations. All things considered, their inferior performance in the testing phase largely lies in the following points: *(i)* Compared to other common modalities like texts, images, and sounds, RSS fingerprint is a challenging modality, inconceivable to human reasoning, and highly susceptible to environmental dynamics, producing a mass of inconsistent clues. As a result, capturing subtle but distinct representations [7–13] for specific locations becomes nontrivial. *(ii)* The transformed input of 2D RSS fingerprints, coupled with noisy additives of false correlations, does not fully benefit from existing CNNs, as convolution kernels assume true spatial correlations among adjacent elements, a property not always applicable to RSS fingerprints. *(iii)* Original 1D fingerprint vectors lack expressiveness of information between anchors, necessitating extensive interventions to discover latent correlations characterizing specific locations. Due to reliance on sequential computation and local convolution filters with a limited receptive field, vanilla RNN- and CNN-based models respectively lose sight of the whole picture in the RSS sequence, failing to extract relevant features for specific locations.

We posit that a given RSS fingerprint encapsulates both inconsistent and consistent clues all pertaining to a specific location. However, these clues were hidden within the fingerprint's structure, with the inconsistent ones being all-pervasive. This is because of the modality's vulnerability to environmental conditions, where measurements of RSS fingerprints to a certain location are likely to be inundated with a large portion of highly noisy information. Moreover, excessive exposure to inconsistent information during the representation learning stage can lead to substantial degradation in localization performance. In contrast, directing attention exclusively to consistent clues associated with relevant locations can yield benefits. For that point, models that were not equipped with an adequate attention scheme are prone to be influenced by adverse information.

With the advent of the self-attention mechanism for drawing global dependencies in sentences, the Transformer [14], characterized by parallel computing and the avoidance of recurrent computations, has demonstrated faster convergence to the optimum in numerous NLP tasks. In computer vision (CV), convolutional architectures remain firmly established as state-of-the-art (SoTA) approaches for various vision tasks, including object detection, image recognition, and 3D reconstruction. However, the landscape has shifted in the past three years, witnessing the widespread success of various Transformer variants [15–18] in a majority of fundamental vision challenges.

Motivated by the evident superiority of Transformers in both NLP and CV domains, we extend the Transformer architecture to tackle challenges in indoor localization by drawing an analogy between RSS fingerprints and textual sentences. Particularly, the context of an excerpt is represented by the meaning of several sentences inside, analogous to the indoor localization field where the context of a location is presumably represented by a set of fingerprints. In this analogy, the nuance of each sentence is defined by its keywords, and similarly, the expressiveness of each fingerprint to the location is implicitly captured by selective inner anchors.

With that in mind, for the sake of conformity that the standard Transformer was not designed for direct application to RSS fingerprints, we first accommodate an *Anchor2Vec* layer to the fingerprints by linearly mapping these sequences of anchors to informative token embeddings as input to the Transformer. These token embeddings thereafter can be treated similarly to word token embeddings in NLP applications on which a stack of multi-head self-attention layers is fully applied. Going through the stacked attention layers can help progressively reveal the latent correlations among sequence anchors, which guides the model to subtle but distinct representations carrying hidden information consistent with considered locations. Illustrated in Fig. 3(b), we propose a novel **A**nchor-**a**gnostic **T**ransformer (eAaT+), incorporating simple yet effective architectural tweaks to further enhance localization performance. The preliminary version of AaTs [19] presented an enhanced version dubbed eAaT that is conducive to the exploration of latent correlations through the multi-head self-attention layers. To clarify, eAaT can avoid feature-level information collapse in an attempt to generate informative token embeddings for the stacked attention layers, thus encouraging considerable restraints on irrelevant features while seeking subtle but distinct representations from the beginning. This ability is enabled by the synergy between two extra sub-constraints and the primary task, optimized together in a novel multi-task learning fashion where all contributions from such constraints are equally adjusted.

[1] An anchor represents one radio-emitting source such as WiFi access points or Bluetooth beacons.

A preliminary version of this work was published in Proceedings International Conference on Pervasive Computing and Communications (PerCom) 2023 [19]. We extend the work in two key aspects. First, we conduct a theoretical analysis to evaluate the effectiveness of different configurations within Transformers, namely Post- (Post-LN) and Pre-Layer Normalization (Pre-LN) that was adopted in the preliminary version of AaTs [19]. Second, we introduce an ultimate version (eAaT+) with tweaked connections and dispositions of functional layers. This version aims to advance the representation learning process under the regularization of the gradient flow through the networks.

Overall, our core contributions are summarized as follows:

- To the best of our knowledge, we are the first to propose a base transformer architecture, adapted with a specialized *Anchor2Vec* layer (bAaT) capable of meaningfully interpreting RSS fingerprints of discrete anchors for specific locations.
- Furthermore, we deliver two enhanced versions of AaTs (eAaT & eAaT+) that not only incorporate two extra sub-constraints deemed as unsupervised tasks but also feature an advanced disposition of functional layers. These versions effectively address the information collapse at the feature level with great attention to detail, for which we again present a novel multi-task learning scenario referred to as Adaptive Random Loss Weighting (Adaptive RLW) to automatically learn the balance between contributions from different learning tasks. This facilitates the comprehensive utilization of learned common knowledge among tasks to enhance performance in the primary task of localization.
- Our approach demonstrates state-of-the-art performances through extensive experiments and visual interpretation on public indoor localization datasets. Besides, this paper expects to provide fresh impetus to the exploration of transformer-based approaches for fingerprint-based indoor localization tasks.

2. Related work

Within the last few years, numerous deep learning models have been suggested and successfully integrated into indoor localization, particularly fingerprint-based techniques. We briefly review the most related literature in the following:

Fingerprint-based localization Approaches. In lieu of jumping on the bandwagon for further advancements in vanilla matching functions (*e.g.*, kNN variants [1,20,21]) with different distance metrics (*e.g.*, Euclidean, Manhattan, and Cosine metrics), some works adopt deep learning models to automatically seek more pertinent features for fingerprint matching. Specifically, CNN-based fingerprinting methods [4,22] were put forward with extra preprocessing steps to convert 1D-RSS measurements into expected input shapes to CNNs. Utilizing inductive biases inherent in such architectures to represent RSS fingerprints, such methods yield decent localizing performance over public indoor localization datasets. WiDeep [23] integrates probabilistic stacked auto-encoders to handle the noise and to capture complex relationships between the WiFi anchors. To incorporate temporal correlations among RSS sequences within one trajectory into the representation learning stage, Hoang [24] et al. applies RNN-based architectures into RSS input vectors without any input shape transformation for the trajectory localizing, which provides sequential output locations at a certain interval. By extension, other approaches [25,26] cast the fingerprinting problem as a domain adaption problem, in which fingerprint collection time and device are considered as independent domains. Accordingly, domain adaption-based frameworks have been adopted to extract domain-independent representations.

Moreover, prior works on transformer-based architectures are worth mentioning. Since their inception 5 years ago [14], many variants have been suggested and gained ground for impressive successes in both NLP and vision tasks among researchers who look for alternative powerful architectures other than traditional models.

NLP Transformers. Transformers were for the first time presented by Vaswani et al. [14] for machine translation, and have since ripened into the SoTA method in many NLP tasks. Continuously, Devlin et al., [27] pretrained a stack of bidirectional transformer-encoders (BERT) on self-supervised pre-training tasks, *e.g*, Mask Language Modeling and Next Sentence Prediction to inject bidirectionally attended representation, and sentence-level understanding biases for non-autoregressive tasks, while the GPT series [28–30] employ language modeling as its pretraining task on transformer-decoders to ameliorate autoregressive tasks.

Vision Transformers. Inspired by the achievements of Transformers on NLP tasks [14,27,30,31], Dosovitskiy et al. [16] proposed vision Transformers (ViT) that can represent fixed-size patches of an image with position-encoded embeddings for image recognition tasks via a Linear Projection of Flattened Patches layer. Whereas ViT cannot outperform state-of-the-art CNNs on the standard ImageNet benchmark, it reaches excellent results when pretrained on the larger JFT-300M dataset. DeiT [32] is devised with the knowledge distillation-based learning to augment learned representations from ViT, which bested ResNet [33] by a significant margin. Some following works such as T2T-ViT [34], LocalViT [35] and CrossViT [36] arrive with improvements in the architecture design of ViT. Another line of research [37–39] attempts to transfer the inductive bias of CNN into Transformers. In addition, some endeavors [18,40–43] are made to accommodate ViT to other vision tasks.

Time-series Transformers. Despite recent advances in the CV and NLP domains, the integration of Transformer-based architectures for supervised learning has seen limited exploration in the Inertial Measurement Unit- (IMU) based Human Activity Recognition (HAR) domain. The work by Shavit et al. [44] laid the foundation, employing a convolutional architecture to encode inertial signals into latent embeddings. These embeddings were seamlessly integrated into a Transformer architecture for activity classification, marking an initial fusion of convolutional and Transformer models in this domain. Drawing inspiration from the efficiency gains demonstrated by MobileViT [45], Sannara et al. [46] simplified the attention process further to reduce computing resources, enabling practical applications on embedded devices, and addressing the challenges of resource-intensive tasks in real-world scenarios. By

Table 1
Key notations: Symbols and description.

Symbol	Description
$\mathbf{X}$	1D RSS fingerprint
$\mathbf{N}$	The number of raw anchors inside each $\mathbf{X}$
$\mathbf{B}$	The number of fingerprints in each batch
$\mathbf{E}'$	Unraveled token embeddings before $\mathbf{E}$
$\mathbf{e}_i$	A single token in $\mathbf{E}$
$\mathbf{E}$	Token embeddings
$\mathbf{E}_{pos}$	Positional embeddings
$\mathbf{z}_j^i$	The output at the ith position in the jth layer
$\mathbf{d}$	The number of attributes inside each anchor $\mathbf{e}_i$
k	The number of tokens $\mathbf{e}_i$ in $\mathbf{E}$
$\mathbf{A}$	Attention map computed in Self-Attention blocks
$\mathbf{Q}$	Query tokens used in Self-Attention blocks
$\mathbf{K}$	Key tokens used in Self-Attention blocks
$\mathbf{V}$	Value tokens used in Self-Attention blocks
LN	Layer normalization
$\mathcal{O}$	Functional operations (e.g., Self-Attentions, Feed-Forward operations)
λ	Weighting factor
J	Objective Constraint
γ	The desired average standard deviation used in Eq. (8)
ρ	Stabilizing scaler used in Eq. (8)

extension, Earthformer [47] proposed cuboid attention for efficient spacetime modeling, which involves decomposing the data into cuboids and applying cuboid-level self-attention in parallel, demonstrating superior performance in weather and climate forecasting. Recently, AirFormer [48] introduced a novel approach featuring a dartboard spatial self-attention module and a causal temporal self-attention module to efficiently capture spatial correlations and temporal dependencies, respectively. Furthermore, it augmented Transformers with latent variables to capture data uncertainty and improve performance in air quality forecasting.

3. Proposed model

To push the boundaries of the aforementioned drawbacks in indoor localization, we directly elaborate a standard transformer encoder architecture with a few tokenizing tweaks tailored for fingerprinting problems. This architecture enables extensive multi-head attention to global dependencies by taking into account the entire context sequence when refining each token. Supported by parallel computation, this ability is further deepened by a stack of attention layers, encouraging the model to efficiently realize correlations and important clues hidden in the sequence anchors. Furthermore, we propose two feature-level sub-constraints as auxiliary unsupervised tasks, namely Covariance J_{cov} and Variance J_{var} losses to seize more control of the representation learning process. In this way, the diversity between tokenized sequences in general, and distinct attributes at each token in particular are more guaranteed, thereby facilitating multi-head attention layers to better disseminate their consideration to tokenized anchors according to considered locations. To this end, eAaT jointly accomplishes these tasks through the proposed Adaptive RLW learning, a novel multi-task learning manner dynamically balancing the influence between subtasks and the primary task on learned gradients over batches. For our final design, specifically eAaT+, we further make some simple but effective tweaks regarding the disposition of functional layers inside the transformer encoder blocks. This comprehensive transformation can alleviate impediments existing in its predecessors, for which the backward flow of gradients with sufficient magnitude to all levels of transformer encoder blocks is ensured. Key symbols used in this paper are listed in Table 1.

3.1. Overview

We mainly follow the settings of BERT [27] in NLP where the input is expected to be a sequence of words. Assume a 1D RSS fingerprint $\mathbf{X} = \left[\mathbf{x}_1, \dots, \mathbf{x}_N\right] \in \mathbb{R}^{1\times\mathbf{N}}$ is defined by a sequence of $\mathbf{N}$ raw anchors, each containing an exclusive RSS value from one of $\mathbf{N}$ emitters. To be fully modeled by the Transformer as a sequence of words, the RSS fingerprint X needs extra refinements. As illustrated in Fig. 1, by dint of our designed tokenizing process at the *Anchor2Vec* layer, detailed in Sub Section 3.3, a given 1D RSS fingerprint $\mathbf{X}$ is interpreted into more meaningful token embeddings $\mathbf{E} = \left[\mathbf{e}_0, \dots, \mathbf{e}_k\right]^{\mathrm{T}} \in \mathbb{R}^{(k+1)\times\mathbf{d}}$, each manifesting important information under $\mathbf{d}$ distinct attributes, akin to word token embeddings in NLP. To provide more details, $\mathbf{X}$ undergoes an initial linear mapping to an intermediate embedding of k dimensions. This intermediate embedding acts as a fixed-size buffer, offering flexibility to accommodate varying numbers of anchors and exhibiting an adaptive effect on representation refinement. Specifically, this effect is anticipated to induce either feature consolidation for sparse RSS fingerprints in high dimensions or feature interpretation for dense RSS ones with lower dimensions. Subsequently, each element within the intermediate embedding is augmented with information-expressiveness in dimension through further expansion to a vector of $\mathbf{d}$ dimensions. Simultaneously, the expansion is closely supervised by $J_{cov\&var}$ both in dimension and batch, ensuring effective and faithful enrichment. This supervision significantly mitigates information collapse in the tokenizing process.

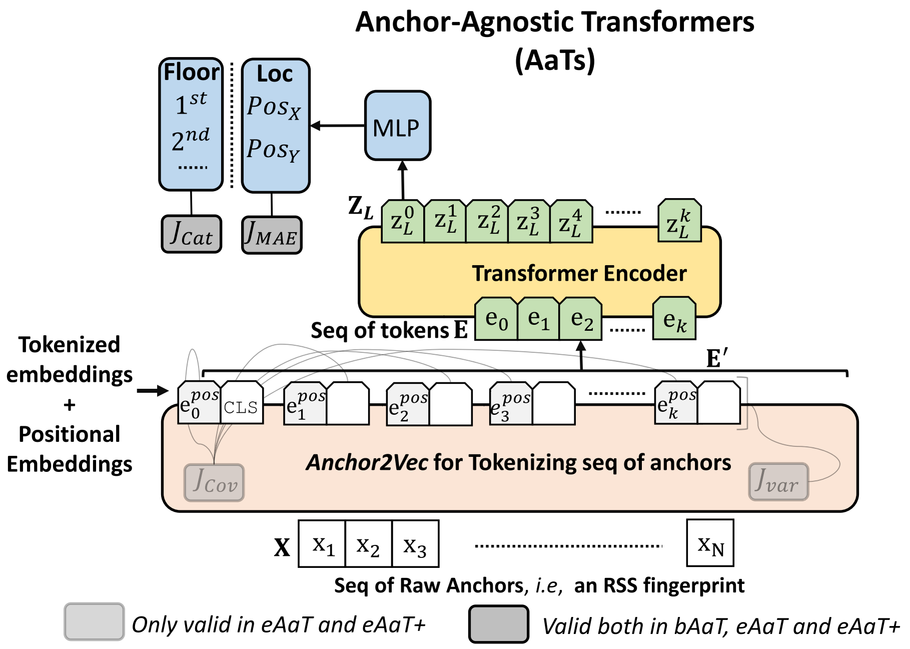

Anchor-Agnostic Transformers (AaTs)
Floor
1^{st}
2^{nd}
......
Loc
Pos_X
Pos_Y
MLP
J_{Cat}
J_{MAE}
$\mathbf{Z}_L$
z_L^0 z_L^1 z_L^2 z_L^3 z_L^4 .... z_L^k
Transformer Encoder
Seq of tokens $\mathbf{E}$
e_0 e_1 e_2 .... e_k
$\mathbf{E}'$
Tokenized embeddings + Positional Embeddings
e_0^{pos} CLS e_1^{pos} e_2^{pos} e_3^{pos} .... e_k^{pos}
Anchor2Vec for Tokenizing seq of anchors
J_{Cov}
J_{var}
$\mathbf{X}$ x_1 x_2 x_3 .... x_N
Seq of Raw Anchors, i.e, an RSS fingerprint
Only valid in eAaT and eAaT+
Valid both in bAaT, eAaT and eAaT+

Fig. 1. An overview of **A**nchor-**a**gnostic **T**ransformers (AaTs). Traditional Transformer architectures are not optimized for handling sequences of discrete anchors, such as Received Signal Strength (RSS) fingerprints represented as $\mathbf{X}$. To address this limitation, we introduce the Anchor2Vec Layer for anchor tokenization. This layer transforms anchor sequences into meaningful token embeddings, denoted as $\mathbf{E}$, capturing rich contextual attributes. Each sequence or fingerprint is appended with [CLS] for summarization, suitable for subsequent classification or regression tasks. Learnable positional embeddings $\mathbf{E}_{pos}$ encode positional information among the tokenized anchors. The resulting sequence $\mathbf{E}$ is input to the Transformer Encoder with $\mathbf{L}$ blocks.

Drawing an analogy with BERT's [CLS] token, we do prepend a learnable embedding [CLS] of $\mathbf{d}$ dimensions to the sequence of token embeddings. The output from the Transformer encoder, denoted as $\mathbf{z}_L^0$ serves as a summarized representation of the entire sequence that can be used both for regression and classification thereafter. Note that spatial information-awareness among the tokens in the sequence, at the same time, is enhanced by incorporating a learnable 1D positional embedding $\mathbf{E}_{pos} \in \mathbb{R}^{1\times(\mathbf{k}+1)\cdot\mathbf{d}}$ into the ultimate token embeddings $\mathbf{E}$, making them ready for utilization by the Transformer.

3.2. Attention mechanism of transformer

As mentioned earlier, the Transformer has established its powerful performance in many vision and language tasks for its multi-head self-attention and parallel computational capabilities. In this regard, the Transformer computes and uses attention $\mathbf{A}$ from three types of inputs, Q (query), K (key), and V (value), which are linearly projected from token embeddings. Its computation for $\mathbf{A}$ is given by

$$\mathbf{A}\left(Q, K, V\right) = \mathrm{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d}}\right)V, \tag{1}$$

where Q, K, and V are all collections of projected features, each of which is represented by a d-dimensional vector. To be specific, $Q = \left[q_0, \dots, q_k\right]^{\mathrm{T}} \in \mathbb{R}^{(k+1)\times d}$ is a collection of $k+1$ features corresponding to the number of tokenized anchors in a sequence. Similarly, K and V are each a collection of $k+1$ features, *i.e.,* $K, V \in \mathbb{R}^{(k+1)\times d}$. In Eq. (1), V is attended with the weights computed from the similarity between Q and K.

The above computation is usually multiplexed in the way called multi-head attention. It enables the model to manipulate multiple attention distributions in parallel on different representation subspaces, aiming to increase representational power. The outputs of H 'heads' are concatenated, followed by linear transformation with learnable weights $W^O \in \mathbb{R}^{d\times d}$ as

$$\mathbf{A}^k\left(Q, K, V\right) = [\mathrm{head}_1, \dots, \mathrm{head}_H]W^O, \tag{2}$$

Where each head is expressed as follows:

$$\mathrm{head}_h = \mathbf{A}\left(QW_h^Q, KW_h^K, VW_h^V\right), h = 1, \dots, H, \tag{3}$$

where $W_h^Q, W_h^K, W_h^V \in \mathbb{R}^{d\times d_H}$ each are learnable weights inducing a linear projection from the feature space of d-dimensions to a lower space of $d_H\,(= d/H)$-dimensions. With the reduced dimension of each head by the number of heads, the total computational

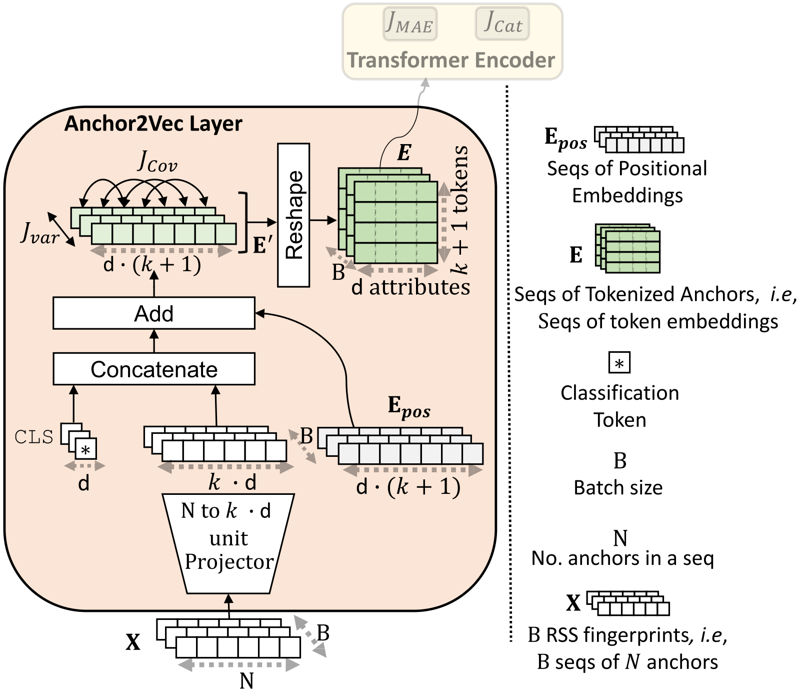


Fig. 2. Anchor2Vec Layer. The layer is to disentangle **B** sequences of **N** discrete anchors into **B** sequences of $k+1$ dimensional embeddings $\mathbf{E} \in \mathbb{R}^{(k+1)\times \mathbf{d}}$, each possessing **d** different attributes, including positional correlations among sequence tokens through learnable positional embeddings $\mathbf{E}_{pos}$. Initially, a vector of **N** anchors is linearly projected to a k-dimensional vector $\in \mathbb{R}^k$ for adaptability (e.g., compression for large $\mathbf{N} \gg k$ or interpretation for minor $\mathbf{N} < k$). Subsequently, the vector is further interpreted to a higher dimension, where each element is expanded to **d** attributes. To prevent feature-level information collapse, eAaT imposes two sub-constraints, compelling the model to generate unique sequences with informative tokens carrying distinct attributes.

cost for multi-head attention remains similar to that of single-head attention with full dimensionality. Overall, one attentional block $\mathbf{A}^k(Q, K, V)$ includes the following learnable weights:

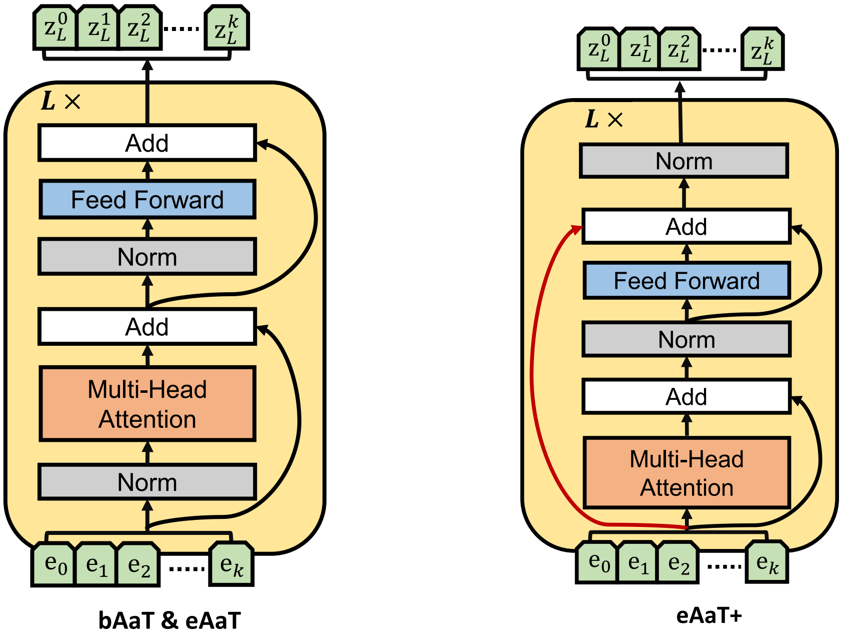
$$\left(W_1^Q, W_1^K, W_1^V\right), \ldots, \left(W_H^Q, W_H^K, W_H^V\right) \text{and} W^O. \tag{4}$$

Self-attention is the main component of transformers, which enables the model to make use of contextual information from neighbor anchors for predicting the current token.

3.3. Anchor2Vec layer

As presented in Fig. 2, the *Anchor2Vec* layer is intended for transforming sequences of raw anchors to sequences of tokenized anchors, *i.e*, sequences of token embeddings. To be more specific, the sequence of raw anchors X is transformed into a 1D intermediate embedding of k dimensions to mainly gain the adaptive effect. With the help of this effect, the overheads of computations and the footprint of learning parameters are considerably shrunk for sparse fingerprint datasets with over hundreds of anchors whereas triggering slight growth in computation for dense ones with a few anchors. Afterward, such a 1D embedding is further enriched to the one with higher dimensions of $k \cdot \mathbf{d}$ to better express salient information among tokenized anchors. Furthermore, [CLS] token responsible for synthesizing the representation of all tokenized anchors involved in the sequence is placed at the first place $\mathbf{x}_0$ of the sequence. Also, the sequence of token embeddings is element-wise fused to the learnable position embeddings $\mathbf{E}_{pos}$ for retaining positional information, which helps the model not perceive the sequence as “a bag of words”. The resulting sequence of token embeddings **E** thereafter serves as input to the encoders. As regards the eAaT, there is an extra step needed, in which the token embeddings **E** are batch- and dimension-wise regulated by the following sub-constraints that play roles as two unsupervised tasks. The imposition of such subtasks to satisfy the primary task of localization is achieved in a multi-task learning fashion.

*Covariance Constraint.*The constraint J_{Cov} is proposed to ensure high information diversity for all attributes of each tokenized anchors $\{\mathbf{e}_i\}_{i=0}^{k} \forall \mathbf{e}_i \in \mathbb{R}^{\mathrm{d}}$ in a sequence **E** by decorrelating the entire attributes of the sequence token embedding over a batch of **B** examples. Therefore, the informational collapse where the attributes of different tokens in the same sequence would vary together or be highly correlated is significantly diminished. As shown in Fig. 2, instead of locally allowing such diversity in one single token embedding of **E**, where only every **d** attributes is independently taken into consideration, we simultaneously dispense it comprehensively to all token embeddings via the sequence $\mathbf{E}' \in \mathbb{R}^{\mathrm{d}\cdot(k+1)}$, a sequence of unraveled token embeddings before **E**, which strictly encourages


z_L^0 z_L^1 z_L^2 ..... z_L^k
$L\times$
Add
Feed Forward
Norm
Add
Multi-Head Attention
Norm
e_0 e_1 e_2 ..... e_k
bAaT & eAaT
z_L^0 z_L^1 z_L^2 ..... z_L^k
$L\times$
Norm
Add
Feed Forward
Norm
Add
Multi-Head Attention
e_0 e_1 e_2 ..... e_k
eAaT+

(a) Transformer Encoder for bAaT and eAaT (b) Tweaked Transformer Encoder for eAaT+

Fig. 3. Comparison of different variants of Anchor-agnostic Transformers. (a) Basic and enhanced versions of AaTs; (b) A version of the enhanced version plus a new disposition of functional layers and an extra residual connection indicated with the red curved arrow.

globally distinct attributes in the entire sequence. To this end, relations among element attributes over a batch should be modeled with a covariance matrix:

$$Cov\left(\mathrm{E}'\right) = \frac{1}{\mathrm{B}-1}\sum_{i=1}^{\mathrm{B}}\left(\mathrm{E}'_i - \bar{\mathrm{E}}'\right)\left(\mathrm{E}'_i - \bar{\mathrm{E}}'\right)^{\mathrm{T}} \tag{5}$$

where

$$\bar{\mathrm{E}}' = \frac{1}{\mathrm{B}}\sum_{i=1}^{\mathrm{B}}\mathrm{E}'_i \tag{6}$$

With Eq. (5), the constraint is deduced and then satisfied by minimizing the following expression:

$$J_{\mathrm{Cov}} = \frac{1}{(k+1)\cdot \mathrm{d}}\sum_{i\neq j}\left[Cov\left(\mathrm{E}'\right)\right]^2_{i,j} \tag{7}$$

The Eq. (7) aims to minimize all off-diagonal values, each of which indicates the level of correlation between two different attributes. At the optimal point, one sequence is expected to hold informative and distinct attributes across its tokens.

Variance Constraint. In a similar way, this constraint remains implemented on $\mathbf{E}'$ for the comprehensive effect. Given a batch of $\mathbf{B}$ sequences, a hinge loss is employed to maintain the variation among sequences of the unraveled token embeddings above a given threshold. This term generally forces sequences of the token embeddings within a batch to be different, which implies that each sequence would bear its own attributes. Thus, the collapse as a result of the shrinkage of the token embeddings towards zero, *i.e.*, different locations in one batch represented by the same sequence of token embeddings is explicitly prevented. In the following, the variance loss is instead computed on a standard deviation of $\mathbf{E}'$ over $\mathbf{B}$ sequences to rule out the cases that the gradient with respect to E'_i becomes close to zero as E'_i comes close to $\bar{E}'$.

$$Std\,(\mathrm{E},\,\rho) = \sqrt{\frac{\sum_{i=0}^{\mathrm{B}}\left(\mathrm{E}'_i - \bar{\mathrm{E}}'\right)^2}{\mathrm{B}-1} + \rho} \tag{8}$$

Based on Eq. (8), we define the variance constraint via a hinge loss function:

$$J_{Var} = \frac{1}{(k+1)\cdot \mathrm{d}}\sum_{j=1}^{(k+1)\cdot \mathrm{d}}\max\left(0,\ \gamma - Std\,(\mathrm{E},\,\rho)\right) \tag{9}$$

where γ is the desired average standard deviation, and ρ is a very small scaler to keep Eq. (8) valid at all times.

With the benefit of Eq. (7), and Eq. (9), the *Anchor2Vec* layer can meaningfully represent sequences of raw anchors with sequences of token embeddings that not only acquire independent details in cross-sequence but also have distinct attributes in intra-sequence. That accordingly facilitates the multi-head self-attention process to efficiently realize subtle but distinct representations.

3.4. Adaptive random loss weighting in multi-task learning

To mediate disputes between learning tasks to effectively learn relevant representation for the primary task, the sub-constraints should be preserved in harmony with the main goal in a multi-task learning (MTL) fashion. Since all the predefined tasks, including two unsupervised tasks and the main one, are not closely related, it thus is nontrivial to training an MTL model than training each of them separately. The discrepancies between the learning tasks might cause the implications of conflicting gradients among these tasks or dominating gradients at a given task to the others [49], thus leading to unsatisfactory performance for other tasks. Such phenomena are related to the task-balancing problem [50] in MTL, where balancing the influences between sub-tasks and the primary task remains very challenging. Rather than directly intervening in learned gradients of considered tasks, the work [51] presents a Random Loss Weighting (RLW) scheme to indirectly balance influences among these tasks. Inspired by this opinion, we propose an advanced weighting approach coined Adaptive Random Loss Weighting manner (Adaptive RLW) in which contribution balance between different tasks through weighting factors $\bar{\lambda} = \left[\bar{\lambda}_1, \bar{\lambda}_2, \bar{\lambda}_3\right]$ in Eq. (10) is utterly learnable. The previous research on RLW has been almost entirely restricted to the random draw for weight factors $\bar{\lambda}$ from only one of the following distributions, *e.g.*, Normal, Dirichlet, Bernoulli, Uniform, and Random Normal which is initialized with a random mean and random standard deviation, yet without any feedback from the model. This works on the assumption that randomly introducing such weighting factors from one of the predefined distributions is regarded as another stochastic stream that enables the model to automatically align its consideration of computed gradients with involved tasks. The concept has already been proved for the first stochastic stream of randomized separate batches of input data from the whole dataset, in which the model has no control of randomization, but is still capable of directing its learning ability to the whole dataset.

$$J_{overall} = \bar{\lambda}_0 J_{MAE/Cat} + \bar{\lambda}_1 J_{Cov} + \bar{\lambda}_2 J_{Var} \tag{10}$$

More flexibility and relaxation in the multi-task balance of RLW could greatly relieve the anxiety of gradient conflict or domination at one task over others. This suggests that each batch needs weighting factors of disparate magnitudes for different distributions to properly regulate the gradients of involved tasks. However, the predecessor only gains one constant advantage of the same distribution for all batches without any interoperation from the model. In light of this, the Adaptive RLW was designed to responsively capture feedback from the model while being wholly able to simultaneously reap multiple gains from all five different distributions to better balance influences among predefined tasks on the model's learning ability. To be more specific, the feedback is achieved by directly appraising the model's state via an analysis on the enhanced feature map E^{ehc} to learn intermediate weights W_{dis} corresponding to five different random distributions, succinctly interpreted in Algo. 1. As a result, those weighting factors from relevant distributions are significantly enhanced whereas those from unnecessary distributions are highly suppressed, which indirectly allows for more effective adjustments to computed gradients from each task jointly, thereby significantly alleviating dominant gradients at a certain task.

Algorithm 1: Adaptive Random Loss Weighting

```
Input: number of tasks $T$, number of weight distributions $P$, learning rate $\alpha$, dataset $D$, model parameters $\theta$, weight distribution $\{\rho_i(\tilde{\lambda})\}_{i=0}^{P-1}$
Output: task-specific weights $\{\bar{\lambda}_t\}_{t=0}^{T-1}$, task-specific losses $\{J_t\}_{t=0}^{T-1}$
Initialization: $\lambda_{dis} \in \mathbb{R}^{T\times P}$, $\bar{\lambda} = [\bar{\lambda}_0, \cdots, \bar{\lambda}_{T-1}]$, $\lambda \in \mathbb{R}^T$
for $t = 1$ to $T - 1$ do
    Compute loss $J_t(D, \theta)$
end for
Extracting the unraveled token embeddings $\mathbf{E}'$
/* Performing enhancements on $\mathbf{E}'$ by taking all the involved sequences within a batch into account */
$S_{seqs} = \mathrm{softmax}(\mathrm{E}'\mathrm{E}'^T)$
$\mathrm{E}^{ehc} = S_{seqs}\mathrm{E}'$
/* Computing distribution weights $W_{dis} \in \mathbb{R}^P$ among $P$ distributions for a batch of $\mathbf{B}$ sequences */
$W_{dis} = \mathrm{softmax}\left(\frac{1}{\mathrm{B}}\sum_{j=0}^{\mathrm{B}-1} Pooling\left(\mathrm{E}^{ehc}{}_j\right)\right)$
// Sampling weights from $P$ distributions
for $i = 0$ to $P - 1$ do
    $\lambda_{dis}[: T, i] \sim \rho_i(\tilde{\lambda})$
end for
for $t = 0$ to $T - 1$ do
    $\lambda[t] = \sum_{p=0}^{P-1} \lambda_{dis}[t, p] \cdot W_{dis}[p]$
end for
$\bar{\lambda} = \mathrm{softmax}(\lambda)$                // Computing task-specific weights
return $\{\bar{\lambda}_t\}_{t=0}^{T-1}$, $\{J_t\}_{t=0}^{T-1}$
```

3.5. Transformer encoder

Variants of AaTs progressively undergo significant upgrades over versions. As shown in Fig. 3(a), the first two versions of our preliminary version [19], namely bAaT, and eAaT, however, were placed on the same transformer encoder architecture, i.e., Pre-LN style. The reason behind this arrangement is that we put a premium on stability over performance during the course of training, especially for the relatively shallow architecture of AaTs. As pointed out in [52–54], although the Pre-LN-styled transformer encoders show more resilience against gradient vanishing when training with deep Transformers, these architectures appear to be vulnerable to the representation collapse instead. This is because the complexity of these networks is simplified quickly with residual connections over their blocks, which makes them less expressive to the complexity of input data. In contrast to Pre-LN-styled architecture, the Post-LN-styled ones can offer better performance, albeit with only a few transformer encoder blocks. The critical distinction between Pre-LN- and Post-LN-styled architectures in operation is generally manifested in the equation below:

$$\begin{aligned}&\mathrm{PreLN}\,(\mathbf{E}) = \mathbf{E} + \mathcal{O}\,(\mathrm{LN}\,(\mathbf{E}))\\&\mathrm{PostLN}\,(\mathbf{E}) = \mathrm{LN}\,(\mathbf{E} + \mathcal{O}\,(\mathbf{E}))\end{aligned} \tag{11}$$

where $\mathbf{E} \in \mathbb{R}^{(k+1)\times\mathbf{d}}$, LN, and $\mathcal{O}$ denote the token embeddings that were tokenized by *Anchor2Vec*, Layer Normalization, and functional operations (e.g., Self-Attentions, Feed-Forward operations), respectively. In the same way, this is also observed in Fig. 3, which depicts the basic differences between Pre-LN (e.g., bAaT & eAaT) and advanced Post-LN (eAaT+) styles in layer disposition. The Pre-LN-styled architectures immediately prioritize the normalization of the input token embeddings. The normalization, however, is executed upon completion of operations for the Post-LN-styled architectures. Such dispositions in both styles could bring different effects on model performance. To effectively achieve certain improvements in performance, we theoretically investigate the occurrence of the problems above during the back-propagation of regular Transformers for these two styles.

3.5.1. Pre-LN - The basic foundation of bAaT and eAaT

Given a stack of L Pre-LN-styled transformer encoder blocks, output E_L at the top block L is defined as follows:

$$\mathrm{E}_L = \mathrm{E}_{L-1} + \mathcal{O}_a\left(\mathrm{LN}\left(\mathrm{E}_{L-1}\right)\right) + \mathcal{O}_f\left(\mathrm{LN}\left(\mathrm{E}_{L-1} + \mathcal{O}_a\left(\mathrm{LN}\left(\mathrm{E}_{L-1}\right)\right)\right)\right) \tag{12}$$

where E_{L-1} is the attended token embeddings in previous blocks, which are first normalized with the LN layer before being further processed by the Self-Attention layer $\mathcal{O}_a$. To which, shortly afterward, a residual connection from E_{L-1} is established. By extension, this process then perpetuates itself for the feed-forward layer $\mathcal{O}_f$.

$$\frac{\partial \mathrm{E}_L}{\partial \mathrm{E}_k} = \mathrm{I} + \sum_{k}^{L-1} \frac{\partial \mathcal{O}_a\left(\mathrm{LN}\left(\mathrm{E}_{L-1-k}\right)\right)}{\partial \mathrm{E}_k} + \frac{\partial \mathcal{O}_f\left(\mathrm{LN}\left(\mathrm{E}_{L-1-k} + \mathcal{O}_a\left(\mathrm{LN}\left(\mathrm{E}_{L-1-k}\right)\right)\right)\right)}{\partial \mathrm{E}_k} \tag{13}$$

For the sake of simplicity, we only make the derivatives among the transformer encoder blocks the central focus of this scrutiny. Such local snapshots remain valid for the functional reflection of the whole model as the course of back-propagation is contiguously computed using the Chain-rule method. According to Eq. (12), we take the derivative of E_L w.r.t E_k at a specific block k, as shown in Eq. (13), in which I denotes an identity matrix. The derivative at the block k represents the magnitude of gradients for an update on its parameters. Due to the constant presence of the identity matrix I, the gradient flow to the entirety of transformer encoder blocks is always ensured. As suggested by the iterative summation in Eq. (13), this also implies that blocks closer to the bottom can receive a larger magnitude of gradients than ones distant, i.e., the top blocks in which the received amount might be inadequate for an update. Furthermore, since the identity matrix I sets a direct flow of gradients from the top blocks to any lower blocks, even the bottom block by the same amount, the complexity of the model is likely being oversimplified, thereby degenerating its expressiveness to the complexity of the data. These findings suggest that the representation collapse as a result of the unregulated gradient flow (e.g., namely an insignificant magnitude of gradients in top blocks, and the direct flow across transformer encoder blocks) is inevitable. This limitation can be solved with the Post-LN styles, in which operations $\mathcal{O}$ are given priority over LN layers.

3.5.2. Post-LN

Compared with the Pre-LN styles, the Post-LN-styled transformer encoder of L blocks is characterized by a different disposition of functional layers, for which the working mechanism is conceptually condensed in the form of the following equation.

$$\mathrm{E}_L = \mathrm{LN}\left[\mathrm{LN}\left(\mathrm{E}_{L-1} + \mathcal{O}_a\left(\mathrm{E}_{L-1}\right)\right) + \mathcal{O}_f\left(\mathrm{LN}\left(\mathrm{E}_{L-1} + \mathcal{O}_a\left(\mathrm{E}_{L-1}\right)\right)\right)\right] \tag{14}$$

Notice that every operation in sequence is performed directly on token embeddings. The LN layers are applied right after the establishment of residual connection, i.e., a knot joining the flows of E_{L-1} and attended token embeddings $\mathcal{O}_a\left(\mathrm{E}_{L-1}\right)$. For a complete transformer encoder block as illustrated in the Eq. (14), the computation is expanded in the same way for $\mathcal{O}_f$.

$$\begin{aligned}&\frac{\partial \mathrm{E}_L}{\partial \mathrm{E}_k} = \frac{\partial \mathrm{LN}(\mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))+\mathcal{O}_f(\mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))))}{\partial(\mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))+\mathcal{O}_f(\mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))))}\cdot\\&\left(\frac{\partial \mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))}{\partial(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))}\mathrm{I} + \frac{\partial \mathcal{O}_f(\mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1})))}{\partial \mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))}\,\frac{\partial \mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))}{\partial(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))}\mathrm{I}\right)\cdot\\&\left(\prod_k^{L-1} \frac{\partial \mathrm{E}_{L-1-k}}{\partial \mathrm{E}_{L-2-k}} + \frac{\partial \mathcal{O}_a(\mathrm{E}_{L-1-k})}{\partial \mathrm{E}_{L-2-k}} \prod_k^{L-2} \frac{\partial \mathrm{E}_{L-2-k}}{\partial \mathrm{E}_{L-3-k}}\right)\end{aligned} \tag{15}$$

Subsequently, we proceed with derivative of E_L w.r.t E_k at a specific layer k. As observed in Eq. (17), none of the direct shortcuts between upper blocks and lower blocks indicated by an independent identity matrix I are established for Post-LN styles. Instead, all backward flows of information to lower blocks in back-propagation are strictly regulated with combinations of derivatives of LN layers and other operations, e.g., $\mathcal{O}_f$, and $\mathcal{O}_a$, which thus relieves the oversimplification in model complexity. However, such updating information in the form of the gradient flow might be subjected to quadratic shrinkage, which is imposed by the serial products of two contiguous derivatives of LN layers for every single step of the conveyance through each block. This implies that necessary updates cannot be made at the bottom blocks. For this situation, we make some tweaks to Post-LN styles in an attempt to preserve their conveyance of informative representations, but also to provide an extra shortcut that can regulate the flow of gradient to all transformer encoder blocks.

3.5.3. *Tweaked transformer encoder-eAaT+*

With the practical benefits of Post-LN-styled Transformers for representation learning, we transform the Pre-LN disposition of AaTs constituted by L Transformer encoder blocks into the Post-LN disposition. In order to mitigate gradient vanishing issues, a residual connection between the input token embeddings from the beginning straight to the point after $\mathcal{O}_f$ at the end is established for each transformer encoder block, as visually indicated with a red arrow in Fig. 3(b). In particular, for the forward pass, the ensuing knot in each block acts as an aggregating point of three different incoming flows. Thereafter, the aggregated information is normalized by the last LN layer as a way of circumventing the direct flow of gradient in the back-propagation.

$$\mathrm{E}_L = \mathrm{LN}\left[\mathrm{E}_{L-1} + \mathrm{LN}\left(\mathrm{E}_{L-1} + \mathcal{O}_a\left(\mathrm{E}_{L-1}\right)\right) + \mathcal{O}_f\left(\mathrm{LN}\left(\mathrm{E}_{L-1} + \mathcal{O}_a\left(\mathrm{E}_{L-1}\right)\right)\right)\right] \tag{16}$$

From theoretical aspects, the residual connection is performed by an introduction of E_{L-1} term to Eq. (16). This introduction is an important factor to ensure stability in back-propagation to bottom blocks where the flow of gradients of sufficient magnitude is hard to reach. Below we theoretically verify the significance of our tweaks.

$$\begin{aligned}\frac{\partial \mathrm{E}_L}{\partial \mathrm{E}_k} = &\frac{\partial \mathrm{LN}(\mathrm{E}_{L-1}+\mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))+\mathcal{O}_a(\mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))))}{\partial(\mathrm{E}_{L-1}+\mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))+\mathcal{O}_a(\mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))))}\cdot \\ &\left(\begin{array}{l}\prod_k^{L-1} \frac{\partial \mathrm{E}_{L-1-k}}{\partial \mathrm{E}_{L-2-k}} + \\ \left(\begin{array}{l}\frac{\partial \mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))}{\partial(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))}\mathrm{I} + \\ \frac{\partial \mathcal{O}_a(\mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1})))}{\partial \mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))}\frac{\partial \mathrm{LN}(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))}{\partial(\mathrm{E}_{L-1}+\mathcal{O}_a(\mathrm{E}_{L-1}))}\mathrm{I}\end{array}\right)\cdot \\ \left(\prod_k^{L-1} \frac{\partial \mathrm{E}_{L-1-k}}{\partial \mathrm{E}_{L-2-k}} + \frac{\partial \mathcal{O}_a(\mathrm{E}_{L-1-k})}{\partial \mathrm{E}_{L-2-k}} \prod_k^{L-2} \frac{\partial \mathrm{E}_{L-2-k}}{\partial \mathrm{E}_{L-3-k}}\right)\end{array}\right)\end{aligned} \tag{17}$$

As observed in Eq. (17), the introduction of a new term E_{L-1} in Eq. (16) opens a shortcut $\frac{\partial \mathrm{E}_{L-1-k}}{\partial \mathrm{E}_{L-2-k}}$ to lower blocks. Since this shortcut is solely regulated by the outermost LN layers, the flow of gradients to lower blocks could be more guaranteed with sufficient information for an update on their parameters, while direct flows that could oversimplify the complexity of the model are completely abstained. For all that, eAaT+ is devised as the ultimate version with end-to-end enhancements on representation issues from *Anchor2Vec* to Transformer Encoder blocks.

4. Experiment

4.1. *Implementation details*

Dataset. Three public indoor databases, namely UJIIndoorLoc [55], UTS [56], and UTS [57] for position estimation and floor classification, are adopted to strictly assess the proposed framework. Those are sparse databases with hundreds of anchors measured in multi-storey buildings. For UJIIndoorLoc, only around 232 out of 520 anchors in each fingerprint really work for each floor. Likewise, 557, and 779 out of 589, and 992 anchors are actually active in UTS, and Tampere respectively. Such missing anchors were filled with 100 dB for all these databases.

Evaluation metrics. For a comprehensive comparison, we adopt two kinds of metrics to evaluate localization performance. Firstly, the meter errors are defined by the Euclidean distance between estimated locations and their corresponding ground truth for a fine-grained localization evaluation. For further examination in the stability brought about by involved methods, two additional sub-metrics, 75th, and 95th Percentiles are in turn introduced. With less rigorous assessments, floor classification accuracy is also considered as the coarse-grained localization evaluation.

Evaluation Methodology. To ensure a fair comparison, we strictly adhere to pre-defined data split policies within datasets. Our evaluation involves examining metrics across state-of-the-art (SoTA) models, including Weighted-KNN [58], RADAR [1], DNN [59], CNNLoc [4], BayesCNN [60], and our preliminary version [19] (e.g., bAaT, and eAaT). Given the unique structural settings in each dataset, encompassing variations in building structures, the number and disposition of WiFi anchors, and layouts of objects, achieving optimal results consistently across different datasets underscores a model's robustness to environmental heterogeneity. Notably, this robustness is rigorously tested on Tampere dataset, designed for an unseen testing scenario, where only 697 points are used for training, while 3951 unseen points are reserved for testing.

Parameter setting. In our experiments, we interpret/consolidate an arbitrary sequence of raw anchors into a sequence of $k = 64$ tokens constituted by $\mathbf{d} = 128$ distinct attributes each for stabilities. Through optimizing embedding-level sub-constraints in Eq. (7),

Table 2
Ablation Study on AaTs.

Dataset	Model	Fixed LW	Adaptive RLW	$J_{Cov} + J_{Var}$	$J_{MAE/Cat}$	Tweaks	MAE (m)	Accuracy (%)
UJIIndoor	*(i)*	✓		✓	✓		8.43 (↑ 0.24%)	94.42
	(ii)		✓	✓	✓		8.40 (↑ 0.59%)	94.69 (↑ 0.29%)
	(iii)		✓	✓	✓	✓	**8.16**(↑ **3.43**%)	**95.14**(↑ **0.76**%)
	(iv)				✓		8.45	94.42
UTS	*(i)*	✓		✓	✓		6.93 (↓ 0.29%)	95.88
	(ii)		✓	✓	✓		6.86 (↑ 0.72%)	96.39 (↑ 0.53%)
	(iii)		✓	✓	✓	✓	**6.78**(↑ **1.88**%)	**97.17**(↑ **1.35**%)
	(iv)				✓		6.91	95.88
Tampere	*(i)*	✓		✓	✓		8.55 (↓ 0.23%)	92.29 (↓ 1.05%)
	(ii)		✓	✓	✓		8.52 (↑ 0.12%)	93.45 (↑ 0.19%)
	(iii)		✓	✓	✓	✓	**8.14**(↑ **4.57**%)	**93.72**(↑ **0.48**%)
	(iv)				✓		8.53	93.27

and Eq. (9) with ρ in sub-Eq. (8) set $1e-4$ as a small scalar preventing numerical instabilities, each token sequence in one batch, therefore, is encouraged to not only remain diverse among other sequences by γ of 1, but also among its $\mathbf{d} \cdot (k+1)$ attributes per se. Moreover, the number of transformer encoder blocks $\mathbf{L}$ is empirically determined 3, and 4 for fine- and coarse-grained scenarios respectively. Our architectures implemented in Tensorflow 2.9 were trained in batches of 256 each on a single GPU NVIDIA A100 40 GB for 400 epochs with the learning rate of $1e-4$ using Adam optimization. In addition, the process of adjustment to weighting factors of the involved tasks $\{\lambda_i\}_{i=0}^{2}$ is jointly self-learned for each batch during the training stage.

4.2. Model ablation

This subsection exhibits the specific contribution of each proposed part by decomposing the whole model into three separate configurations:

(i): An upgraded model with the inclusion of the proposed sub-constraints $J_{Cov}\&J_{Var}$ but the impacts between considered tasks are manually balanced by a traditional *Fixed Loss Weighting* (Fixed LW) manner where weighting factors were determined by a grid-search. The duration of the search process can be considerably mitigated with prior knowledge. Specifically, the searching range of weighting factors is restricted to $[1, 10]$ by the stride of 1, given that the $\lambda_0\&\lambda_2$ should be similar and always few times greater than λ_1, not only because we more emphasize the primary task and diversities among examples to avoid information collapse in within-batch examples, but also because the variation in attributes is inherently much more sensitive to model performance than that in within-batch examples. Thus, the λ_1 of 1 is kept unchanged throughout the adjustment of other factors. We carried out a thorough search on the UJIndoorLoc, which requires roughly 2 days to complete. The discovered weighting factors then are directly applied to the remaining datasets.

(ii): An end-to-end learning model (eAaT) that still keeps utilizing the same advantages of ongoing constraints as *(i)*, yet radically exerts these by an *Adaptive Random Loss Weighting* (Adaptive RLW), which is iteratively optimized in the training process with few light operations. This process is completed in one training round of 5 h, in which the model is learned how to resolve, and balance learning tasks at the same time.

(iii): An all-embracing model (eAaT+) that inherits all advantages of the proposed constraints from its predecessor, eAaT. Furthermore, this ultimate version is also undergone a complete overhaul, namely a new disposition of functional layers in all transformer encoder blocks. The transformation allows for the sufficient flow of the supervisory signals to the bottom blocks, which facilitates the exploitation of their attention mechanism for learning subtle but distinct representations.

(iv): A base model (bAaT) with all proposed tasks removed from *Anchor2Vec* layer that is trained singly with the primary task $J_{MAE/Cat}$. In this configuration, the model could solely employ the bare adaptive effect of *Anchor2Vec* without embedding-level enhancements, where the input sequence of raw anchors is simply interpreted/condensed depending on the input size to $k+1$.

Advantage of Constraints $J_{Cov}\&J_{Var}$. In Table 2, the models *(i)*, *(ii)*, which all incorporate sub-constraints $J_{Cov}\&J_{Var}$ into representation learning process of tokens, mostly show explicit improvements over the base model *(iv)* both in coarse- and fine-grained scenarios on three indoor datasets. For example, compared to the base model *(iv)* on UJIIndoorLoc, 0.24% and 0.59% MAE improvements are achieved by models *(i)*, and *(ii)* respectively in the fine-grained scenario while only 0%, and 0.29% accuracy differences are shown in the coarse-grained one. This is because such a coarse-grained scenario, *i.e.*, floor classification requires learning more relaxed representations that can carry more coarse and general information for the specific floors. However, the straight adoption of such constraints instead comes with counterproductive effects of imposing more restrictions on fine and meticulous cues in representation learning, thus curbing relaxation of seeking for expected representations. This restriction is no longer the case or at least mitigated when applied with Adaptive RLW at model *(ii)*. Still, the degradation in performance is still witnessed in model *(i)*, typically a marginal decline of 0.29% in MAE on UTS and 1.05% accuracy drop in Tampere compared to bAaT *(iv)*. The sparsity of those datasets can be explained as an important determinant associated with such deterioration. To clarify, fingerprints for the corresponding locations/floors in UTS and Tampere each comprise 589 and 992 anchors respectively

Table 3
Fine-grained localization analyses on three public datasets.

Dataset	Method	Mean absolute error (m)	75th percentile (m)	95th percentile (m)
UJIIndoorLoc	CNNLoc [4]	11.78/259.65†	299.24†	380.98†
	BayesCNN [60]	41.79	49.28	75.25
	Weighted-KNN [58]	9.33	11.19	26.86
	DNN [59]	133.40	170.85	213.10
	RADAR [1]	9.21	11.05	25.88
	bAaT [19]	8.45	10.64	20.41
	eAaT [19]	8.40	10.66	**20.33**
	eAaT+	**8.16**	**10.14**	24.84
UTS	CNNLoc [4]	7.60/14.53†	21.12†	28.64†
	BayesCNN [60]	16.38	23.15	34.75
	Weighted-KNN [58]	9.34	11.87	22.26
	DNN [59]	17.80	26.18	33.91
	RADAR [1]	9.26	11.76	22.26
	bAaT [19]	6.91	8.96	15.08
	eAaT [19]	6.86	**8.73**	**14.78**
	eAaT+	**6.78**	9.11	15.30
Tampere	CNNLoc [4]	10.88/-	-	-
	BayesCNN [60]	14.70	19.30	39.57
	Weighted-KNN [58]	10.45	12.56	32.06
	DNN [59]	32.83	44.80	62.57
	RADAR [1]	10.98	13.41	33.46
	bAaT [19]	8.53	10.17	24.35
	eAaT [19]	**8.52**	10.03	24.04
	eAaT+	**8.14**	**9.64**	**22.37**

†: Tested with publicly available weights on the author's GitHub [61].

but are almost contaminated with a large portion of missing anchors whose measurements filled out with default values of 100 dB. Accordingly, the fixed weighting-based constraints without any flexibility in optimization are not always able to effectively work out, or even worsen when forcing the model to synthesize such inherently sparse, and noisy fingerprints that retain merely a bit of useful information into tokens of distinct attributes.

Advantage of Adaptive Random Loss Weighting. Considering models *(i)*, *(ii)*, not only is its effectiveness expressed by performance boost both in MAE and Accuracy, but also the exploration costs regarding time- and computing- resources are considered. The observations from Table 2 suggest that Adaptive RLW in multi-task balancing could bring more convincing benefits to model performance with comparable, even better results at no further costs in exploration, optimization, and later inference. As above-mentioned, it is worth noting that Fixed LW needs a calibration interval for seeking hyper-parameters, namely weighting factors $\bar{\lambda}$, which were diminished to roughly 2 days with our prior knowledge. That is completely changed to one single training round of 5 h for *Adaptive RLW*, yet with even more excellent outcomes. Equally important, in most sparse datasets where the applications of Fixed RLW show slight reductions in accuracy (1.05% on Tampere) or insignificantly deteriorated performance in UJIIndoorLoc and UTS though, the use of Adaptive RLW constantly raises performance over the base model *(iv)* by notable margins of 0.12%, 0.72%, and 0.0.59% on Tampere, UTS, and UJIIndoorLoc, respectively.

Advantage of Tweaks. To fairly assess the benefits of the introduced tweaks, we compare and contrast the performance gain from individual models *(ii)* and *(iii)* on the base model *(iv)* as presented in Table 2. For the fine-grained representation scenarios, model *(iii)* constantly shows lower MAE than eAaT *(ii)* in leaps and bounds over three datasets. Notably, 3.34%, and 4.57% improvements over the base model *(iv)* in UJIndoor and Tampere respectively can be clearly witnessed in eAaT + *(iii)*, whilst, in this regard, mere insignificant improvements of 0.592% and 0.12% were made by eAaT *(ii)*. These advantages reflect the fact that the flow of gradients to every aspect of the whole architecture was fairly regulated with the tweaks made in layer disposition, especially for the bottom blocks. Furthermore, its validity to coarse-grained representation learning is also confirmed by a rise in accuracy compared with the counterpart *(ii)*. Typically, an accuracy increase of 1.35% over the bAaT *(iv)* on UTS is attained by eAaT + *(iii)*, approximately twice as greater as a mere improvement of 0.53% in eAaT *(ii)*. With plainly manifested benefits in the above scenarios, its success is supposedly attributed to regularization in the gradient flow that assists blocks close to the bottom with an adequate amount of information to accommodate both to data complexity and changes in the top blocks. In a sense, the localization performance of a given network heavily relies on the learning ability of the bottom blocks, over which the strength and solidity of underlying representations between RSS fingerprints and the corresponding locations are cumulatively established to supply to the top blocks. Therefore, maintaining a focus on the quality of representations learned from the bottom blocks is of importance to bring about a great boost in performance.

4.3. Fine-grained localization

From the results in Table 3, it can be seen that the proposed models completely surpass SoTA methods, which proves their competence in looking for detailed clues representing specific locations compared to others. More specifically with quantitative

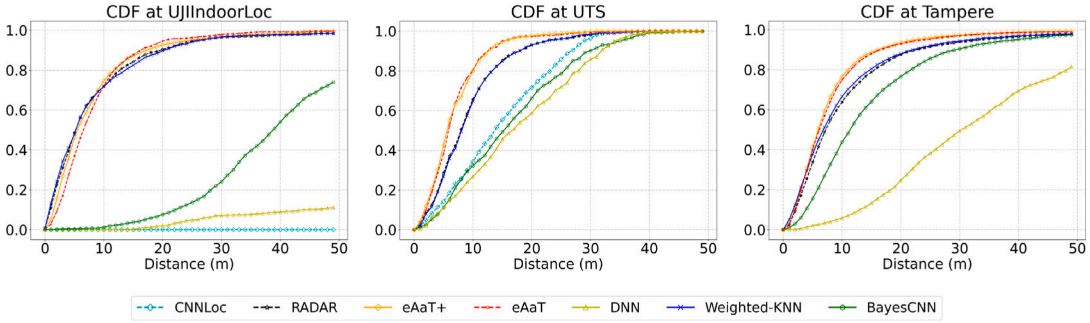


Fig. 4. Empirical Cumulative Distribution Function of SoTA methods on three public indoor localization datasets.

Table 4
Coarse-grained localization analyses for Floor hitting rate (%) on three public datasets.

Datasets	CNNLoc [4]	BayesCNN [60]	DNN [59]	**bAaT** [19]	**eAaT** [19]	**eAaT+**
UJIIndoorLoc	**96.03**	90.64	41.58	94.42	94.69	95.14
UTS	94.57	84.28	5.41	95.88	96.39	**97.17**
Tampere	**94.22**	91.24	32.01	93.27	93.45	93.72

results, the performance of BayesCNN dramatically plunges in all sparse datasets, particularly to 41.79 m in UJIIndoorLoc, which is even worse than conventional methods, such as RADAR, and Weighted-KNN with MAE of 9.21 m and 9.33 m respectively. Unlike these methods, our AaT variants show its generalization at the 1st place, typically, roughly 2.4 m MAE with 3.03 m in 75th Percentile, and 7.48 m in 95th Percentile lower than that of Radar in UTS. For the qualitative assessment, our consistency in performance is also clearly illustrated in Fig. 4 which markedly contrasts eAaT+&eAaT with the others via the accumulated errors. For the attestation of consistency and stability in the fine-grained scenario, eAaT+&eAaT absolutely triumphed over its counterparts by convincing margins across three datasets. This superiority over its rivals is credited for the advantages of the proposed constraints in the *Anchor2Vec*, which facilitate the stringency in representing sequences of raw anchors with sequences of more meaningful tokens. Apparently, the ability to attend relevant tokens in the form of multiple heads for subtle but distinct representations to specific locations should be considered. This is in contrast with the other competitors that were not equipped with any attention mechanism to such representations, thus being easily trapped by irrelevant information that is only useful in the training phase. Furthermore, with the aid of the introduced tweaks to alter layer dispositions, the ultimate version, i.e., eAaT+ completely overtook its predecessor (i.e., eAaT) to deliver superior localization performance with confidence. Concretely, eAaT+ outstripped eAaT presented in the preliminary version [19] with 2.86% and 4.46% reductions in MAE on UJIIndoor and Tampere respectively.

4.4. Coarse-grained localization

In lieu of scrutinizing detailed clues as in the fine-grained task, the coarse-grained task requires more relaxation in seeking coarse-grained patterns for different floors. As shown in Table 4, CNNLoc inheriting certain compressive impacts from the used vanilla autoencoder, and architectural inductive biases can obtain good accuracy on sparse datasets. Considering BayesCNN has no compressing ability, but inductive biases, it still achieves comparable performance in some sparse datasets. This adversity is mainly due to its input construction scheme inducing more noisy information instead. Compared to CNNs, simple DNN having no powerful inductive biases to exploit typical patterns severely suffers from the worst performance. For all that, the AaT variants can perform very well over sparse datasets. Our inferiority could largely lie in one of the model's inductive biases, namely *the locality* for which the model can pay attention to details through a stack of multi-head attention layers. Owing to this ability, its vision to general and coarse clues that are necessary for the floor classification task is partly limited. Nevertheless, some alleviation efforts have been made by the proposed sub-constraints using *Adaptive FLW*, which could be recognized by explicit differences between eAaT and its variant in Table 2, and Table 4. This limitation on the relaxation of learning coarse representations is ameliorated to some extent with eAaT+ showing comparable, or even sounder accuracy of 97.17% in UTS over all its rivals. This is because the expressiveness of the model is somewhat augmented with the regulated flow of gradients to bottom transformer encoder blocks, which far better provides the upper blocks with more general information to refine.

4.5. Inference time

Table 5 shows our flexibility to the rivals when having three variants of bAaT,eAaT and eAaT+ with small amounts of inference time of 2.5 ms, 4.2 ms, and 7.1 ms respectively. Through scenarios in Sub Section 4.3, and Sub Section 4.4, although DNN can reach only 1.2 ms per sample, its uncertainty to the prediction is varied to a great extent. However, our architectures which are much

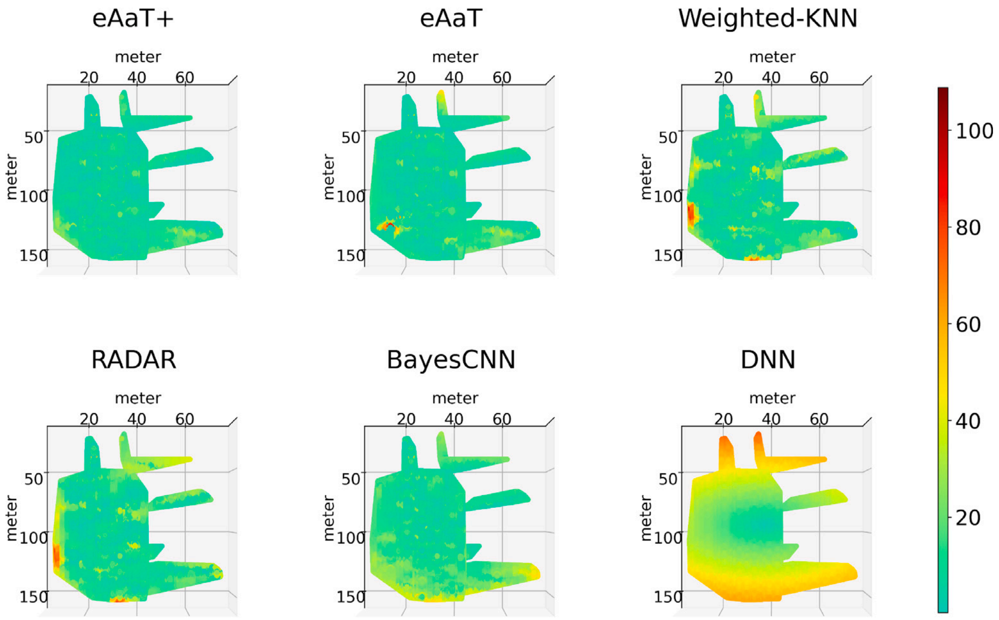


Fig. 5. 3D Top-view MAE Heat Maps performed by six typical methods, namely eAaT+, eAaT, Weighted-KNN, RADAR, BayesCNN and DNN on the 1st Floor of Tampere Map. Please zoom in to check the details. Best viewed in color.

Table 5
Overall inference time comparison for SoTA methods on UTS dataset.

Latency (ms)					
CNNLoc [4]	BayesCNN [60]	DNN [59]	bAaT [19]	eAaT [19]	eAaT+
6.9	4.1	1.2	2.5	4.2	7.1

more complicated than CNNLoc can achieve superior results within a mere 2.5 ms for bAaT and an extra 1.7 ms for eAaT. In an advanced manner, the mass introduction of architectural tweaks to arrange the layer disposition in all transformer encoder blocks for performance gain has slightly placed a computation burden on the ultimate version eAaT+ with an excess of 2.9 ms compared with eAaT. Established with its superiority all over its rivals in fine- to coarse-grained scenarios, trading efficiency for efficacy is worthy of further attention.

5. Visualization

In this section, the general effectiveness of the involved models can be visually observed in 3D Top View MAE heatmaps on the 1st floor of the Tampere Map as presented in Fig. 5. We adopt the 1st floor of the map for the qualitative evaluations since Tampere is the only dataset that provides the large number of testing points which is 5.6× greater than the total number of training points (3951 points vs 697 points). This domination not only encourages and expresses the model generalization to unseen locations but also is more plausible to display the heatmaps while efforts to interpolate an excessive number of missing points are completely superfluous. More specifically, we compute the MAEs at 1236 testing locations on the 1st floor map and then segment those into seven areas for four typical methods, namely eAaT, DNN, BayesCNN, and Weighted-KNN. For each area, the resolution is correctly enhanced by linearly interpolating missing points with computed internal MAEs points. These high-resolution areas thereafter are put back together into the floor map.

It can be simply noticed that DNN poorly performs in outer areas on the map, as indicated by the extremely warm color. Marked differences from this method could be observed in the Weighted-KNN and BayesCNN with clearer inner and outer areas in relatively cold tones both. Compared to the learning methods showing a smooth transition between colored areas, e.g., BayesCNN which however accumulates with yellow edge areas, the interpolation that requires no learning process is deemed to be partly beneficial to the Weighted-KNN. This is implied by only several sudden warm-colored areas on a sheer scale. Nevertheless, such sudden warm-colored areas suggest that the extrapolation or generalization to ones that are out of the training data cannot be guaranteed. In contrast to those, eAaT can deliver stable and reliable performance, where any errors from DNN, Weighted-KNN, and even BayesCNN are then significantly mitigated on eAaT side in cold tones. Having said that, a common red area showing a serious failure in estimation can be clearly observed in most of the potential contenders, including eAaT, Weighted-KNN, and RADAR. This inconsistency, however, is radically resolved in the presence of eAaT+, as indicated with all cold-colored areas on the 1st floor map of Tampere. As a result, the contributions of the introduced tweaks analyzed in theoretical and empirical aspects to overall AaTs are validated.

6. Conclusion

In this paper, we arrive at a novel view that learning one location from its fingerprints is rather analogous to capturing the context of an excerpt from its inner sentences in NLP. To this end, we propose for the first time the variants of the *Anchor-agnostic Transformers*, namely bAaT, eAaT, and eAaT+ that have been meticulously elaborated to effectively work on indoor fingerprint datasets. Ultimately, we successfully demonstrate that interpreting fingerprints into word sentences to represent specific locations can significantly address insurmountable problems of inconsistency in indoor localization.

CRediT authorship contribution statement

Son Minh Nguyen: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Duc Viet Le:** Supervision, Writing – review & editing. **Paul J.M. Havinga:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Son Nguyen reports financial support was provided by Key Digital Technologies Joint Undertaking.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the InSecTT project, https://www.insectt.eu/, funded by the ECSEL Joint Undertaking (JU) under grant agreement No 876038. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey. The document reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

This work made use of the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-2584.

References

[1] P. Bahl, V.N. Padmanabhan, RADAR: An in-building RF-based user location and tracking system, in: Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064), Vol. 2, Ieee, 2000, pp. 775–784.

[2] J. Niu, B. Wang, L. Cheng, J.J. Rodrigues, WicLoc: An indoor localization system based on WiFi fingerprints and crowdsourcing, in: 2015 IEEE International Conference on Communications, ICC, IEEE, 2015, pp. 3008–3013.

[3] W. Shao, H. Luo, F. Zhao, Y. Ma, Z. Zhao, A. Crivello, Indoor positioning based on fingerprint-image and deep learning, IEEE Access 6 (2018) 74699–74712.

[4] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, G. Fang, A novel convolutional neural network based indoor localization framework with WiFi fingerprinting, IEEE Access 7 (2019) 110698–110709.

[5] H.J. Jang, J.M. Shin, L. Choi, Geomagnetic field based indoor localization using recurrent neural networks, in: GLOBECOM 2017-2017 IEEE Global Communications Conference, IEEE, 2017, pp. 1–6.

[6] Z. Wang, L. Wu, Theoretical analysis of the inductive biases in deep convolutional networks, Adv. Neural Inf. Process. Syst. 36 (2024).

[7] J. Yang, H. Zhao, Deepening hidden representations from pre-trained language models for natural language understanding, 2019, arXiv preprint arXiv:1911.01940.

[8] I. Zimerman, L. Wolf, On the long range abilities of transformers, 2023, arXiv preprint arXiv:2311.16620.

[9] J. He, J.-N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, C. Wang, Transfg: A transformer architecture for fine-grained recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, No. 1, 2022, pp. 852–860.

[10] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, J. Gao, Focal attention for long-range interactions in vision transformers, Adv. Neural Inf. Process. Syst. 34 (2021) 30008–30022.

[11] K. Gavrilyuk, R. Sanford, M. Javan, C.G. Snoek, Actor-transformers for group activity recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 839–848.

[12] D. Gong, J. Lee, M. Kim, S.J. Ha, M. Cho, Future transformer for long-term action anticipation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 3052–3061.

[13] Y. Deng, F. Tang, W. Dong, C. Ma, X. Pan, L. Wang, C. Xu, Stytr2: Image style transfer with transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11326–11336.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16 × 16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[17] Y. Wang, R. Huang, S. Song, Z. Huang, G. Huang, Not all images are worth 16 × 16 words: Dynamic transformers for efficient image recognition, Adv. Neural Inf. Process. Syst. 34 (2021) 11960–11973.

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.

[19] S.M. Nguyen, D.V. Le, P.J. Havinga, Learning the world from its words: Anchor-agnostic transformers for fingerprint-based indoor localization, in: 2023 IEEE International Conference on Pervasive Computing and Communications, PerCom, IEEE, 2023, pp. 150–159.

[20] Z. Gu, Z. Chen, Y. Zhang, Y. Zhu, M. Lu, A. Chen, Reducing fingerprint collection for indoor localization, Comput. Commun. 83 (2016) 56–63.

[21] Y. Xie, Y. Wang, A. Nallanathan, L. Wang, An improved K-nearest-neighbor indoor localization method based on spearman distance, IEEE Signal Process. Lett. 23 (3) (2016) 351–355.

[22] J.-W. Jang, S.-N. Hong, Indoor localization with wifi fingerprinting using convolutional neural network, in: 2018 Tenth International Conference on Ubiquitous and Future Networks, ICUFN, IEEE, 2018, pp. 753–758.

[23] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, M. Youssef, WiDeep: WiFi-based accurate and robust indoor localization system using deep learning, in: 2019 IEEE International Conference on Pervasive Computing and Communications, PerCom, IEEE, 2019, pp. 1–10.

[24] M.T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, K. Reddy, Recurrent neural networks for accurate RSSI indoor localization, IEEE Internet Things J. 6 (6) (2019) 10639–10651.

[25] S.J. Pan, V.W. Zheng, Q. Yang, D.H. Hu, Transfer learning for wifi-based indoor localization, in: Association for the Advancement of Artificial Intelligence (AAAI) Workshop, Vol. 6, The Association for the Advancement of Artificial Intelligence Palo Alto, 2008.

[26] Z. Sun, Y. Chen, J. Qi, J. Liu, Adaptive localization through transfer learning in indoor Wi-Fi environment, in: 2008 Seventh International Conference on Machine Learning and Applications, IEEE, 2008, pp. 331–336.

[27] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[28] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving Language Understanding by Generative Pre-Training, OpenAI, 2018.

[29] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI Blog 1 (8) (2019) 9.

[30] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.

[31] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S.R. Bowman, GLUE: A multi-task benchmark and analysis platform for natural language understanding, 2018, arXiv preprint arXiv:1804.07461.

[32] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: International Conference on Machine Learning, PMLR, 2021, pp. 10347–10357.

[33] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[34] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F.E. Tay, J. Feng, S. Yan, Tokens-to-token vit: Training vision transformers from scratch on imagenet, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 558–567.

[35] Y. Li, K. Zhang, J. Cao, R. Timofte, L. Van Gool, Localvit: Bringing locality to vision transformers, 2021, arXiv preprint arXiv:2104.05707.

[36] C.-F.R. Chen, Q. Fan, R. Panda, Crossvit: Cross-attention multi-scale vision transformer for image classification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 357–366.

[37] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, Cvt: Introducing convolutions to vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 22–31.

[38] S. d'Ascoli, H. Touvron, M.L. Leavitt, A.S. Morcos, G. Biroli, L. Sagun, Convit: Improving vision transformers with soft convolutional inductive biases, in: International Conference on Machine Learning, PMLR, 2021, pp. 2286–2296.

[39] K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, W. Wu, Incorporating convolution designs into visual transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 579–588.

[40] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 568–578.

[41] A. El-Nouby, N. Neverova, I. Laptev, H. Jégou, Training vision transformers for image retrieval, 2021, arXiv preprint arXiv:2102.05644.

[42] M. Zhao, K. Okada, M. Inaba, Trtr: Visual tracking with transformer, 2021, arXiv preprint arXiv:2105.03817.

[43] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, C. Feichtenhofer, Multiscale vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 6824–6835.

[44] Y. Shavit, I. Klein, Boosting inertial-based human activity recognition with transformers, IEEE Access 9 (2021) 53540–53547.

[45] S. Mehta, M. Rastegari, Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer, 2021, arXiv preprint arXiv:2110.02178.

[46] S. EK, F. Portet, P. Lalanda, Lightweight transformers for human activity recognition on mobile devices, 2022, arXiv preprint arXiv:2209.11750.

[47] Z. Gao, X. Shi, H. Wang, Y. Zhu, Y.B. Wang, M. Li, D.-Y. Yeung, Earthformer: Exploring space-time transformers for earth system forecasting, Adv. Neural Inf. Process. Syst. 35 (2022) 25390–25403.

[48] Y. Liang, Y. Xia, S. Ke, Y. Wang, Q. Wen, J. Zhang, Y. Zheng, R. Zimmermann, Airformer: Predicting nationwide air quality in china with transformers, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, No. 12, 2023, pp. 14329–14337.

[49] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, C. Finn, Gradient surgery for multi-task learning, Adv. Neural Inf. Process. Syst. 33 (2020) 5824–5836.

[50] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, L. Van Gool, Multi-task learning for dense prediction tasks: A survey, IEEE Trans. Pattern Anal. Mach. Intell. (2021).

[51] B. Lin, F. Ye, Y. Zhang, A closer look at loss weighting in multi-task learning, 2021, arXiv preprint arXiv:2111.10603.

[52] S. Xie, H. Zhang, J. Guo, X. Tan, J. Bian, H.H. Awadalla, A. Menezes, T. Qin, R. Yan, ResiDual: Transformer with dual residual connections, 2023, arXiv preprint arXiv:2304.14802.

[53] S. Takase, S. Kiyono, S. Kobayashi, J. Suzuki, B2t connection: Serving stability and performance in deep transformers, in: Findings of the Association for Computational Linguistics: ACL 2023, 2023, pp. 3078–3095.

[54] L. Liu, X. Liu, J. Gao, W. Chen, J. Han, Understanding the difficulty of training transformers, 2020, arXiv preprint arXiv:2004.08249.

[55] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J.P. Avariento, T.J. Arnau, M. Benedito-Bordonau, J. Huerta, UjiIndoorLoc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems, in: 2014 International Conference on Indoor Positioning and Indoor Navigation, IPIN, IEEE, 2014, pp. 261–270.

[56] X. Song, X. Fan, X. He, C. Xiang, Q. Ye, X. Huang, G. Fang, L.L. Chen, J. Qin, Z. Wang, Cnnloc: Deep-learning based indoor localization with wifi fingerprinting, in: 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI, IEEE, 2019, pp. 589–595.

[57] E.S. Lohan, J. Torres-Sospedra, H. Leppäkoski, P. Richter, Z. Peng, J. Huerta, Wi-Fi crowdsourced fingerprinting dataset for indoor positioning, Data 2 (4) (2017) 32.

[58] G. Jekabsons, V. Zuravlyov, Refining Wi-Fi based indoor positioning, in: Proceedings of 4th International Scientific Conference Applied Information and Communication Technologies, AICT, Jelgava, Latvia, 2010, pp. 87–95.

[59] G. Félix, M. Siller, E.N. Alvarez, A fingerprinting indoor localization algorithm based deep learning, in: 2016 Eighth International Conference on Ubiquitous and Future Networks, ICUFN, IEEE, 2016, pp. 1006–1011.

[60] S. Sinha, D.V. Le, Completely automated CNN architecture design based on VGG blocks for fingerprinting localisation, in: 2021 International Conference on Indoor Positioning and Indoor Navigation, IPIN, IEEE, 2021, pp. 1–8.

[61] X. Song, Cnnloc, 2019, https://github.com/XudongSong/CNNLoc.