Learning the world from its words: Anchor-agnostic Transformers for Fingerprint-based Indoor Localization

Son Minh Nguyen, Duc Viet Le, Paul J. M. Havinga
University of Twente, Enschede, the Netherlands
{m.s.nguyen, v.d.le, p.j.m.havinga}@utwente.nl

***Abstract*—In this paper, we propose Anchor-agnostic Transformers (AaTs) that can exploit the attention mechanism for Received Signal Strength (RSS) based fingerprinting localization. In real-world applications, the RSS modality is inherently well-known for its extreme sensitivity to dynamic environments. Since most machine learning algorithms applied to the RSS modality do not possess any attention mechanism, they can only capture superficial representations, yet subtle but distinct ones characterizing specific locations, thereby leading to significant degradation in the testing phase. In contrast, AaTs are enabled to focus exclusively on relevant anchors at every Received Signal Strength (RSS) sequence for these subtle but distinct representations. This also facilitates the model to neglect redundant clues formed by noisy ambient conditions, thus achieving better accuracy in fingerprinting localization. Moreover, explicitly resolving collapse problems at the feature level (*i.e.*, none-informative or homogeneous features) can further invigorate the self-attention process, by which subtle but distinct representations to specific locations are radically captured with ease. To this end, we enhance our proposed model with two sub-constraints, namely covariance and variance losses that are mediated with the main task within the representation learning stage towards a novel multi-task learning manner. 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 provide much higher accuracy.**

***Index Terms*—Transformer, Self-Attention, CNNs, Indoor Localization, Indoor positioning, Deep Learning**

I. Introduction

The ever-increasing worldwide demand for smart space ecosystems (*e.g*, 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 wayfinding, asset tracking, and patient monitoring, to name but a few. For all that, owing to the ubiquity of WiFi, and Bluetooth signals, Received-Signal-Strength (RSS) fingerprint-based indoor localization has been well-established as the most flourishing stream where a massive number of off-the-shelf deep learning architectures from vision and natural language processing (NLP) tasks were altogether transferred. These methodologies absolutely prevail over the traditional methods, namely multilateration techniques with Time and Angle of Arrival that strictly require time-synchronization between radio emitters and receivers. Basically, the fingerprinting approaches initially require an offline phase for establishing a radio map (*i.e.*, fingerprint database), in which sequences of RSS anchors* (*i.e*, RSS fingerprints), and associated locations are manually surveyed. In the online phase, users' locations are determined by matching fingerprint observations against the fingerprint database.

Despite certain achievements in indoor localization using simple methodologies for kNNs' enhancements [1], [2], their performance is inclined to show more degradation both under sparse radio maps and dynamic environments (*e.g.*, temperature, humidity, static and moving objects that all lead to reflection, scattering, absorption in radio propagation, and power-constraint policies at emitters), not to mention a considerable amount of look-up time required for inference. With the arrival of CNN [3], [4]- and RNN [5]-based models, some of these deteriorations are partially mitigated by dint of architectural inductive biases that are directly beneficial to representation learning. To inherit transferable knowledge from pretrained CNN models on exploiting RSS sequences, prior works [3], [6], [7] have reshaped these RSS sequences into expected input sizes to the pretrained models. A proliferation of artifacts and false correlations among anchors in RSS sequences is fabricated and significantly intensified accordingly. This might help the model yield good results in the training phase but worsens it in the testing phase thereafter, due to the inconsistency of exploited patterns in these phases. In response, RNNs [8] were applied to involve temporal relationships among sequences of anchors while the structure of the input sequence is kept intact.

Although the aforementioned methods successfully gain leverage of CNN- and RNN-related inductive biases for

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.

*An anchor implies a radio-emitting source (*e.g.*, WiFi access points, Bluetooth beacons.

978-1-6654-5378-3/23/$31.00 ©2023 IEEE

seeking consistent features in terms of spatial and temporal domains respectively, the course of input transformation and sequential computation causes more artifacts in the transformed input and local exploration of superficial correlations between sequence anchors and the considered locations. Such detrimental effects are likely instead to guide the models to irrelevant representations. In short, their inferior performance in the testing phase largely lies in the following points: (1) RSS fingerprint is a hard modality that is inconceivable to human reasoning and very susceptible to environmental dynamics in time. Such adverse factors produce a mass of inconsistent clues to fingerprints. As a result, it is nontrivial to comprehensively capture subtle but distinct representations for specific locations. (2) The transformed input with noisy additives could not entirely benefit from the existing CNNs. That is, the convolution kernel in CNNs is well-functioned based on the assumption that adjacent elements must hold a true spatial correlation. It has proved the point to the image property where every pixel's color is highly relevant to its neighbors. (3) An original 1D fingerprint vector composed of independent anchors that are associated with the corresponding RSS values has very limited expressiveness of information between these anchors for any specific location. Thus, extensive interventions are desired to exactly discover latent correlations among these anchors that characterize the corresponding location. However, because of the reliance on sequential computation and local convolution filters that merely provide a small receptive field on local patterns, such vanilla RNN- and CNN-based models respectively lost sight of the whole picture on the RSS sequence, and thus failed to run their course to extract relevant features for specific locations.

We assume that a given RSS fingerprint corresponding to one location both holds inconsistent and consistent clues that were hidden in its latent structure, whereas the inconsistent ones are always all-pervasive. This is because of its vulnerability to environmental conditions where measurements of RSS fingerprints to a certain location are likely to be filled out with a large portion of highly noisy information. Moreover, excessive exposure to such inconsistent information in the representation learning stage induces significant degradation in localization performance, while being able to direct attention to the only clues that are consistent with related locations can bring benefits instead. For that point, there exists a high chance that one model without any adequate attention scheme is easily trapped with adverse information.

Heading for the adoption of the self-attention mechanism to draw global dependencies between input and output in a sentence, the Transformer [9] with the capability of parallel computing, which totally eschews recurrent computations, can faster reach the state of the art in many NLP tasks. In computer vision (CV), the convolutional architectures remain firmly established as state-of-the-art (SoTA) approaches over a wide variety of vision tasks, *e.g*, object detection, image recognition, and 3D reconstruction. In the past three years, the picture has been very different as a large number of the transformer's variants [10]–[13] have massively come into play with roaring success in a majority of fundamental vision challenges.

Motivated by the overwhelming superiority of transformers in NLP and CV, we attempt to adapt the Transformer to indoor localization by analogy between input sentences in NLP and RSS fingerprints. Compared to familiar modalities, such as audio, image, and text that have intrinsically proved to be spatially and temporally correlated among inner elements, fingerprints do not bear explicit correlations among inner anchors. In addition, not only is such a modality far more demanding for qualitative analysis than the familiar modalities because of the inconceivability and uninterpretability, but also extremely vulnerable to any levels of variations in the environment over time, which results in a big gap between training and testing phases. Nevertheless, the fingerprints at one location could show a few close resemblances to text sentences in several key respects. In particular, the context of an excerpt is represented by the meaning of several sentences inside, which goes the same for the indoor localization field where the context of one location is supposedly represented by a set of fingerprints. For that excerpt, the nuance of each sentence inside is defined by some of its keywords. In the same way, 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 to perform straight on 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 are treated the same way as word token embeddings in NLP applications where a stack of multi-head self-attention layers is fully applied. Going through these stacked attention layers can help progressively reveal the latent correlations among sequence anchors, which guides the model to subtle but distinct representations that are supposed to carry hidden information consistent with considered locations. Furthermore, we also proposed an enhanced Transformer dubbed **e**nhanced **A**nchor-**a**gnostic **T**ransformer (eAaT) that is conducive to the exploration of such latent correlations in the multi-head self-attention layers. To clarify, eAaT can avoid feature-level information collapse in an attempt to generate very informative token embeddings before the stacked attention layers, thus encouraging considerable restraints on irrelevant features while seeking subtle but distinct representations. This ability is enabled by being synergistic between two extra sub-constraints and the main task that are optimized together towards a novel multi-task learning fashion where all contributions from such constraints are equally adjusted.

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) that could meaningfully interpret an RSS fingerprint which is constituted by discrete anchors for a specific location.
- Furthermore, we also deliver an enhanced transformer (eAaT) that is furnished with two extra sub-constraints

deemed as unsupervised tasks to address the information collapse at the feature level, for which we again present a novel multi-task learning scenario referred to as Adaptive Random Loss Weighting (Adaptive RLW) to automatically learn to balance contributions between the learning tasks. This aims at easily and comprehensively utilizing the learned common knowledge among the learning tasks to improve performance on the main task of localization.
- We demonstrate our SoTA performances through extensive experiments and visual interpretation of public indoor localization datasets. Besides, this paper expects to provide fresh impetus for more transformer-based approaches to fingerprint-based indoor localization tasks.

II. Related Work

Within the last few years, many 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], [14], [15]) based on 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], [6] 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 [16] 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 a trajectory into the representation learning stage, Hoang [8] *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 [17], [18] 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 fully adopted to extract domain-independent representations.

Moreover, prior works on transformer-based architectures are worth mentioning. Since their inception 5 years ago [9], 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 sluggish models.

Transformers. Transformers were for the first time presented by Vaswani *et al.* [9] for machine translation, and have since ripened into the SoTA method in many NLP tasks. Continuously, Devlin *et al.*, [19] 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 [20]–[22] 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 [9], [19], [22], [23], Dosovitskiy *et al.* [11] 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 [24] is devised with a knowledge distilling-based learning to augment learned representations from ViT, which bested ResNet [25] by a significant margin. Some following works such as T2T-ViT [26], LocalViT [27] and CrossViT [28] arrive with improvements in the architecture design of ViT. Another line of research [29]–[31] attempts to transfer the inductive bias of CNN into Transformers. In addition, some endeavors [13], [32]–[35] are made to accommodate ViT to other vision tasks.

To push the boundaries of the aforementioned drawbacks in indoor localization, we directly elaborate a standard end-to-end transformer encoder architecture with a few tokenizing tweaks to fingerprinting problems. This adapted architecture is then capable of extensively directing its multi-head attention to global dependencies by taking into account the entire context sequence at once when refining one token embedding. With the support from parallel computation, this ability is further deepened by a stack of attention layers, which encourages the model to efficiently realize correlations and important clues hidden in the sequence anchors. Furthermore, we propose two embedding-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 every token in particular are more guaranteed, thereby facilitating multi-head attention layers to better disseminate their consideration to tokenized anchors according to the considered locations. To this end, eAaT jointly accomplishes these tasks by the proposed Adaptive RLW learning, a novel multi-task learning manner, in which the influence between subtasks and the main task on the learned gradients should be dynamically balanced over batches via the corresponding losses.

III. Proposed Model

A. Overview

We mainly follow the settings of BERT [19] in NLP where the input is expected to be a sequence of words. Assume a 1D RSS fingerprint $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{1\times\mathbf{N}}$ is defined by a sequence of $\mathbf{N}$ raw anchors, each of which contains a given RSS value obtained exclusively 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

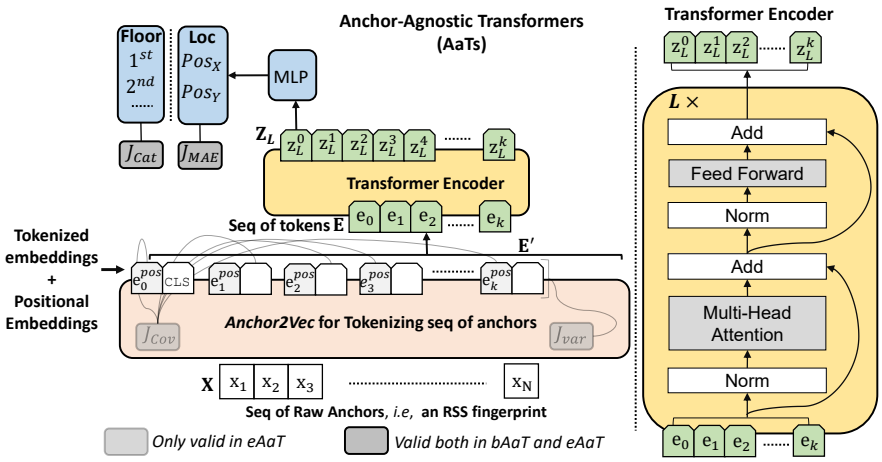

Fig. 1: An overview of **A**nchor-**a**gnostic **T**ransformers (AaTs). The pure Transformer is not designed to handle sequences of discrete anchors $\mathbf{X}$, *i.e* so-called RSS fingerprints, which also cannot be adequately unique to be represented for any given location. To this end, we design an *Anchor2Vec* Layer as anchor tokenization where these sequences are interpreted into sequences of meaningful token embeddings $\mathbf{E}$ in high dimensions, such that every single element inside is capable of carrying more contextual attributes. For each sequence or an individual fingerprint, the tokenized anchors were attached with `[CLS]` serving as a summarized representation of the entire sequence, which can be used further for classification or regression. We also injected learnable positional embeddings $\mathbf{E}_{pos}$ to include positional clues among the tokenized anchors. The resulting sequence of vectors $\mathbf{E}$ is then ready for the basic Transformer of $\mathbf{L}$ encoders (bAaT).

in Fig.1, owing to our designed tokenizing process at the *Anchor2Vec* layer, which is detailed in Subsec.III-C, a given 1D RSS fingerprint $\mathbf{X}$ is interpreted into more meaningful token embeddings $\mathbf{E} = [\mathbf{e}_0, \cdots, \mathbf{e}_k]^{\mathrm{T}} \in \mathbb{R}^{(k+1)\times \mathbf{d}}$, which each better manifests important information under $\mathbf{d}$ distinct attributes as word token embeddings did in NLP. To be more specific, $\mathbf{X}$ is first linearly mapped to an intermediate embedding of k-dimensions, which serves as a fixed-size buffer that is flexible to the various number of anchors, suggesting an adaptive effect on representation refinement. In particular, the effect expects to give rise to either feature consolidation for sparse RSS fingerprints in high-dimension or feature interpretation for dense RSS ones with lower dimension. Subsequently, every element inside the intermediate embedding is enriched with information-expressibility in dimension by further expansion to a vector of $\mathbf{d}$ dimensions. In the meanwhile, the expansion is closely supervised by $J_{cov\&var}$ both in dimension and batch for effective and faithful enrichment. That can significantly diminish information collapse in the tokenizing process.

By analogy with BERT's `[CLS]` token, we do prepend a learnable embedding `[CLS]` of $\mathbf{d}$ dimensions to the sequence of token embeddings. Its output from the Transformer encoder $\mathbf{z}_L^0$ serves as a summarized representation of the entire sequence that can be used both for regression and classification down the line. Note that spatial information-awareness among the tokens in the sequence, at the same time, was also raised with the addition of a learnable 1D positional embedding $\mathbf{E}_{pos} \in \mathbb{R}^{1\times(\mathbf{k}+1)\cdot\mathbf{d}}$ for the ultimate token embeddings $\mathbf{E}$ that is ready for use of the Transformer thereafter.

B. 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

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

where Q, K, and V are all collections of projected features, each of which is represented by a d-dimensional vector. To be specific, $Q = [q_0, \cdots, q_k]^{\mathrm{T}} \in \mathbb{R}^{(k+1)\times d}$ 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

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

Where each head is expressed as follows:

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

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

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

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.

C. Anchor2Vec Layer

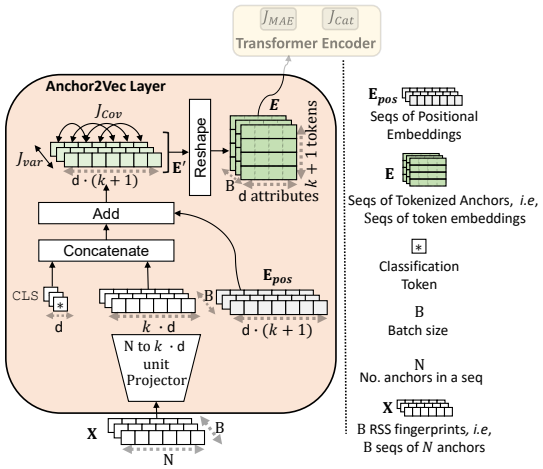

Fig. 2: Anchor2Vec Layer. The layer is to disentangle $\mathbf{B}$ sequences of $\mathbf{N}$ discrete anchors into $\mathbf{B}$ sequences of $k+1$ dimensional embeddings $\mathbf{E} \in \mathbb{R}^{(k+1)\times \mathbf{d}}$, each possessing $\mathbf{d}$ different attributes, including positional correlations among sequence tokens by means of learnable positional embeddings $\mathbf{E}_{pos}$. For clarification, a vector of $\mathbf{N}$ anchors are initially linear-projected to a k-dimensional vector $\in \mathbb{R}^k$ for the adaptive effect (*e.g*, compression to large $\mathbf{N} >> k$ or interpretation to minor $\mathbf{N} < k$). After that, the vector is further interpreted to another higher dimension, in which each element inside is expanded to $\mathbf{d}$ attributes. This suggests that the whole vector $\mathbf{E}'$ should hold totally $(k+1)\cdot\mathbf{d}$ different attributes for the considered location, including `[class]` token of $\mathbf{d}$ dimensions before being reshaped into the token embeddings $\mathbf{E}$. To eschew feature-level information collapse, two sub-constraints are imposed in eAaT for strictly forcing the model to produce unique sequences together with informative tokens inside that carry distinct attributes.

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 $\mathbf{E}$ thereafter serves as input to the encoders. As regards the eAaT, there is an extra step needed, in which the token embeddings $\mathbf{E}$ 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 main 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 $\mathbf{E}$ by decorrelating the entire attributes of the sequence token embedding over a batch of $\mathbf{B}$ 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 $\mathbf{E}$, where only every $\mathbf{d}$ 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 $\mathbf{E}$, which strictly encourages 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

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

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

$$J_{\mathrm{Cov}} = \frac{1}{(k+1)\cdot\mathrm{d}}\sum_{i\neq j}\left[Cov\left(\mathrm{E}'\right)\right]_{i,j}^{2} \tag{6}$$

The Eq.6 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 $\bar{E}'$.

$$Std\left(\mathrm{E},\,\rho\right) = \sqrt{\frac{\sum_{i=0}^{\mathrm{B}}\left(\mathrm{E}'_i - \bar{\mathrm{E}}'\right)^2}{\mathrm{B}-1} + \rho} \tag{7}$$

Based on Eq.7, we define the variance constraint via a hinge loss function:

$$J_{Var} = \frac{1}{(k+1)\cdot\mathrm{d}}\sum_{j=1}^{(k+1)\cdot\mathrm{d}}\max\left(0,\ \gamma - Std\left(\mathrm{E},\,\rho\right)\right) \tag{8}$$

where γ is the desired average standard deviation, and ρ is a very small scaler to keep Eq.7 valid at all times.

With the benefit of Eq.6, and Eq.8, 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.

D. Adaptive Random Loss Weighting in Multi-task Learning

To mediate disputes between learning tasks to effectively learn relevant representation for the main 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 [36], thus leading to unsatisfactory performance for other tasks. Such phenomena are related to the task-balancing problem [37] in MTL, where balancing the influences between sub-tasks and the main task remains very challenging. Rather than directly intervening in learned gradients of considered tasks, the work [38] 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.9 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.

$$J_{overall} = \bar{\lambda}_0 J_{MAE/Cat} + \bar{\lambda}_1 J_{Cov} + \bar{\lambda}_2 J_{Var} \tag{9}$$

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 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 α, dataset D, model parameters θ, weight distribution $\{\rho_i(\bar{\lambda})\}_{i=0}^{P-1}$
Output: task-specific weights $\{\bar{\lambda}_t\}_{t=0}^{T-1}$, task-specific losses $\{J_t\}_{t=0}^{T-1}$

Initialization: $\lambda_{dis} \in \mathbb{R}^{\mathrm{T}\times P}$, $\bar{\lambda} = [\bar{\lambda}_0, \cdots, \bar{\lambda}_{T-1}]$, $\lambda \in \mathbb{R}^T$
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} = 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} = 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**
  $\lambda_{dis}[:T, i] \sim \rho_i(\bar{\lambda})$
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
$\bar{\lambda}$ = softmax(λ) // Computing task-specific weights
return $\{\bar{\lambda}_t\}_{t=0}^{T-1}$, $\{J_t\}_{t=0}^{T-1}$

IV. Experiment

A. Implementation Details

Dataset. Three public indoor databases, namely UJIIndoorLoc, UTSIndoorLoc, and TampereIndoorLoc 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 UTSIndoorLoc, and TampereIndoorLoc respectively. Such missing anchors were filled with 100 dB for all these databases.

Evaluation metrics. To have a fair 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.

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.6, and Eq.8 with ρ in sub-Eq.7 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 40GB 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.

B. 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 main 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 hours, in which the model is learned how to resolve, and balance learning tasks at the same time.

(iii): A base model (bAaT) with all proposed tasks removed from *Anchor2Vec* layer that is trained singly with the main 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 Tab.I, 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 *(iii)* both in coarse- and fine-grained scenarios on three indoor datasets. For example, compared to the base model *(iii)* 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 UTSIndoorLoc and 1.05% accuracy drop in Tampere compared to bAaT *(iii)*. 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 UTSIndoorLoc and Tampere each comprise 589 and 992 anchors respectively 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 Tab.I 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 hours 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 UTSIndoorLoc though, the use of Adaptive RLW constantly raises performance over the base model *(iii)* by notable margins of 0.12%, 0.72%, and $0.0.59\%$ on Tampere, UTSIndoorLoc, and UJIIndoorLoc, respectively.

C. Fine-grained Localization

From the results in Tab.II, 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 detailed

TABLE I: Ablation Study on AaTs

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

TABLE II: 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 [7]	41.79	49.28	75.25
	Weighted-KNN [39]	9.33	11.19	26.86
	DNN [40]	133.40	170.85	213.10
	RADAR [1]	9.21	11.05	25.88
	bAaT	8.45	**10.64**	20.41
	eAaT	**8.40**	10.66	**20.33**
UTS	CNNLoc [4]	7.60 / 14.53†	21.12†	28.64†
	BayesCNN [7]	16.38	23.15	34.75
	Weighted-KNN [39]	9.34	11.87	22.26
	DNN [40]	17.80	26.18	33.91
	RADAR [1]	9.26	11.76	22.26
	bAaT	6.91	8.96	15.08
	eAaT	**6.86**	**8.73**	**14.78**
Tampere	CNNLoc [4]	10.88 / -	-	-
	BayesCNN [7]	14.70	19.30	39.57
	Weighted-KNN [39]	10.45	12.56	32.06
	DNN [40]	32.83	44.80	62.57
	RADAR [1]	10.98	13.41	33.46
	bAaT	8.53	10.17	24.35
	eAaT	**8.52**	**10.03**	**24.04**

†: Tested with publicly available weights on the author's GitHub [41].

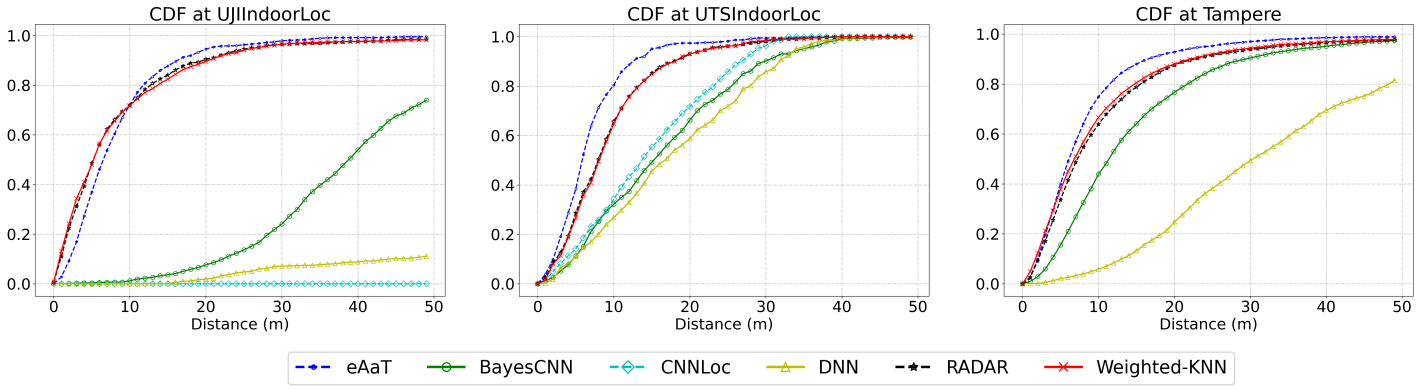

Fig. 3: Empirical Cumulative Distribution Function of SoTA methods on three public indoor localization datasets.

quantitative results, the performance of BayesCNN dramatically plunges in all sparse datasets, particularly to 41.79m in UJIIndoorLoc, which is even worse than conventional methods, such as RADAR, and Weighted-KNN with MAE of 9.21m and 9.33m respectively. Unlike these methods, our AaT variants show its generalization at the 1^{st} place, for example, roughly 2.4m MAE with 3.03m in 75^{th} Percentile, and 7.48m in 95^{th} Percentile lower than that of Radar. For the qualitative assessment, our consistency in performance is also clearly illustrated in Fig.3 which contrasts the eAaT with the others via the accumulated errors. Throughout these plots of the fine-grained competitions for consistency and stability verification, the eAaT still triumphed over its counterparts by convincing margins. This superiority over its rivals is attributed to the advantages of the proposed components in the *Anchor2Vec* for thoroughly representing sequences of raw anchors with sequences of more meaningful tokens. Moreover, the intensive, and extensive attention ability of the proposed model via its multiple heads in multiple stages to relevant tokens for subtle but distinct representations to the specific locations comes into effect. 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.

TABLE III: Coarse-grained localization analyses for Floor hitting rate (%) on three public datasets.

Datasets	CNNLoc	BayesCNN	DNN	**bAaT**	**eAaT**
UJIIndoorLoc	**96.03**	90.64	41.58	94.42	94.69
UTS	94.57	84.28	5.41	95.88	**96.39**
Tampere	**94.22**	91.24	32.01	93.27	93.45

TABLE IV: Overall inference time comparison for SoTA methods on UTS dataset.

Method	Time (*ms*)
CNNLoc	6.9
BayesCNN	4.1
DNN	1.2
bAaT	2.5
eAaT	4.2

D. 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 Tab.III, CNNLoc inheriting certain compressive impacts from the used vanilla auto-encoder, 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, especially eAaT 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 despite some alleviation efforts having been made by the proposed sub-constraints using *Adaptive FLW*, which could be recognized by explicit differences between eAaT and its variant in Tab.I, and Tab.III.

E. Inference Time

Tab.IV shows our flexibility to the rivals when having two variants of bAaT and eAaT with small amounts of inference time of 2.5 *ms* and 4.2 *ms* respectively. Through scenarios in Subsec.IV-C, and Subsec.IV-D, 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 more complicated than CNNLoc can achieve superior results within a mere 2.5 *ms* for bAaT and an extra 1.7 *ms* for eAaT.

V. Visualization

In this section, the general effectiveness of the involved models can be visually observed in 3D Top View MAE heatmaps on the 1^{st} floor of the Tampere Map as presented in Fig.4. We adopt the 1^{st} floor of the map for the qualitative evaluations since Tampere is the only dataset that provides the large number of testing points which is $5.6\times$ 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

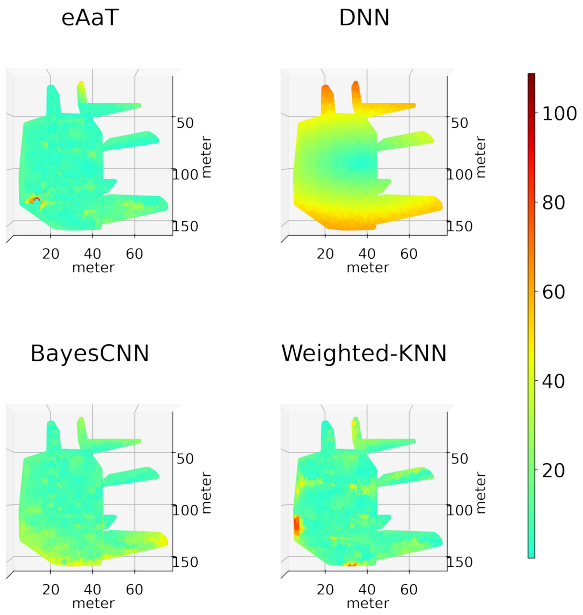

Fig. 4: 3D Top-view MAE Heat Maps performed by four typical methods, namely eAaT, DNN, BayesCNN, and Weighted-KNN on the 1^{st} Floor of Tampere Map. Please zoom in to check the details. Best viewed in color.

number of missing points are completely superfluous. More specifically, we compute the MAEs at 1236 testing locations on the 1^{st} 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 ability of BayesCNN which has yellow edge areas, the interpolation property that requires no learning process is deemed to be partly beneficial to the Weighted-KNN. This is implied by some sudden warm-colored areas, which suggests that the extrapolation or generalization to ones that are out of the training data cannot be guaranteed. In contrast to those, the eAaT can deliver stable and reliable performance, where any errors from DNN, Weighted-KNN, and even BayesCNN are then significantly mitigated on the eAaT side in cold tones.

VI. 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, 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 a specific location can significantly address insurmountable problems of inconsistency in indoor localization.

REFERENCES

[1] P. Bahl and 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, and 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, and A. Crivello, "Indoor positioning based on fingerprint-image and deep learning," *Ieee Access*, vol. 6, pp. 74 699–74 712, 2018.

[4] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, and G. Fang, "A novel convolutional neural network based indoor localization framework with wifi fingerprinting," *IEEE Access*, vol. 7, pp. 110 698–110 709, 2019.

[5] H. J. Jang, J. M. Shin, and 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] J.-W. Jang and 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.

[7] S. Sinha and 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.

[8] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate rssi indoor localization," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 639–10 651, 2019.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[11] 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 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[12] Y. Wang, R. Huang, S. Song, Z. Huang, and G. Huang, "Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 960–11 973, 2021.

[13] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[14] Z. Gu, Z. Chen, Y. Zhang, Y. Zhu, M. Lu, and A. Chen, "Reducing fingerprint collection for indoor localization," *Computer Communications*, vol. 83, pp. 56–63, 2016.

[15] Y. Xie, Y. Wang, A. Nallanathan, and L. Wang, "An improved k-nearest-neighbor indoor localization method based on spearman distance," *IEEE signal processing letters*, vol. 23, no. 3, pp. 351–355, 2016.

[16] M. Abbas, M. Elhamshary, H. Rizk, M. Torki, and 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.

[17] S. J. Pan, V. W. Zheng, Q. Yang, and 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.

[18] Z. Sun, Y. Chen, J. Qi, and 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.

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

[20] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[21] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[22] 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," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

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

[24] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and 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.

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

[28] C.-F. R. Chen, Q. Fan, and 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.

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

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

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

[32] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and 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.

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

[34] M. Zhao, K. Okada, and M. Inaba, "Trtr: Visual tracking with transformer," *arXiv preprint arXiv:2105.03817*, 2021.

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

[36] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5824–5836, 2020.

[37] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

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

[39] G. Jekabsons and 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.

[40] G. Félix, M. Siller, and 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.

[41] X. Song, "Cnnloc," https://github.com/XudongSong/CNNLoc, 15th March 2019.